Beyond Design Thinking for Personal Efficiency and Creativity

The Path of the Personal Innovator

Mats Danielson

DECISION
RIGHT
WRONG

Beyond Design Thinking for Personal Efficiency and Creativity

The Path of the Personal Innovator

Mats Danielson

Sine Metu

Published by Sine Metu Productions, Stockholm, Sweden
Website: www.sinemetu.se

Substantial efforts have been made to publish only accurate information. However, the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. Nobody makes any money from this book. Rather, the author has spent a substantial amount of time and money producing it. This free open-access e-book is made available under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 licence, and we hope you enjoy it. You are free to share and redistribute it as long as you or any other party involved do not change it, make any money out of it or have it exchanged for any goods, services or other gains. Up to 300 words can be freely quoted if the source is acknowledged – for larger quotes, contact the publisher.

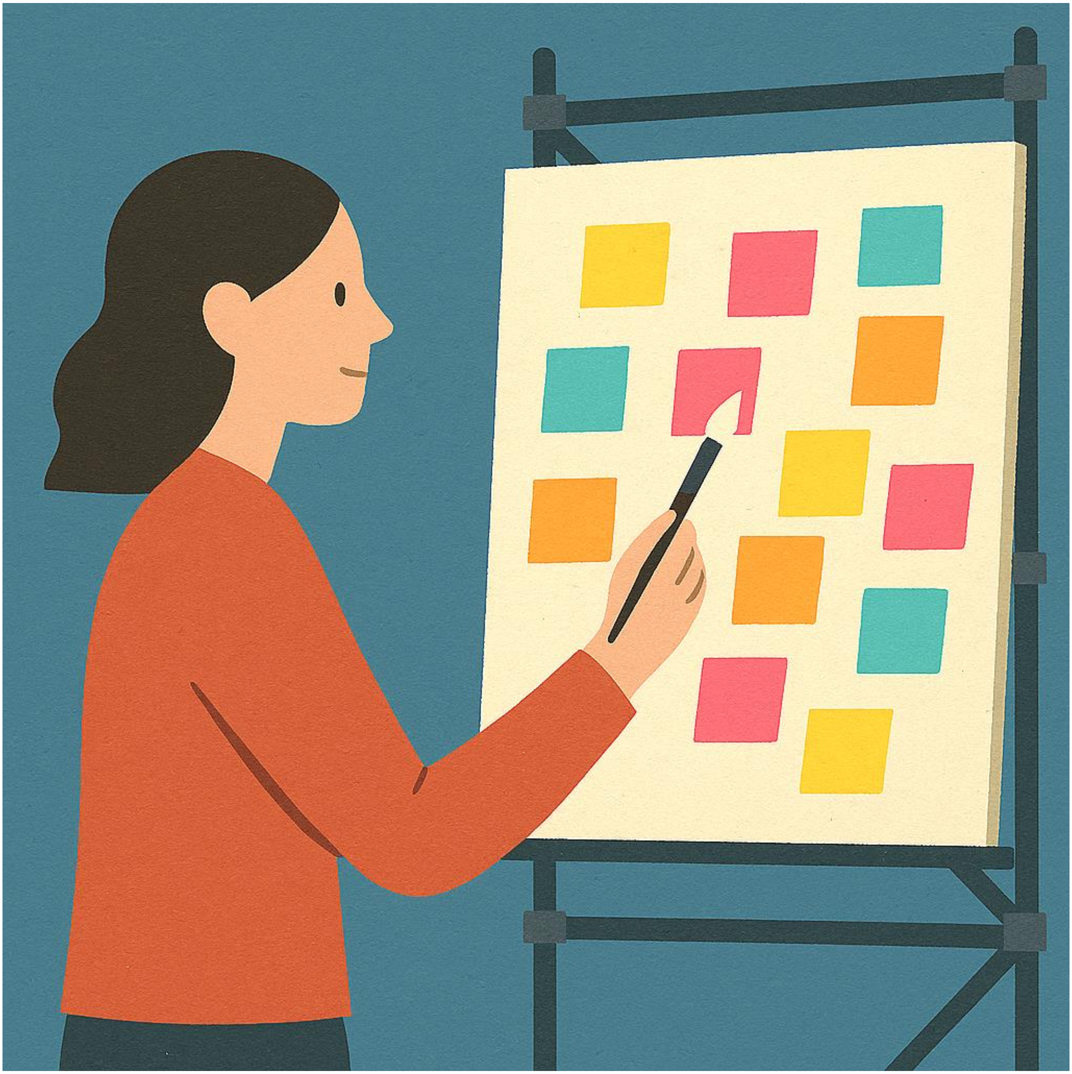
Images: Artwork by Martno

ISBN 978-91-527-5305-7

Second printing, December 2022

Design Thinking taps into capacities we all have but that are overlooked by more conventional problem-solving practices. It is not only human-centred; it is deeply human in and of itself.

Tim Brown, Change by Design, 2009



Personal innovators use many tools to attain their goals

Contents

Preface

This book is for everyone who wants to become more innovative and creative in their daily life. It is not a traditional treatise of team-based Design Thinking processes, nor is it another self-help manual promising quick personal transformation. Rather, it introduces a set of tools and perspectives that can be used by everyone: students, professionals, educators, and anyone else seeking to approach life with more intention, imagination and flexibility. These tools come primarily from Design Thinking, which we build upon and extend for a unique purpose: to support personal innovation, development and growth. In these pages, the reader is invited to explore how Design Thinking, along with other creative and innovative methods, can be reinterpreted and recombined to cultivate what this book calls a *personal innovator*.

The idea of the personal innovator sits at the heart of this book. It is a term with dual significance. On the one hand, it refers to innovating your own life. To shaping your goals, habits, relationships and decisions with greater clarity and creativity. On the other, it refers to being a person who contributes to innovation in the workplace and other social settings, not only as a participant in structured design projects but also as an individual contributor or team member working in contexts where formal innovation processes may not even exist. Many of us face problems in our work lives that are messy, poorly defined, or stuck in legacy patterns. Most of these are not traditional Design Thinking settings, but they can still benefit from using such tools and methods. The personal innovator brings that possibility to life.

Design Thinking (DT) is often presented as a structured process containing the phases Empathise, Define, Ideate, Prototype and Test. This model has been widely applied in product design, service innovation, etc. Some well-known books aim to make this process also apply to individuals. They either have a classical self-help tone or presume that the reader is part of a design team or organisation actively applying DT. This book departs from both assumptions. It does not sell general inspiration, nor does it assume that you work in an organisation that uses sticky notes or innovation workshops. Instead, it offers something more broadly applicable: a way of thinking that can improve how you live and how you contribute to the lives and work of others, using a carefully selected and clearly explained set of methods.

That distinction is important. A vast number of self-help books already exist, including some that draw upon Design Thinking. These often present simplified personal narratives, focus on mindset over method or promote optimism without sufficient attention to tools. There are also hundreds of books on Design Thinking as a method, most of which describe in detail how traditional DT teams operate within companies or consultancies. These are valuable in their own right, but they tend to be oriented towards professional designers in corporate contexts. This book fills a different gap. It shows how DT can be used by individuals, not only to support creative teamwork but also to frame and solve the challenges that arise in personal and professional life. Challenges that are often complex, ambiguous, and unique to each person’s path. We adhere to the view that personal development is a designable process. Your life, like any product or service, is shaped by the choices you make and the systems you build around yourself. You can either choose to leave it to chance or habit, or you can choose to engage with it more consciously. Design Thinking provides a disciplined yet open framework for doing just that. It encourages curiosity, empathy, and experimentation. It prompts you to frame problems more clearly, explore alternatives more boldly, and learn from failure more systematically.

However, Design Thinking alone is not enough. Traditional DT tends to focus on collective innovation in structured environments. But much of life’s creativity takes place outside such boundaries. That is why this book expands the toolbox. In addition to Design Thinking methods, we introduce techniques from adjacent fields such as creativity training, systems thinking, visual mapping, scenario development and reflective practice. The aim is not to overload the reader with frameworks but to provide a versatile set of practical tools that can be mixed and matched based on the nature of the problem or opportunity at hand. The emphasis is always on usability and clarity. You do not need to be a trained designer or psychologist to use these methods effectively.

This approach is especially valuable in work environments that are not explicitly innovation-driven. In most organisations, innovation is not a dedicated project with a neat beginning and end. Instead, it is something that arises sporadically in meetings, negotiations, conflicts, and everyday decision-making. Most professionals do not lead design sprints or run workshops. Yet, they still face situations that call for fresh thinking: a service that is failing its users, a process that could be reimagined, or a new role that demands rapid adaptation. This is where the tools in this book become particularly useful. They are not tied to job titles or project roles. They are meant for anyone with the motivation to bring more creativity, clarity and responsiveness to their work regardless of sector and hierarchical position.

The concept of the personal innovator also invites a shift in identity. We often think of innovation as something that others do: artists, specialists, leaders, and innovators. This mindset limits our potential. By contrast, to see yourself as a personal innovator is to recognise your capacity to change things in novel ways, both for yourself and for those around you. It is a quiet but radical stance. It suggests that innovation is not the exclusive domain of ultra-gifted minds or design studios, but a broader capacity that can be cultivated across life roles and domains. It also reminds us that most innovations are not technological. Sometimes, they are shifts in how we understand a problem, relate to others, or reframe a stuck situation.

Importantly, this book does not promote its methods as a silver bullet. It respects the fact that many, but far from all, problems can be solved with post-it notes and brainstorms. Nor does it assume that the reader will always have the time or resources to run through a full design process. What it offers instead is a flexible repertoire. A way to think, reflect and act with greater creativity, no matter what constraints you face. You might use one technique from the book in a moment of reflection, another in a job conversation, and a third in mapping out options for a life transition. Over time, you may develop your own personalised version of the toolbox, selecting the methods that resonate most with your style and situations.

The tone throughout is intentionally pragmatic. While the book is grounded in theory, it avoids academic jargon. It explains each method with sufficient background to understand where it comes from and why it works, but the focus is always on practical application. Most chapters contain examples, instructions and variations. The goal is to make the techniques as accessible as possible without reducing them to empty formulas. Readers are encouraged to experiment and adapt rather than follow rigid rules. This pragmatic spirit also extends to how the book defines success. In a world saturated with performance metrics and optimisation hacks, it is tempting to see innovation as a competitive race or a personal branding exercise.

This book takes a different view. To be a personal innovator is not to outshine others or to become a constant idea generator. It is to notice problems worth solving, to frame challenges constructively, to test possible responses, and to learn from what happens. This process is not linear or predictable. It involves doubt, iteration and reflection. But it is also very rewarding. It leads to insights that are deeply personal and often transferable. And it develops skills that are useful across roles and careers.

The term *personal innovator* is used throughout the book with care. It captures two key convictions: first, that meaningful change begins when individuals take initiative, rather than waiting for external permission or ideal conditions; and second, that such change need not be abrupt, forceful or disruptive. Innovation in this sense is not about upheaval, it is about thoughtful exploration, creative reframing, and the gradual reshaping of habits, decisions and environments. The personal innovator engages with life as something to be designed, not dictated. The book is structured to support different entry points. Readers who are new to Design Thinking will find a gentle introduction and a focus on real-life applications. Those familiar with DT will discover extensions and combinations that take the methods into new territory. Throughout, the emphasis is on helping the reader build confidence, not only in using creative tools but also in trusting their own capacity for change. The ultimate aim is empowerment: to help more people see themselves as active participants in shaping their lives and work environments.

In short, this book is an invitation to rethink what it means to innovate. It argues that innovation is not reserved for a lucky few but is a distributed capacity that anyone can develop. It shows how DT and related methods can be used beyond their original domains to support professional and personal growth. And it proposes a new kind of practitioner, the personal innovator, who bridges the space between structured innovation projects and everyday problem-solving. Whether you are facing a career decision, leading a team, planning a project, or simply seeking more clarity in your next steps, the tools in this book can support you. You do not need to wait for permission, resources or perfect conditions to begin. In fact, you should not. The work of personal innovation starts where you stand right now.

Personal innovators use many tools to attain their goals

1. Design Thinking

Design Thinking (DT) has gained prominence as a human-centred approach to innovation in business and design contexts. It is traditionally presented as a team-based, user-focused process for developing products or services, characterised by stages such as empathising with users, defining problems, ideating solutions, prototyping, and testing. Tim Brown, CEO of the design consultancy IDEO, defined Design Thinking as a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and what a viable business strategy can convert into customer value and market opportunity. Importantly, Brown and others emphasise that DT allows people who aren't trained as designers to use creative tools to address a vast range of challenges. In other words, DT is not exclusive to design teams, it is fundamentally a versatile problem-solving toolkit that anyone can learn and apply.

While most literature and case studies describe Design Thinking in the context of collaborative innovation (e.g. multidisciplinary teams developing a new product), the same principles, models, and techniques can be adapted to an individual's daily work. In fact, scholars and practitioners note that Design Thinking begins with personal mindset shifts. It starts with ourselves and working on ourselves as we tackle problems. The question explored in this chapter is how a personal innovator or professional can harness DT as a personal efficiency and creativity toolkit. Rather than treating DT as a broad organisational process, we will examine it as a practical set of methods and mental models that can improve one's work style and performance on an everyday basis, thus perhaps becoming a life-changing philosophy in the end.

The need for personal innovation tools is greater than ever in the modern knowledge economy. Professionals often face complex, ambiguous problems in their work, from figuring out how to present information in a compelling way, over streamlining workflows, to generating creative solutions under tight constraints. Traditional productivity advice (such as to-do lists and time hacks) has its place, but when problems are ill-defined or require creative insight, Design Thinking offers a robust framework. By adopting Design Thinking mindsets, such as focusing on human needs, reframing problems, brainstorming prolifically, visualising ideas, and iterating through rapid experimentation, individuals can open up to innovation and new ways of thinking about problems and solutions in their personal work.

This chapter will first outline the key mindsets and principles of Design Thinking relevant to personal application. It will then delve into a range of DT tools and techniques (reframing, empathy, ideation methods, prototyping, systems thinking, and visualisation). We will illustrate how each can be employed by a personal innovator or small-group contexts, with support from academic insights. Importantly, we include concrete examples of how personal innovators or consultants might use these approaches in daily practice. The focus is on real, grounded applications of DT as a personal toolkit for efficiency and problem-solving, rather than abstract theory.

Design Thinking Mindsets

Applying Design Thinking at a personal innovation level begins with adopting the core mindsets that underlie the methodology. DT is often described not only as a process but also as a way of thinking, embracing certain attitudes towards problems and solutions. Key mindsets include being human-centred, comfortable with ambiguity, inclined towards action and experimentation, and collaborative and iterative in approach. An individual practitioner of DT methods needs to cultivate these principles in their work:

Human-centred focus (Empathy). At its heart, Design Thinking means looking past the technical and listening to people. In a team setting, this refers to deeply understanding users or customers. For a personal innovator, a human-centred mindset involves considering the needs, perspectives, and feelings of the people who will be affected by one’s work. For example, if you are drafting a report or creating a presentation, applying empathy means stepping into the shoes of your audience or stakeholders. What do they value, what are their pain points, and what will resonate with them? By focusing on the human element of any task, individuals can uncover problems or opportunities that might be missed by a purely task-oriented approach. Empathising with others (or even with one’s own future self as the user of a new personal routine) helps ensure that solutions are relevant and impactful. Research on DT highlights that personal experience and observation can illuminate problems you may not have considered and reveal novel solutions. Thus, a consultant preparing a client proposal might, for instance, informally interview a client team member or reflect on prior client feedback (empathise) before defining the proposal strategy.

Embracing ambiguity and reframing challenges. Design Thinking accepts that many problems are messy or ill-defined initially. Instead of seeking immediate clarity, design thinkers acknowledge that the world doesn't always provide clear answers and that many things will be unclear until they're tried out. For an individual, this means not freezing in the face of a vague challenge, but rather proceeding with the best understanding available and treating each attempt as a learning opportunity. A key ingredient of this mindset is reframing, i.e., looking at a challenge from different angles to find the real problem or a more optimistic perspective. Rather than seeing a situation as a dead-end, design thinkers reframe it as an open-ended question or opportunity. In practice, an analyst who finds "I have too much data to sift through" might reframe this as "How might I identify the most important data efficiently?" thereby turning a complaint into a solvable design question. Adopting a stance of curiosity and reframing problems unlocks creativity and prevents getting stuck in a negative mindset. In their book *Designing Your Life*, Bill Burnett and Dave Evans, who together with Bernie Roth's *The Achievement Habit* pioneered applying DT to personal decisions, call out the dysfunctional belief "I'm stuck" and suggest reframing it to "I'm never stuck because I can always generate a lot of ideas". This exemplifies the individual attitude that any problem can be approached from a new angle.

Bias towards action and experimentation. Another key idea of Design Thinking is bias to action, the idea that it is often more useful to try something quickly than to over-analyse or endlessly plan in the abstract. For an individual, this translates to a willingness to experiment with ideas in small ways to see what works. Rather than striving for a perfect plan on paper, a personal design thinker will build a quick prototype, run a trial, or otherwise get their hands dirty and observe the results. This mindset is closely tied to learning from failure: recognising that a failed attempt is not wasted effort but a chance to gather feedback and improve. Negative input isn't necessarily a sign you did something wrong, it provides an opportunity to make a better solution and improve your skills in the process. In everyday work, this could mean if you have an idea for streamlining your task workflow, implement a rough version of it for a week and see what happens, rather than spending a month debating if it is the best approach. The iterative trial-and-error approach is often more illuminating. Entrepreneurial methods like *lean thinking* echo this principle with the mantra build – test – learn, which aligns well with DT. You can always start really small and do it in ways that no one can tell you not to. You should feel empowered to conduct low-risk experiments within your own sphere of control to prove concepts.

Iterative process orientation. Design Thinking also stresses iteration, recognising that one rarely arrives at the optimal solution in a single step. Solutions evolve through cycles of prototyping, feedback, and refinement. An individual who internalises this will approach their work as a continuous improvement process. For instance, rather than finalising a complex document in one go, you might iterate through an outline, a rough draft, and then several edited drafts, each time incorporating new insights. This contrasts with a one-and-done mentality and can lead to higher-quality outcomes. It requires humility (accepting that your first idea can be improved) and persistence. Iteration is facilitated by the above mindsets of reframing (since each iteration might re-examine the problem) and action (since each prototype or draft is an action-generating feedback). An iterative mindset also encourages documenting lessons learned: after trying a solution, pause to evaluate. Does it work to solve the need? How could it be improved? By capturing these answers, you formalise the learning and set the stage for the next cycle of improvement.

Holistic and systems thinking. Although Design Thinking often zooms in on specific user needs, it also benefits from zooming out to see the bigger picture, what we call *systems thinking*. For an individual, this means understanding the context and interdependencies surrounding a problem. Instead of addressing issues in isolation, a personal design thinker asks: What larger system is this part of, and could my solution have unintended side effects elsewhere? Taking a systems view guards against solving the wrong problem or implementing a fix that doesn’t stick because the root cause lies elsewhere. For example, if a consultant finds that project deliverables are often delayed, a narrow view might blame personal time management, but a systems-thinking view might reveal organisational factors (e.g. the process for approvals or the communication channels) that need redesign. A contemporary example, also in a design context, is electric cars. It seems like a great solution to reduce pollution, but if the electricity is generated by coal, as it is in many places in the world, the *overall system* might end up causing more pollution. The lesson for individuals is to consider the whole system of a problem, both the upstream and downstream factors, so that one addresses the true leverage points. Adopting a systems perspective often goes hand-in-hand with reframing the problem at an appropriate scope (zooming out to redefine what is really at issue). More on this in Chapter 3.

Collaboration and feedback-seeking. Even when not working in a team, a Design Thinking mindset values diverse perspectives and input. Design Thinking in teams relies on radical collaboration, bringing different people together to spark ideas. An individual can simulate some of this effect by actively seeking feedback or by stepping outside of their own expertise to gather ideas. In practice, this might mean showing an early prototype to a colleague or friend for critique or consulting end-users informally. It could also mean using empathy tools to incorporate the imagined perspectives of different stakeholders. The point is that one shouldn't design in a vacuum. Being open to others' insights typically improves the outcome and can catch issues early. For instance, an analyst might ask a co-worker to review a new dashboard they designed (a prototype) to ensure it is understandable before it is finalised. Additionally, collaboration could involve co-creating solutions with the people who face the problem, even if the people are just a small circle. The Design Thinking literature often states that a designer is not the user. Similarly, a personal innovator recognises the limits of their own viewpoint and checks their ideas against external reality through collaboration and feedback.

So before diving into specific techniques, an individual adopting Design Thinking should internalise these attitudes: empathise with people, embrace ambiguity and reframe problems, experiment early and often, iterate based on feedback, take a holistic view of systems and seek input from others. These mindsets form the foundation of using any Design Thinking tool effectively. With this foundation set, we now explore concrete methods and how they can be leveraged by someone working on personal or professional challenges.

Reframing a Problem

One of the most powerful aspects of Design Thinking is its emphasis on properly defining the problem before jumping to solutions. As the saying goes, a problem well stated is a problem half solved. In DT, the *Define* stage is devoted to synthesising insights and clarifying the challenge at hand. For an individual, time spent on thoughtful problem framing is an investment that can pay off enormously in clearer focus and more effective solutions. Research in creativity underscores that problem framing is pivotal in guiding the design process and influencing the effectiveness of solutions developed. It transforms vague, often misunderstood issues into clearly defined challenges that invite creative thinking and problem-solving.

Reframing is the technique of viewing a problem from a new perspective, essentially changing its frame of reference to uncover alternative interpretations and solutions. In personal applications, reframing often means challenging your initial problem statement and asking if it is the right problem to solve. Design Thinking provides a simple but effective method for this: turn statements into *How Might We* questions. By taking a problem statement and phrasing it as How might we...?, you open it up for exploration. For example, instead of saying "I need to manage my time better," you might reframe it as "How might I redesign my daily schedule to maximise my productive energy?" This subtle shift transforms a fuzzy goal into a design challenge that invites ideas focusing on *redesign* and *energy*, not just managing time. A well-framed challenge acts like a compass, it keeps your creative efforts directed.

To illustrate, consider a personal innovator who initially thinks the problem is "My team's weekly meetings are a waste of time." If taken at face value, one might jump to a solution like cancelling all meetings or blaming attendees for not focusing. But by reframing, the problem could become How might we make our team check-ins more efficient and valuable? This invites many creative approaches: changing the meeting format, frequency, or content, rather than simply eliminating meetings. The reframed question assumes there *is* value to be unlocked, i.e. meetings can be efficient and valuable, which is a more constructive starting point. As a result, the solutions generated will target improving the meeting (perhaps by tightening the agenda or moving status updates to a shared document) instead of discarding it entirely or making superficial tweaks.

Academic and professional sources alike stress the importance of proper problem-framing. Defining the context and implications of a problem leads to more innovative solutions. It is often noted that problem framing is the crux of the design process. In practical terms, spending effort to define the real challenge ensures you don't waste time solving the wrong problem. As one facilitation guide puts it, *doing well that which should not be done at all* is a common efficiency pitfall. A clear problem definition helps avoid that trap. Several practical tools can aid in reframing and defining problems:

The Five Whys. This involves asking why repeatedly to peel away layers of symptoms and reach an underlying cause. For example, if you consistently miss deadlines, you might ask: Why? – Because my task estimates are wrong. Why are they wrong? – Because unexpected urgent tasks come up. Why do those derail you? – Because I don't reschedule other work when urgent tasks come. Why? – Because I feel everything is a top priority. In this case, the root issue might be a lack of prioritisation or unrealistic workload acceptance. The real problem to solve could be reframed as How might I prioritise my workload to meet key deadlines? rather than simply to work faster or say no to urgent tasks, which might ignore that some urgent tasks are legitimate. The Five Whys method reframes the problem at a more fundamental level, often revealing that what we thought was the problem is just a symptom.

Challenge assumptions. Explicitly list the assumptions you are making about the problem or solution space, then question each. For instance, if you think "I cannot delegate this task because no one else can do it right," that's an assumption to challenge. What if someone *could* do parts of it or do it with guidance? By surfacing assumptions (for example, this task must be done by me, it must be done in this format, or even that it must be done at all), you can reframe the problem with fewer constraints. Flipping an assumption can turn a stuck problem into an open question. For example, assuming that we must have a weekly meeting can be challenged and reframed as How might we keep the team aligned with minimal disruption, which might lead to alternatives like a shared update board.

Point-of-View statements. In the classic Stanford d.school approach, designers craft a PoV statement to articulate the problem from the user's perspective, often in the format: *[User] needs a way to [need] because [insight]*. An individual can use this by thinking of themselves or their stakeholders as the user. For example: A busy project manager needs a way to get quick project updates because they often feel blindsided by issues too late. This PoV highlights the need (for quick updates) and insight (feeling blindsided, implying current updates are insufficient or delayed). From a PoV statement, you can derive one or more *How Might We* questions, such as How might we provide real-time project updates to the team? This approach forces clarity about who has the problem and why it matters, leading to a human-centred reframing of the challenge.

Visualise the problem. Sometimes drawing a simple diagram of the problem can reveal new angles. For instance, sketch a flowchart of your current process, or a mind map of the factors affecting the situation. Visualising the elements and their relationships can make gaps or misalignments obvious. A consultant mapping out why a project is delayed might draw the workflow and mark where things bottleneck. The visual might show that approvals are a major holdup, reframing the problem to How might we speed up or streamline approvals? rather than a generic work faster. Visual thinking is known to help break complex challenges into smaller pictures to see the problem differently, often removing entrenched beliefs and helping people get unstuck. By literally *seeing* the problem, you may notice aspects that you hadn't considered when it was just words.

As an example of reframing in action, imagine a manager initially defining their problem as My team isn't innovative. That's broad and somewhat accusatory. Using empathy and reframing, they might discover through conversation that team members feel afraid to take risks. So the manager reframes the problem as How might I create a team environment where people feel safe to propose and try new ideas? This reframing shifts the onus from blaming the team to an actionable leadership challenge. It leads the manager to solutions focused on culture (like rewarding experiments and explicitly saying failure is okay) rather than simply telling the team to be more innovative.

By investing effort in defining the right problem, individuals set themselves up for success in the subsequent ideation and solution phases. A well-framed problem acts as a guiding star, ensuring that creativity and effort are channelled effectively. As a result, you avoid the frustration of solving one issue only to find it didn't address the real concern. Reframing and problem definition may feel like delaying action, but it often *accelerates* progress by preventing false starts and dead ends. We will now look at each Design Thinking step in more detail.

Empathy: Understanding Needs

Empathy is the cornerstone of human-centred design, which Design Thinking is an instance of. In team-based Design Thinking, this stage involves user research: observing people, conducting interviews, and immersing in the user's experience to gather deep insights. For an individual applying Design Thinking, empathy still plays a vital role. Even if you may not conduct actual research, the principle is to ground your work in real human needs and perspectives.

Being empathetic in personal work means deliberately stepping outside of your own mindset and considering the experience of others who interact with your work or are affected by the problem. It aligns closely with the human-centred focus discussed earlier. By concentrating on people, whether they are clients, colleagues, customers, or end-users, you ensure your solution actually addresses the right problem in a way that resonates with those people, not just how *you* see the problem.

Empathise with end-users or an audience. If your work product has end-users (for example, you create reports for decision-makers, software for clients, or training sessions for employees), it is important to understand their needs, pain points, and context. An individual can do this by gathering information directly or indirectly. Direct methods include informal conversations, surveys, or simply asking for feedback. Suppose you are a data analyst preparing monthly reports for managers. Empathy might involve asking a couple of managers what they find most valuable in the report and what they struggle with. You might discover, for instance, that they are overwhelmed by too much detail and really only care about a few key metrics and insights. That insight can guide you to simplify the report. Indirectly, you can create an *empathy map* for your audience, charting what they say, think, do, and feel regarding the task or information at hand. Even without new interviews, you can hypothesise based on your knowledge: e.g., a manager might *say* "I need the data to make a decision," *think* "I hope this report highlights issues early," *do* skim the report quickly, and *feel* frustrated if she cannot quickly find actionable insights. This exercise helps you step into their world. You might realise that adding an executive summary or traffic-light indicators could address their feelings and behaviours. By centring on the user's perspective, you reorient your work to serve their needs better.

Empathy tools commonly used in design, like *personas* and *customer journey maps*, can be downscaled for individual use. If you are, say, a freelance consultant developing a new service offering, you might sketch a persona of your ideal client (their role, goals, challenges) and map out their journey in working with you (from first contact to project completion). This can highlight points where the client might feel uncertainty or lack of value, giving you clues on what to improve or emphasise in your service.

Empathise with yourself (as a user). Interestingly, you can also apply empathy inward. Sometimes *you* are effectively the user of your own processes (like when creating a personal task system). You can try to view your own experience objectively: observe your work habits and emotional reactions. In *The Achievement Habit*, Bernie Roth suggests doing empathy on yourself by paying close attention to your own behaviours and feelings as data. For example, notice which tasks you procrastinate on. This might indicate a design flaw in how you approach those tasks, such as they're too undefined or you lack a needed skill. By empathising with your future self, you can design better supports. If future-you tends to forget context when returning to a task, you can leave yourself notes (a solution born from empathy for your own limitations). If you find a particular routine draining, empathise with yourself by asking what aspect of it is causing friction or stress, and address that. Essentially, treat yourself as one of the people for whom you are designing solutions.

Self-empathy example. Imagine you want to exercise in the mornings but consistently fail to do so. Instead of just blaming yourself for laziness, try to empathise with the version of you at 6 am. That user (half-asleep you) might need certain conditions to succeed. Maybe a warm room, workout clothes laid out, a clear plan of what exercise to do, and motivating music. By understanding what early-morning-you *feels* (cold, groggy), *thinks* (just 15 more minutes of sleep…), and *does* (hits snooze), you can design a better system (set the thermostat, prepare clothes and a quick routine, and maybe find a workout buddy expecting you). This is a personal efficiency improvement achieved by empathising with your own future state rather than just issuing strict goals to yourself.

Empathise with colleagues and stakeholders. In many cases, your work outputs or changes will affect others around you, such as a boss, team members, or collaborators in other departments. Applying empathy here means viewing the situation from their perspective. If you plan to introduce a new process for your team, consider how it addresses their needs or potentially burdens them. For example, say you want to use a new project management tool to increase transparency. Empathise with a team member: they might *feel* anxious about learning a new tool and *think* it is extra work with no benefit to them. Understanding this, you can address their concerns by choosing a very user-friendly tool, providing a quick cheat sheet or training, and highlighting how it will make their life easier (e.g., fewer status meetings). Empathy thus leads you to implement changes in a more user-centred way, increasing adoption and success. In the context of a personal innovator influencing change without formal authority, this empathetic approach can be the difference between getting buy-in or facing resistance.

Empathise with those the problem affects. If you are solving a problem that affects a certain group (even if they're not users of a product per se), try to engage with them. For instance, if you are trying to improve employee engagement in your team, talk to team members to gauge their feelings about work, or at least put yourself in the shoes of a disengaged employee and ask why they might feel that way. Maybe you identify factors like a lack of recognition or unclear growth paths. That insight directs you toward appropriate solutions (like implementing shout-outs or one-on-one career chats). Without empathy, you might have guessed wrong (e.g., assumed people wanted higher pay when really they wanted acknowledgement). A practical technique often used is the *empathy interview*, a casual yet attentive conversation focusing on the other person's experiences. An individual can do mini-empathy interviews with colleagues or clients by asking open-ended questions and listening more than talking. For example, Can you walk me through how you currently handle X? or What are the biggest frustrations you face with Y? and then really listen. This can reveal nuances one wouldn't think of alone.

Observation. Empathy is not only about asking, but also about watching. People don't always articulate everything. You can learn a lot by observing behaviour. If you introduce a new template for reports and notice people still using the old one, that observation is telling you something (perhaps the new template is too cumbersome or they weren't convinced of its value). If you see team members frequently messaging each other for clarification on a process, it might indicate the process documentation isn't clear. These clues, gathered empathetically (i.e., without judging the people, but understanding something in the process is at fault), guide you to refine your solution.

A classic example of empathy in DT comes from healthcare: designers shadowing nurses during shift changes realised much vital patient information was lost or miscommunicated in hurried verbal handoffs, leading to solutions like new handoff protocols and tools. On a personal scale, you might shadow or track yourself or a colleague through a routine to discover pain points. If you find yourself repeatedly copying data between two software tools, empathy for your own frustration might spur you to create a simple automation. You observed a pain point and solved it.

In essence, empathy in personal work is about always asking: *Who is this for, and what do they really need?* When the answer is *myself*, it is about being kind but honest in assessing your own needs. When the answer is others, it is about stepping out of your expert or insider perspective and seeing it with fresh eyes. This human-centric grounding ensures that whatever solutions you develop are anchored to reality and address the core issues people face. It can save enormous time by preventing you from solving the wrong problem or delivering something in a way that doesn't resonate. Empathy makes your work more efficient, not by speeding it up but by *making it count*. A solution that truly fits the need will naturally feel like a more efficient one because it actually gets used and solves something.

Ideation Techniques

When it comes to idea generation (the *Ideate* phase of Design Thinking), people often picture a lively team brainstorming session with sticky notes and wild ideas bouncing around the room. However, solo ideation can be just as powerful. In fact, psychological research has found that individual brainstorming can produce *more* ideas than traditional group sessions because it avoids phenomena like groupthink and production blocking. Much literature on group brainstorming has actually found it to be less effective than individual brainstorming. The key for the personal innovator is to use structured techniques to break out of habitual thinking patterns. Alone, one has the freedom to explore even the craziest notions without fear of judgment, a fertile ground for creativity if leveraged well. Here are several ideation techniques and mental models that individuals can use to generate solutions:

Personal Brainstorming (with rules). The classic rules of brainstorming: defer judgment, aim for quantity, and encourage wild ideas, can be applied by yourself. Set a timer and a target number of ideas. For example, challenge yourself: *In the next 10 minutes, I will write down 20 different solutions to this problem, no matter how outlandish.* This self-imposed pressure pushes you beyond the obvious ideas. The first few might be conventional, but as you strive to hit 20, you will likely venture into unexpected territory. The key is *not to stop or evaluate* until you've hit the quota. You can do this on paper or a blank document. After completion, you can review and highlight promising ideas. This technique ensures you explore a broad space. A consultant facing a client's challenge might list everything from conservative tweaks to radical pivots, exhausting assumptions and exploring edge cases, which often produce gems that wouldn't surface otherwise.

SCAMPER. SCAMPER is a checklist-based ideation method that stands for *Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, and Reverse*. It prompts you to systematically apply each operation to your problem or current solution. For example, if you are improving a workflow. *Substitute:* Can you replace a step or tool? *Combine:* Can two steps be merged or handled together? *Adapt:* Is there something working in a different context you can borrow? *Modify:* Can you change the process's scale or duration (e.g., meet biweekly instead of weekly)? *Put* *to other use:* Can an output of the process be repurposed elsewhere (increasing its value)? *Eliminate:* What if you removed a step entirely? *Reverse:* What if you did it in the opposite order? By going through each letter, you generate a variety of ideas. For instance, applying SCAMPER to a daily routine might yield: Substitute checking email in the morning with planning the day; Combine the morning coffee break with a team huddle; Adapt a colleague's method of time-blocking; Modify (reduce) meeting lengths; Put your commute time to use by listening to industry podcasts; Eliminate multitasking during critical work; and Reverse the order you tackle tasks (hard tasks first vs last). Not every prompt gives a winner, but often one or two lead to a promising idea. SCAMPER ensures you are not reliant on sheer inspiration. It methodically nudges you into creative directions.

Mind mapping. Mind mapping is a visual brainstorming technique that leverages association. You start with a central concept and branch out with related ideas, then sub-ideas, and so on. Mind maps mimic how our brains jump from thought to thought. They are excellent for exploring all facets of a problem. For example, if you want to improve personal productivity, you might write Productivity in the middle, then branches like Physical Energy, Time Management, Focus, and Tools. Under Tools, branch further into Task List App, Calendar, Automation, etc., and under each of those, more ideas (like under Automation, you might branch into email filters, macros, and text shortcuts). This sprawling map might reveal a particular branch that's ripe with ideas (say, you end up with many ideas under Automation, indicating lots of potential there). Research suggests that mind mapping aligns with our brain's natural way of generating and connecting ideas and can lead to more creative outcomes. It is also a great way to see the whole landscape of a problem on one page, which can spur solutions that combine elements from different branches. See also analogy maps in Chapter 2 for a different use of these maps.

Analogies and metaphors. Sometimes thinking *within* the domain of your problem limits you. A powerful ideation strategy is to seek analogies from completely different domains. Ask, What is this problem like? For instance, if you are trying to streamline information flow in your team, you might ask: *How do ants or bees communicate and organise their colony?* This might suggest ideas about simple signals or decentralised control. Or if you need to design a training program, you might ask:

How do games keep players engaged and learning? yielding ideas like levels, instant feedback, and challenges. By finding a parallel in a different field, you can transfer insights back to your context. Many famous innovations have come from analogies. For example, Velcro was inspired by burrs clinging to dog fur. As an individual, you can do this intentionally: list things or systems the problem reminds you of, and then examine how those systems work or solve similar issues. This often produces fresh perspectives that straightforward brainstorming might miss.

Random stimuli. A more playful method is to introduce a random element and force a connection. Open a dictionary or a random web page and take a word or image. Then challenge yourself to connect it to your problem. For example, you see the word bridge. How can *bridge* relate to your challenge of team communication? It prompts thoughts like bridging between people. Maybe you need a liaison role or a shared platform that bridges two departments. The randomness can jolt you out of your usual patterns. This technique (sometimes called *random entry* or *provocation*) can yield silly connections, but also some creative leaps. The key is not to dismiss even bizarre associations immediately; explore them a bit to see if they spark an idea. You might ask *In what ways is my problem like [random object]?* Even if the comparison is forced, it might highlight aspects of the problem you hadn't considered. This technique is related to start reversed, in which you think of how to worsen the situation as much as possible and then take the opposite as a good idea.

Challenging constraints (What if…?). Another ideation approach is to temporarily remove or alter constraints and see what ideas emerge. Ask extreme questions like: What if I had to solve this with zero budget? or What if I only had one day to implement a solution? or the opposite, What if money/time were no object? These scenarios push your thinking. For instance, if zero budget, you might lean on open-source tools or reuse existing resources, perhaps discovering a frugal solution that is actually quite good. If unlimited resources, you might imagine an ideal solution and then scale it down to something feasible (this often helps clarify the key features that provide value). By altering constraints, you free yourself to explore solutions that the original constraints might have discouraged you from considering. Later, you can adjust for reality.

Brainwriting and idea remixing. If you have a list of ideas (from brainstorming or SCAMPER sessions), you can do a second round of ideation by combining or tweaking those ideas. Take Idea A and Idea B and imagine them implemented together. Does that solve more of the problem or create a new concept? For example, from a brainstorming list on improving meetings, you have stand-up meetings and pre-recorded video updates as separate ideas. Combining them could yield an approach where routine updates are pre-recorded (as video or memos) and the meeting itself is a short stand-up focused only on Q&A or issues. Often, initial ideas can be components that mix into a stronger solution. You can also improve one idea by applying another to it (like using SCAMPER on a brainstormed idea to iterate it further).

After generating numerous ideas, an individual should switch to a convergent mindset: evaluate and select the most promising ideas to prototype or develop. Techniques like setting simple criteria (impact and feasibility), or doing a quick *pros/cons* or *SWOT analysis* on top ideas, can help identify which one(s) to pursue. Since you may not have a team to vote or critique, you could involve a trusted colleague or mentor at this stage to sanity-check your choices.

Example: Personal ideation for a work challenge. Sarah is a knowledge analyst frustrated with how her team shares research findings. Currently, they email long reports that few people read. She defines the problem as needing a better way to share key insights quickly. Alone at her desk, Sarah decides to ideate. She uses a mind map, starting with Share Insights at the centre. Branches include Format (under which she lists ideas like infographics, one-page summary, video summary, and interactive dashboard), Timing (e.g., real-time Slack updates, weekly digest, and on-demand database), and Audience Interaction (e.g., Q&A sessions, internal newsletter, lunch-and-learn presentations). Next, she uses SCAMPER on the current practice of emailing reports. *Substitute:* replace reports with infographics. *Combine:* combine a report with a meeting (email report + quick meeting). *Adapt:* adopt a style from social media (perhaps a short blog-style post for each insight). *Eliminate:* no more attachments, just bullet points in the email body. *Reverse:* instead of pushing (email out), use pull, a repository where people can find insights when needed. This generates many ideas. She then picks a few intriguing ones: creating an internal blog for research findings, making a one-slide visual summary for each report, and hosting a monthly call. She quickly jots down the pros/cons for each. The blog idea, while modern, might require people to remember to check it. The one-slide summary is easy and could be attached to emails or posted on the office wall. The monthly call might reach only those who join. She decides the one-slide summary is worth prototyping first (it is simple and directly addresses making insights quick to grasp). She also plans a small experiment with the blog idea (maybe cross-post the content to the new internal blog) to see if it gains traction. In this ideation process, Sarah, a personal innovator, has generated and filtered a robust set of options, far beyond the initial maybe-we-need-shorter-reports thought. By applying structured creativity methods, she found a solution that is both creative and practical.

So individual ideation is about diverging freely and widely, and then converging thoughtfully. Techniques like solo brainstorming with quotas, SCAMPER, mind mapping, and analogy thinking are readily accessible and can dramatically expand your solution space. One of the advantages of working alone is that you have full freedom to explore wild ideas. Use that to your benefit by indeed letting some wild ideas out. You can always rein them in later. By equipping yourself with these creativity tools, you ensure that when it is time to solve a problem, you won’t be stuck with just the first idea that comes to mind (which is often the same idea everyone else might have). Instead, you will have a range of innovative possibilities to choose from, making it more likely that you land on an effective and efficient solution.

Prototyping and Experimentation

Design Thinking is a very hands-on approach, and nowhere is that more evident than in the prototyping phase. Prototyping means creating a tangible (or at least experiential) representation of your idea so that it can be tested and improved. In product design, this might be a physical model or a beta version of software. In the context of personal or work process improvement, a prototype can be thought of as a *trial run* or a *mock-up* of a new approach. The mantra is *build to think*. By making something concrete, you push the idea forward, reveal flaws, and gather feedback far faster than if you just discussed it in theory. For an individual, prototyping has a slightly different flavour than for a design team, but the principles remain:

Make a draft or model of the solution. If your idea is a new way of doing something, *try it on a small scale*. For example, if you want to implement a new weekly report format, create one example report (perhaps taking an old report and reformatting it) as a prototype and see how it looks or how others react. If your idea is a new scheduling system for yourself, prototype it by applying it for just one week as an experiment. If you are planning a new presentation style, make a rough storyboard or slide deck outline and show it to someone for quick feedback. The prototype should be incomplete and low-cost by design, It is not the final product, but a means to learn. The Interaction Design Foundation notes that prototypes are often early, inexpensive, and scaled-down versions of the solution, made to test ideas and reveal issues.

Keep it low-fidelity initially. One mistake is trying to perfect the prototype. The goal is to learn, not to impress. If you spend too much time polishing, you become attached and also waste time if it turns out you need a different approach. Instead, aim for a prototype that is just realistic enough to get valid feedback on the core idea. For instance, if you are prototyping a new data dashboard for your team, you could sketch it on paper or create a dummy version with mock-up data in PowerPoint or Figma. That might be enough to show colleagues and see if they understand it and find it useful. You don’t need to code a whole app to test the concept. In knowledge work, prototypes can take forms like outlines, storyboards, role-play scenarios, sample outputs, or pilot implementations. If your idea is a new meeting format, prototype it by running the next meeting in that format as a trial. If it is a new personal habit, treat the first week as a prototype where you journal what happens. There are two distinct reasons for the low-fi model. The first is that you will get feedback on the actual functions you want to prototype, not many of the others that look neat but obfuscate the core idea under scrutinisation. The second is that you are much more likely to receive negative feedback when such is called for if the giver can tell that you did not spend a very long time and put your soul into the prototype.

Focus prototypes on the uncertainties. Ask yourself, What do I need to learn or validate with this idea? Then design the prototype to test that. For example, if your idea for a team knowledge base depends on people actually contributing to it, your prototype might be setting up a bare-bones wiki and inviting a few entries to see if colleagues participate. The outcome will tell you if your assumption about participation holds, and you can adjust accordingly (maybe you learn they need incentives or easier templates). If your uncertainty is whether an audience will understand a new presentation format, prototype by presenting a portion to a small audience and asking if they got the point. By prototyping the riskiest or most uncertain aspects of an idea, you de-risk the implementation.

Embrace the experimental mindset. Treat each prototype as an experiment. Enter it with a hypothesis: *I believe that doing X will result in Y*. For instance, "I believe that sending a one-page infographic instead of a full report will result in managers being more likely to read and recall the information." Your prototype (sending an infographic one week) will test that hypothesis. Maybe you will measure recall by asking a question later or gauge response by whether they follow up with fewer questions. Viewing it this way makes it clear that if Y doesn't happen, X might not be the right solution, and that's valuable learning too. This approach is essentially a light version of the scientific method (see Chapter 5) applied to your innovations, and it ties nicely with the build – test – learn cycle from lean methodologies.

Multiple prototypes and parallel ideas. If feasible, you can prototype more than one idea on a small scale to compare. For personal workflow changes, maybe you try Method A this week and Method B next week, then decide which felt better. For deliverables, you could make two versions of a slide and see which communicates more clearly by getting a colleague's opinion (A-B testing in a sense). Personal innovators have the flexibility to quickly do multiple prototypes because it is often about altering your own approach slightly. For example, to improve remote meetings, you might prototype two different facilitation techniques in two separate meetings and see which yields more engagement.

After prototyping comes testing, often overlapping as you observe your prototype in action. As you test, stay open-minded. If a prototype fails in the sense that it didn't produce the desired result, it is not a failure of you or even of the idea, but a success in terms of learning. Maybe the idea needs tweaking, or maybe you discover a new insight about the problem. Prototyping in a personal context should be a quick, low-stakes exercise. You can easily discard what doesn't work and try another approach.

Example: Rapid prototyping a personal productivity hack. Antonio has the idea to implement a two-hour focus block every morning to tackle deep work because he suspects constant email checking is hurting his efficiency. Instead of overhauling his schedule permanently, he prototypes this idea for one week. He sets an expectation with himself (and tells his co-workers in case they wonder) that he will be offline from 9–11 AM, focusing on a key task each day. This is his prototype of the new routine. He also keeps a quick journal during that week, noting what he accomplished, how difficult it was to ignore email, any emergencies missed, and his subjective focus level. At the end of the week, he reviews: he finds he accomplished significantly more on his core projects and nothing bad happened with email (plus he felt less stressed). The prototype was successful. However, he also noted that one client really wanted a response faster one morning. So for the next iteration, he decides to include a 5-minute email scan at 10 AM just to catch anything truly urgent, then extend focus time to 11:05. He tries that the following week (a minor prototype tweak). This still preserves the majority of his focus block but addresses the concern. Over these two iterations, Antonio validated his hypothesis that a morning focus block would boost his productivity and identified a small adjustment to make it sustainable. Now he formally blocks his calendar every day for focused work. If the initial experiment had shown problems (say he couldn’t resist checking email or he found his energy was low in the morning), he would have learned that without much loss and could then prototype a different approach (maybe an afternoon focus block, or use an app to lock email). This example shows how prototyping a change rather than just implementing it full-scale or debating it endlessly allowed quick discovery of its value and any needed changes. The cost of the prototype was essentially zero (just one week of trying a new schedule), but the insight gained was high.

In a team or deliverable context, consider another example: A manager wants to implement a new project update template for her team to use in meetings. Rather than forcing it on everyone immediately, she prototypes it by using the template herself in the next update meeting and maybe asking one willing team member to try it too. After the meeting, she asks the team for feedback: Did the template make updates clearer? Did it miss any important information? Perhaps the team says it was clearer, but they suggest adding a section for risks. The manager can then iterate the template and confidently roll it out, having tested it in a low-pressure way.

The prototyping stage, when done in an agile, small-scale fashion, prevents analysis paralysis because it encourages moving forward with something tangible. It also mitigates the risk of big failures because you catch issues early. It is a way of *failing fast but cheaply*. If an idea isn't going to fly, it is far better to learn that via a one-day prototype than after six months of effort.

For a personal innovator, prototyping also has a motivational benefit: it gets you doing something. Often when trying to change habits or implement improvements, the hardest part is starting. By telling yourself "It is just a test" or "This is just a pilot run," you reduce the psychological barrier. It is not a permanent commitment, it is an experiment and you will evaluate after. This can trick your brain into cooperating with change more readily. So, prototyping for personal and professional efficiency means *acting out your idea in miniature* to see if it holds water. It converts abstract ideas into lived experiences from which you can learn. By doing so, you transform uncertainty into knowledge and inch closer to a solution that truly works, with minimal wasted effort.

Testing and Iteration

Prototyping and testing are two sides of the same coin. When you prototype something (a new method, a draft deliverable, a pilot process), you must also *test* it, meaning you observe it in action, gather feedback, and evaluate its effectiveness. For a personal innovator, this testing phase might be informal and quick, but it is critical. It is where you close the loop and ensure the idea actually solves the problem. Testing turns guesses into knowledge. Key aspects of testing and iteration in a personal context include:

Gather feedback and data. Depending on what you are testing, feedback could be your own observations, metrics, or input from others. If your prototype was a new personal routine, your data might be how you felt, what you accomplished, or even quantitative measures, such as the number of Pomodoro (time-boxed) cycles completed, or your sleep quality if it is about a sleep habit. If your prototype was something delivered to others (like a new report format), then definitely ask those others for feedback. Did they find the new report easier to understand? Do they prefer it? Sometimes, just one or two targeted questions can yield actionable feedback. For example, after sending an infographic-style report, you might ask a manager: *Did you get what you needed from that format?* If the answer is lukewarm, ask what was missing or could be improved. People are often happy to give feedback when they know you are experimenting to better meet their needs (it shows you care about their experience). In the absence of direct feedback, observation is your friend: Did the recipient actually open the email with the new report? Did meeting participants engage more or less with your new meeting format? These outcomes give clues to success.

Be objective and avoid ego. When testing something you created or a change you initiated, it is natural to be biased or defensive. Design Thinking encourages a *learning* mindset. View the prototype as separate from yourself. If it fails, it is not you failing; it is just an idea that didn't work *in that form*. Treat each result as useful information. For example, if your attempt to delegate more tasks leads to confusion, that's the feedback that maybe the delegation approach needs clarity, not that delegation doesn't work or that you are bad at managing. This ties to the concept of *failing forward*: each iteration, even if not perfect, is a step forward in knowledge. By being objective, you will iterate more effectively.

Iterate based on insights. After testing, incorporate what you've learned into a new iteration of the idea. This might mean tweaking the prototype and testing again or making a decision between alternatives. Iteration can be very quick. For example, if on Day 1 of trying a new email schedule you find you missed an important message, on Day 2 you adjust the schedule (perhaps check email briefly mid-morning). You don't have to wait a long time to iterate; continuous adjustment is good as long as you give each change a fair observation. In other cases, you might complete a full prototype test cycle (say a week or a project) and then plan a 2.0 version. The cycle of prototype – test – refine – test again can be repeated until you are satisfied that the solution is working well or until you decide to pivot to a different idea.

Know when to pivot or stop. Part of iteration is also knowing when an idea isn't salvageable. The beauty of prototyping is that you invested little, so if the tests are consistently negative, you can comfortably drop the idea. Maybe your test indicates the problem is different from what you thought, requiring a reframe. In Design Thinking, if tests invalidate your approach, you either iterate (change the approach and test again) or pivot (take a different approach altogether, potentially going back to ideation for a new idea). This is not a setback; it is progress because you ruled out something that doesn't work, and you get a better understanding for the next attempt. As a personal innovator, it is important to be honest at this stage. Sometimes we get attached to an idea and keep tweaking it when evidence suggests it is fundamentally not hitting the mark. Iteration is not about endless micro-adjustments to save a flawed concept; it is about refining good concepts and identifying when a new concept is needed.

Scale up gradually. Once an idea has gone through a couple of successful iterations on a small scale, you can implement it more broadly or permanently. Essentially, your prototype graduates to a new standard. Even then, keep an iterative mindset. Conditions change, and what works now might need further tweaking later. For instance, your new meeting format works well for your current team of five, but if the team grows to 15, you might need to iterate again. Design Thinking is a continuous loop, not a one-time sequence. Many individuals do this intuitively: they adopt a new productivity app, and later find one feature doesn't suit them, so they adapt their workflow or try a plugin. That's iteration in action.

Document learnings. It is useful to jot down what you learned from each experiment, successful or not. This builds your personal knowledge base. For example, you might note in your diary: Tried using Pomodoro technique, learned it works well in mornings but not after lunch (energy dips) … maybe use a different technique in afternoons. These reflections will guide future self-design efforts. Also, if you ever share these processes with others or train someone, you have concrete experiences to draw on.

Example: Iterating a team process. Let us say as a team lead, you prototyped a new code review process in your software team by using a checklist to ensure quality. On the first project where you test it, you notice it catches many issues (good), but developers complain it is too time-consuming (bad). In a retrospective (test/feedback phase), you gather suggestions: perhaps some checklist items are low-value and could be trimmed. You iterate the checklist, removing or automating a few steps. Next project, you try again. This time the developers are happier, and code quality is still high. You decide to adopt the checklist as a standard. But a few months later, new types of issues are appearing (maybe related to a new technology the team started using), so you add a new item to the checklist. This ongoing refinement is iteration at work. The process is never static, and you continuously test its effectiveness.

Example: Personal habit iteration. You attempt a new habit of reading for 30 minutes every night to broaden your knowledge (prototype). After two weeks (test period), you find you often skip it due to fatigue. Reflecting, you realise the *timing* might be the issue, not the habit itself, but you are too tired at night. So you iterate the plan: now you try reading 30 minutes in the morning instead (next prototype). This test goes better; you manage to do it most days and feel more alert while reading. However, you notice your mornings are now a bit rushed. So iteration 2: you start waking up 15 minutes earlier and sometimes split reading into two 15-minute blocks (morning and lunch). After adjusting, you find a sustainable rhythm. You learned through testing that *when* you do a habit can be as important as *what* the habit is. Through iteration, you found a way to integrate reading into your life efficiently, whereas if you had stubbornly stuck to the initial plan (even as it failed), you might have abandoned the goal entirely.

Through cycles of testing and iteration, individuals ensure that their bright ideas actually translate to positive outcomes. This stage is where the rubber meets the road: you validate whether your understanding of the problem and your proposed solution are correct. If not, you gain valuable insight to guide the next attempt. Iteration essentially reduces wasted effort in the long run. It is better to spend a little time adjusting and improving a solution than a lot of time implementing a sub-par solution widely. Another benefit is that iteration builds resilience and continuous improvement into your working style. Rather than seeing a project or method as done and then being blind to its issues, you maintain a mindset that everything can be improved. This doesn’t mean you never finish anything; it means you are always open to making things better when evidence suggests they could be. Over a career, this can compound into significant skill and efficiency gains, as you steadily refine how you operate.

In conclusion of the iteration mindset: treat your workflows and solutions like software in beta, always gathering user (or self) feedback and updating. This ensures relevance and effectiveness. It also makes your work more engaging, as you are actively learning and evolving rather than passively executing routines. Design Thinking at the personal level thrives on this principle of iterative refinement. By the time you settle on a final solution, it will have been test-driven and tuned, giving you confidence that it truly addresses the need in the best way you've discovered so far.

Practical Examples

To bring together these concepts of mindsets, methods, and iterative development, let us examine a couple of scenarios where a personal innovator applies Design Thinking techniques to solve everyday work challenges. These examples show how the stages of Design Thinking flow into each other in practice, and how tools can be combined.

Example 1: Redesigning Team Meetings as a Personal Innovator

Context: Maria is a project coordinator who leads a weekly team meeting. Lately, the meetings have run over time and felt unproductive. People seem disengaged, and the important discussions often get rushed.

Empathise: Maria starts by gathering perspectives. She chats one-on-one with a few team members (stakeholders) to ask how they feel about the weekly meeting. She also observes behaviours during the meeting: people checking phones, certain quieter members never speaking, and discussions veering off-topic. From these, she infers that attendees *feel* the meeting is a time sink, and some *think* their time could be used better. From a manager's perspective, she knows the meeting's purpose is to align the team, but it is not serving that need well. She also puts herself in each attendee's shoes (creating a persona for a busy team member). Such a person likely wants meetings to be concise and relevant, and to leave with clear action items.

Define: Based on empathy, Maria defines the problem as: Our 60-minute weekly meeting is trying to do too many things (status updates, issue resolution, planning) and is losing effectiveness; the team needs a more efficient way to align and address issues. She reframes the vague complaint of unproductive meetings into a concrete challenge: *How might we restructure the weekly team meeting to maximise value and minimise waste of time?* She sets a success criterion: team members should leave the meeting feeling it was a good use of time, and the meeting should reliably finish on time or early.

Ideate: Maria generates ideas. She uses brainstorming and writes down everything that comes to mind: shorten the meeting to 30 minutes, have a strict agenda timer, split the meeting into two separate meetings (one for status updates, one for problem-solving), eliminate the meeting and replace with an email update, use a round-robin format where each person speaks for 2 minutes, etc. She also applies SCAMPER: *Substitute* open discussion with a structured update round; *Combine* the meeting with a Monday morning huddle so it replaces two meetings; *Eliminate* in-meeting status updates by collecting them beforehand in a shared document; *Reverse* the order of the agenda (start with issues instead of status). From this, a few promising ideas emerge: (1) Make it a stand-up meeting (everyone literally stands, to keep it short), focused only on blockers and decisions, with status updates handled via a shared document ahead of time. (2) Split the meeting: a 15-minute check-in for quick updates, and a separate optional 30-minute deep dive for those who need to resolve issues. (3) Use an asynchronous update (like a Slack channel or email) and convert the meeting entirely into a bi-weekly brainstorming session instead of weekly. She evaluates these against her problem statement and decides to try idea (1) first, as it directly addresses efficiency and still keeps everyone in the loop.

Prototype: Maria prototypes the new format on a trial basis for the next meeting. She announces to the team: Let us experiment with a new meeting style this week. Please fill out this brief status update form before the meeting. In the meeting, we’ll all stand and only discuss blockers or questions for 15 minutes. She prepares a simple Google form for status updates (asking each member: What did you do last week? Plans for this week? Any blockers?). She also prepares an agenda listing the blockers reported to ensure they get addressed. This is the prototype of her redesigned meeting. It is essentially a pilot run of the new format.

Test: During the meeting, she notes the dynamics: the stand-up format does make people more brisk. The meeting finishes in 20 minutes, a drastic improvement. However, she notices a downside: two members had blockers that required longer discussions, but everyone was standing and itching to leave, so they postponed those discussions. After the meeting, she asks a few team members for feedback. One says they loved the shorter format. Another says it felt a bit too stiff and rushed, and that they missed the open discussion where sometimes new ideas surfaced. Maria realises that while efficiency went up, some collaborative problem-solving might have been lost.

Iterate: Taking the feedback, Maria tweaks the format for the next week (iteration 2). She decides on a hybrid: a 10-minute stand-up for quick updates and blocker announcements, followed immediately by a 20-minute sit-down discussion for one or two big issues (only if needed). She makes sure that if no big issue exists, the discussion can be skipped and everyone gets time back. She prototypes this the following week. In testing, this format hits a sweet spot: the quick part keeps updates efficient, and the short discussion section allows one deep-dive topic to get attention. The meeting still ends at or before 30 minutes. The team feedback this time is positive. They felt heard on their issues and still appreciated the brevity. Maria has effectively *designed* a new meeting structure through empathy, ideation, prototyping, and iteration.

Over the next weeks, Maria will implement this as the new standard. She remains open to adjusting as needed (for example, if multiple deep-dive topics arise, she might schedule an extra session rather than derail the main meeting). By applying Design Thinking, she transformed a dragging weekly meeting into a high-value touch point, improving team morale and efficiency. Importantly, she did this by involving team members (empathetically) and by safely experimenting, rather than by top-down decree.

Example 2: Creating a Personal Knowledge-Sharing Toolkit

Context: John is an independent consultant who produces reports and insights for different clients. He realises that he often reinvents the wheel when writing reports, and many contain similar research or explanations. He wants to create a personal toolkit (a knowledge base of slides, visuals, and data points) to reuse content and work more efficiently.

Empathise: Although John is both the creator and user of this toolkit, he practices empathy by considering his future self and the variation in his client's needs. He notes pain points: he often *feels* stressed on tight deadlines when he has to dig up info he knows he used before, and he *observes* that he spends a lot of time formatting similar charts repeatedly. He also empathises with his clients' perspective: they *need* clarity and customisation in the reports. If he over-systematises (just copy-pastes old material), he might risk not tailoring it to the client. So he must balance efficiency with personalisation. He defines the user needs for his toolkit: it should save him time (quickly find and reuse previous content), but also be flexible enough to adapt to each new project's context.

Define: John frames the problem as: How might I create a personal knowledge repository that speeds up report writing while allowing easy customisation for each client? His success criteria include: reducing writing time by, say, 30%, and maintaining or improving client satisfaction with reports (so quality doesn't drop). He specifically notes that the repository should be easy to search and update, otherwise it might become a burden itself.

Ideate: John brainstorms different solutions: a simple folder of past reports to draw from, a more structured database of insights (tagged by topic), creating modular slides or text blocks that can be assembled like LEGO pieces for a report, using a note-taking app (like Evernote or Notion) to store snippets and sources, or even training a personal AI on his past work to retrieve relevant content. He sketches out a concept of a report template where common sections are pre-filled with placeholders for client-specific info. He also thinks of analogies: how do chefs reuse ingredients for different recipes? They have base sauces that they then tweak with spices per dish. Similarly, he could have base analysis sections ready to spice up with client data. From his ideation, he favours creating a digital library of reusable content blocks (e.g., paragraphs of analysis, charts, case examples), tagged by theme (market trends, consumer behaviour, regulatory info, etc.), because it directly addresses reusing research. He also likes the idea of a modular template for reports.

Prototype: John decides to prototype on his next report. Instead of writing from scratch, he invests a couple of hours upfront to comb through five recent similar reports and copy out the relevant pieces of content into a single document (this is a mini repository prototype). He organises it by headings (e.g., Industry Overview, Competitive Landscape, Recommendations) and under each puts bullet points or sentences he is used before that could apply. Some of them, he edits to be more generic. Now he uses this cheat sheet to assemble the new report, tailoring each piece as needed. Essentially, the prototype repository is just a document with reusable content. He also uses a simple template structure for the report based on common sections he always includes.

Test: As he writes the new client report using the prototype toolkit, he tracks his time. He finds he completed the draft in, say, 12 hours instead of the usual 16, a notable improvement. Quality-wise, he ensures to tweak the tone and specifics in each section so it reads as a coherent, customised report. To test the quality, he might ask a colleague to skim it to see if it feels pieced together or not (the colleague finds it consistent). The client's feedback on the report comes back positive. They didn’t notice any difference in quality (in fact, they appreciated the thorough background John provided, which was easier for him since he reused a well-crafted background from a previous report, a typical consultant trick also in high-profile consultancies). However, John observed some friction in using his content document. It was a bit cumbersome to scroll through and find the right pieces, and not everything was easily adaptable. For example, a chart he wanted to reuse had data specific to an old client, and replacing it with new data was almost as much work as making a new chart.

Iterate: After this test run, John refines his approach. He decides to use a better tool for the repository: he imports those content pieces into a note-taking app that allows tagging and searching. He breaks them into smaller chunks so that each insight or graphic is an individual note with tags like market trends or financial chart. He also realises he should standardise some figures. For instance, create a generic chart template where he can just plug in new numbers easily. In iteration 2, he sets up this repository in Notion with key sections and tags. For the next project, he tests this more structured repository. Searching by tag or keyword makes it much faster to find relevant bits (e.g., he types "market size graph" and finds a ready-made chart format). Again, he measures time and finds another improvement. He continues to iterate by adding new content to the repository after each project (making it a living knowledge base) and discarding outdated information.

Over a few projects, John has built a robust personal knowledge toolkit. In one instance, he even managed to deliver a first draft to a client in record time, impressing them with his responsiveness (while in truth he had a lot of the groundwork done via his repository). He keeps an eye to ensure he doesn't fall into the trap of being too formulaic. Empathy for the client reminds him to always frame the content in the client's context. However, the heavy lifting of background research and standard analysis is largely alleviated by reusing content. This example shows Design Thinking applied to a personal workflow: John empathised (with himself and clients), defined the problem clearly (need reuse without losing quality), ideated many approaches, prototyped a basic solution (a content document and template), tested and measured it, and iterated to a better system (tagged repository and refined templates).

Through these two examples, we see how a personal innovator can cycle through empathy, definition, ideation, prototyping, testing, and iteration to tackle both team-related and personal workflow challenges. The *methods* were varied: Maria used in-person observation and an agenda prototype; John used a content library and time measurement, but the *approach* was fundamentally the same. They treated their problems as design challenges, used creativity and user-centric thinking to develop solutions, and tried those solutions in practice with a willingness to learn and adapt. These stories underscore a key point: Design Thinking for personal efficiency is not an abstract theory but a very practical, do-it-now approach. You don't need formal authority or big resources to do it. Maria didn't wait for a corporate initiative to improve meetings, she tried something with her team the next week. John began with a Word document. The tools and mindset became accessible to anyone.

The Double Double Diamond

The evolution of Design Thinking has brought with it a number of methodological refinements, yet few have captured the spirit of iterative insight and practical rigour as completely as the double double diamond model, developed and practised by Openlab Stockholm, the innovation arena for societal innovations where the author works. This model, called D3 in this book, represents a deliberate deepening of the well-established double diamond process originally articulated by the British Design Council and which is very similar to the Stanford d.school process. It is intended for large projects, where the problem and thus upcoming innovation has many facets and a rich complexity in various dimensions. By executing the full double diamond sequence twice, the first time to clarify and test the direction of innovation and then the second time to develop and test specific solutions, Openlab Stockholm has formulated a framework that balances conceptual soundness with operational excellence. The first diamond is focused on doing the right thing. This cycle is not merely an initial exploration but a disciplined process of divergence and convergence designed to interrogate assumptions, reframe challenges, and validate that the process is going in a meaningful direction. Rather than jumping too quickly into creating solutions, this first double diamond sequence allows the process (team or individual) to immerse in the complexity of the problem, synthesise insights across stakeholder groups, and determine what truly constitutes the underlying problem. It ends not with a solution but with a validated problem framing (point of view) and one or more articulated solution directions (called concepts) that have sustained initial conceptual testing. The aim is not to test solutions but to test the directions on which those solutions would later be built.

Having completed this first round, the process then enters the second double diamond in D3. This is where the focus shifts from doing the right thing to doing the thing right. Armed with a well-justified solution direction, the person or team now returns to divergence, exploring a range of solution spaces, often more freely and confidently than would be possible without the clarity established in the first cycle. Because the conceptual foundations have already been tested, the design team is now better positioned to push boundaries within a defined scope. Ideas in this second double diamond tend to be more grounded and to the point, more precisely aligned with user and stakeholder needs, and on the way to becoming feasible to implement. As with the traditional double diamond, the second cycle in D3 continues through to prototyping and testing, but now with a higher level of maturity and sharper criteria for success. The discipline of iterating twice through empathy, definition, ideation, prototyping, and testing means that both the why and the how of the final solution are interrogated and refined.

The advantages of D3 lie not in its complexity but in its structured process. It formalises a pace that design teams often would follow intuitively but do not always have the time to pursue systematically. Many innovation efforts fail not because of poor execution but because they were solving the wrong problem in the first place. Others fail by remaining trapped in abstraction, never progressing to practical implementation. While already the traditional double diamond addresses this, Openlab Stockholm has found that cycling twice through the double diamond often leads to much more well-grounded solutions to large societal challenges. D3 ensures that the first risk is addressed in full before moving into the second domain. It avoids the trap of rushing to build without reflection, while also avoiding the paralysis that can come from endless conceptual refinement. For smaller or less complex problems, however, the traditional double diamond suffices. Doing that or D3 can be decided on the fly when a standard Design Thinking innovation process starts to feel that it cannot embrace the whole problem it set out to solve because of its multifacetedness.

From a personal development or individual design perspective, the model has equally powerful applications. When tackling a major challenge, be it improving a workplace process or reshaping a personal life habit, there is immense value in first asking whether one is even looking in the right direction. The first double diamond offers a structure for that self-inquiry. Only once that direction has been validated does it make sense to begin detailing the solution. In a world where time is often short and pressure to deliver is high, the D3 model offers a deliberate alternative: pause first to examine the surroundings, then proceed with intent.

2. More Thinking Tools

By applying Design Thinking in the ways of the previous chapter, the personal innovator can continuously improve their effectiveness. Small inefficiencies that most people simply tolerate can be opportunities for creative redesign. Over time, this can significantly enhance productivity, creativity, and even enjoyment in one's work or personal life. The individual becomes a *designer* of their processes, documents, and collaborations, leading to outcomes that are not just more efficient, but often more satisfying and innovative as well. Each success (such as a meeting that runs better or a report done faster) also builds confidence to tackle the next challenge in a design-thinking way, creating a positive feedback loop of personal growth and improved performance. The Design Thinking methods from Chapter 1 are a good basic toolset. However, we will in this chapter look at four more tools and techniques that work well together with Design Thinking and together make up a stronger overall toolset: analogy mapping, value-focused thinking, core process mapping and pretendotyping. We begin with a technique for the idea generation stage.

Analogy Mapping

Mind maps are among the most widely promoted tools in creativity and self-development contexts. Marketed as a simple yet powerful way to structure thoughts, they have become a go-to recommendation for students, managers, and professionals seeking clarity in complexity. Many popular books showcase diagramming techniques, templates, and case studies that promise enhanced memory, creativity, and understanding through radial diagrams with central nodes and branching subtopics. The appeal is undeniable: the visual form is non-linear, colourful, and seemingly intuitive. Yet despite their popularity, mind maps often fail to deliver what they promise in the context of creative processes. This failure is not due to technical limitations but to a fundamental misconception: that a mind map structures knowledge. In reality, a mind map reflects the structure already present in a person's thinking. For those who possess a well-organised mental model of the domain in question, mind mapping can be a useful exercise in documenting or recalling that structure. But for everyone else, the overwhelming majority, the mind map becomes a mirror of confusion. It captures disjointed thoughts, fragments of associations, and premature hierarchies without helping to resolve them.

The result is often a visually appealing but cognitively hollow artefact. The central term is arbitrarily chosen. Branches multiply rapidly, but their relationship to one another remains ambiguous. Subtopics proliferate not because they cohere, but because the user feels compelled to fill the space. The user ends up with a colourful tangle of semi-related terms. A reflection of their pre-existing uncertainty, not a remedy for it. This is the mind map illusion: the belief that placing concepts on a page in a radial format somehow produces clarity or structure. However, insight and clarity cannot be imposed top-down through diagrams. It must emerge from deeper conceptual realisations and connections. Mind mapping fails to facilitate this emergence. Instead, it often encourages the user to confuse collection with connection, to list things without understanding how or why they relate. For the personal innovator, this is a lost opportunity. When faced with ambiguity, pressure, or innovation challenges, traditional mind mapping does not guide thinking. Rather, it documents mental noise and confusion. In such moments, what is needed is not organisation, but provocation. Not memory, but transformation. This is where analogy mapping emerges as a superior usage of the mind-mapping technique.

An analogy map takes the same basic notation as a mind map, a central node with branches and sub-branches, but reverses the purpose. Instead of trying to categorise or sort ideas into logical silos, analogy maps seek to cross domains, break categories, and find unexpected connections. The map is not a representation of what you already know, but a medium for discovering what you don’t. At its core, analogy mapping works through structured analogy chains. Rather than starting with a central concept and listing attributes or subcategories, you ask: What is this like? Each branch becomes an analogy: A is like B in the following respect. From B, you ask again: What is B like, and in what respect? This chaining leads not to a hierarchical structure but to semantic drift, a deliberate movement across contexts. The result is not neat and is not intended to be, but it is fertile.

Analogical reasoning is among the most powerful cognitive tools available for idea generation. Research in creativity and cognitive science, for example by

Gentner and Holyoak, shows that analogies enable people to transfer knowledge from one domain to another, revealing patterns, gaps, and possibilities that were previously invisible. Analogies drive innovation precisely because they violate the boundaries imposed by domain-specific thinking. Analogy maps exploit this principle. The messiness is a feature, not a problem. Where a mind map attempts to tidy thoughts into branches, an analogy map disrupts the branches by introducing foreign material. From data dashboards, one might go to cockpit controls, and from there to air traffic coordination, and from there to emergency triage rooms. Each jump adds conceptual bridges. This tension is a birthplace of innovation.

Importantly, analogy mapping involves not just naming the analogy, but specifying the dimension of similarity. A is like B *in terms of feedback loops*. B is like C *in terms of visual language*. This tangent-tracing builds multi-dimensional conceptual scaffolds. You are not just free-associating, you are tracking functional, aesthetic, structural, or temporal dimensions across domains. As the map grows, it becomes a field of latent ideas: unexplored intersections, hidden gaps, or unspoken assumptions waiting to be reassembled. To construct an analogy map, follow these steps:

1. Start with a focal concept. Not a category, but a concrete element (e.g., consulting workshop, client on-boarding email, or remote team collaboration).
2. Generate first-level analogies: What is this like? or What performs a similar function elsewhere? Choose a few and note the dimension of similarity.
3. Branch analogies from analogies. Take one of the analogies and ask again: What is this like? Continue to document analogies and specify *how* they relate.
4. Encourage drift, allow yourself to move far afield. From a consulting workshop, you may pass through conflict resolution (from your last workshop) to symphony rehearsal (what you did afterwards). With each step, note the specific function or experience that is similar.
5. Loop back and recombine. After several branches, look across the map and ask: Can I combine these? or What have I not explored?
6. Flag actionable nodes. Look for analogies that suggest implementable shifts. Could your consulting workshop borrow the tempo structure of a set list? Could your team's coordination mimic a pit crew model?

The map will not be tidy, and it is not meant to be. That is not the point. The point is to discover what is concealed by conventional categories. Don't confuse it with a mind map. So why do analogy maps work when mind maps do not?

- They externalise cognitive movement, not stasis. Mind maps represent static content. Analogy maps document the movement of thought and how concepts mutate and propagate.
- They privilege relationship over categorisation. Mind maps group similar things. Analogy maps trace *how things are similar*, which is far more generative.
- They surface hidden constraints. When you notice that your idea behaves like an airlock mechanism in terms of access control, it suggests security features you had not considered. Analogy reveals, while mind-mapping rarely does.
- They foster intentional divergence. Creative ideation requires exploring beyond the obvious. Analogy maps systematise divergence rather than depending on intuition alone.
- They are structurally rich. Mind maps are typically radial and depth-limited. Analogy maps can evolve in any direction: lateral, recursive, or nested. They grow like underground root systems (rhizomes), not trees.

Analogy maps are useful in various steps of the Design Thinking or D3 process. Some examples include: i) *Innovation sprints*: When you are stuck in domain-based thinking, analogy maps provide an escape route. ii) *Proposal or concept development*: Use analogy to differentiate your approach by importing models from other domains. iii) *Personal workflow improvement*: Want to reimagine your task system? What is it like? A railway timetable? A relay race? A farm? iv) *Idea expansion*: You have a vague idea. Analogy mapping lets you inflate it into variations and scenarios drawn from structurally similar systems. v) *Pattern recognition*: Over time, analogy maps help you develop a library of patterns and metaphors that recur across domains.

The allure of mind mapping is understandable. It looks creative and feels active. But for most people, it is only a visual map of their confusion, or at best of their preconceptions. The structure must precede the map. If it doesn't, the mind map only amplifies disarray. Analogy mapping offers a truer path to insight. It embraces chaos but does so *intelligently*. It gives individuals working alone a powerful way to think through movement, to find what is like what and ask *why*, and to uncover the idea that wasn't obvious until it was made strange. So the next time you are told to draw a mind map, ask instead: *What is this like, and in what way?* That is the start of an idea worth pursuing.

Value-Focused Thinking

Value-Focused Thinking (VFT), introduced by Ralph Keeney in the book with the same name, was originally developed as a method for improving decision quality. Instead of starting from a given set of alternatives (a reactive mode), VFT prescribes that one should begin by identifying the values that matter in a given context and then creatively generate alternatives that satisfy those values. While its roots lie in decision analysis, the method is readily adaptable to processes of innovation and ideation. Used this way, VFT becomes a tool not for choosing among pre-made ideas but for generating better ones in the first place. When repurposed for idea generation, VFT offers a structured and reflective approach that can significantly enhance the quality and relevance of personal creativity. It does so by placing emphasis not on the *what* (the ideas) but on the *why* (the values that motivate ideation).

In the context of Design Thinking, where the emphasis is often on empathising with users, generating ideas, and refining solutions through iteration, the contribution of Keeney's value-focused approach may at first seem like a detour into decision theory. But far from being a digression, it offers a depth and precision that many design processes silently crave. Innovation and creative design consist to a substantial degree of constantly making (mostly subconscious) decisions, so-called micro-decisions. What to focus on, which features to emphasise, or which directions to explore or skip. Keeney's insight is deceptively simple: most decision processes start too late. They begin with the options, not with the values. In his view, alternatives are merely means to an end. The real work of decision-making, and by extension, of design, should begin by articulating what matters. This reversal of logic from alternative-focused to value-focused thinking creates a powerful opportunity for Design Thinking to integrate a deeper, more strategic dialogue about purpose.

Design Thinking often starts from human needs. Through the empathy stage (a kind of ethnographic research), including journey mapping and persona building, it seeks to uncover, for example, pain points and aspirations. This user-centred orientation rightly grounds the process in empathy. But there is a subtle risk here: the needs uncovered may be fragmented, tactical, or circumstantial. A user might express frustration with a specific interface, a bottleneck in a service, or a missing feature. The designer(s) may then leap via the Define phase into ideation mode, generating dozens of ways to fix those pain points. What gets lost in this excitement is the question of values. Not only what the user is trying to fix or avoid, but what they are trying to achieve or express. Keeney challenges us to rise above problem-solving as a reaction and to instead treat each design challenge as an opportunity to deliberately create value. In his framework, values are not static preferences but expressions of what an individual or organisation cares about. They serve both as goals and as criteria. They are the *why* behind every meaningful *what*.

When integrated into a DT toolbox, value-focused thinking becomes a natural extension of the Define phase, but with stronger structural scaffolding. It asks the innovator(s) not just to synthesise user insights into a point-of-view statement but to construct a full hierarchy of values, starting from fundamental objectives and then branching into means objectives. This hierarchy gives clarity to the design challenge in ways that conventional framing tools might miss. Suppose a design team is working on improving urban mobility, a typical challenge in the Openlab Stockholm context. Without value articulation, the team might quickly move toward concepts like bike sharing, improved signage, or mobile ticketing. With Keeney's method, the team would first identify values such as freedom of movement, safety, environmental impact, cost accessibility, and personal dignity. From there, they would derive what design features or interventions might realise those values. This transforms the problem framing and point-of-view work from what-do-we-fix to focus more on what-do-we-want-more-of and what-should-guide-our-trade-offs.

Keeney's approach also has much to say about the test phase of Design Thinking, where prototypes are evaluated and iterated. Without clear values, testing can become a scattered undertaking. Teams may gather user feedback, look at usability metrics, or track behavioural data, but without a structured sense of which outcomes matter most, the evaluation risks being reactive or inconsistent. Value-focused thinking insists that the criteria for success be defined before solutions are considered. It offers a framework to evaluate prototypes not only on superficial preference but on alignment with declared values. This creates a common language within the design team and among stakeholders. If the value hierarchy was built collaboratively, including users, clients, and possibly regulators, then it becomes the north star against which alternatives are judged. This does not constrain creativity. On the contrary, it focuses creative energy on the design space that matters most. It filters out noise and reveals conflicts, such as between efficiency and inclusivity, that can then be negotiated consciously rather than discovered accidentally.

The discipline of generating values before solution concepts also counters one of the known traps in design: premature convergence. When a team becomes enamoured with a clever idea early on, they may distort the rest of the process to validate that idea rather than asking whether it truly aligns with what the users value. Keeney refers to this as being trapped by the structure of alternatives. By contrast, value-focused thinking invites teams to generate alternatives as a response to values, not in isolation. The method also includes techniques to help teams expand their mindset deliberately. These include wish lists, adversarial thinking, and generic value-driven creative thinking. Such tools are easily embedded into the Ideate phase of Design Thinking, not to replace sticky-note brainstorming, but to enrich it with purpose. A session guided by value-driven prompts, such as How else might we enhance dignity? or What surprising way might we achieve low emissions without compromising speed?, generates richer ideas than a purely problem-based frame. This keeps the ideation anchored while still allowing for divergence.

Another key contribution of value-focused thinking is its ability to support structured trade-off discussions. Many design projects, especially when working at the system level, involve competing objectives. A design for an elderly care app, for instance, might need to balance user autonomy with caregiver oversight. A redesign of school lunches might need to weigh cost against nutrition and variety against consistency. Keeney’s framework encourages explicit articulation of such trade-offs by requiring that values be defined independently and measured consistently. It leads to better conversations. Not, Do we like designs A or B?, but How do designs

A and B support or hinder our key values? From there, multi-attribute evaluation methods can be applied to compare alternatives transparently. Even if quantification is not precise, the structure disciplines thinking. It enables a more respectful dialogue between stakeholders, especially when values are in tension. The model creates a language that separates disagreement about *concept choices* from disagreement about *values*, an important distinction in complex design challenges.

VFT is not meant to be purely rationalist. Although it draws from decision theory and contains tools for modelling preferences, it is fundamentally aspirational. It invites people to imagine what they want to become, not merely what they want to avoid. This vision-driven mindset aligns beautifully with the most transformative forms of Design Thinking, where the goal is not just to fix but to reimagine. VFT urges decision-makers to ask What are the values you could create through this decision? This question translates directly to the designer's core brief. It shifts attention from constraints to opportunity. It also allows for more authentic engagement with long-term outcomes. Many design challenges suffer from short-termism. Solutions are optimised for immediate usability or cost but fail to deliver deeper or lasting value. By rooting the process in explicitly defined fundamental values, teams are encouraged to design not just for acceptability but for impact.

In practice, the methods are highly compatible. A DT team might begin with user research, but instead of moving directly to persona building, they could start extracting values from their findings. Interview transcripts, observation notes, and user quotes are mined not just for needs but for underlying goals. The team then constructs value hierarchies, distinguishing between what users want (means) and what they care about (fundamentals). These hierarchies are refined through dialogue, testing their clarity and independence. From there, ideation sessions focus on generating alternatives that serve those values in varied ways. Prototypes are selected or constructed to probe trade-offs. One might favour security, another flexibility. During testing, feedback is interpreted through the lens of values: which values does the user mention? Which values seem compromised? What tensions emerge? The values thus become both a guide and a compass, shaping the design process from start to finish.

From an educational or organisational standpoint, value-focused thinking also strengthens alignment. Often in design projects, different stakeholders bring different priorities: marketing might care about brand fit, operations about feasibility, and users about convenience. These perspectives can conflict or collide if not harmonised. VFT makes those priorities explicit and places them into a structure where they can be discussed constructively. When applied early, this alignment process can save enormous time and prevent frustration later. It also builds trust. Stakeholders who see their values represented in the design framework are more likely to support solutions, even when compromises are made. They recognise the process as legitimate.

On an individual level, value-focused thinking offers tools for personal design as well. When people use Design Thinking to reframe their lives, plan careers, or build habits, they are often driven by dissatisfaction or vague aspirations. VFT encourages them to clarify what they truly care about first. What do I value in my work? What kind of relationships do I want? What personal traits do I want to cultivate? From there, alternative life designs can be generated and tested. Instead of reacting to options presented by circumstance, one becomes the architect of options that are consistent with personal values. This mirrors the deepest character of design: to shape the future with intention.

The integration of value-focused thinking into Design Thinking does not demand a wholesale change in methodology. It is less a replacement than an enhancement. It sharpens the Define phase, grounds the Ideate phase, enriches the Test phase, and gives coherence to the whole process. It allows teams to make better decisions not only because they are more informed but because they are more attuned to what really matters. DT has always claimed to be about empathy and innovation. Value-focused thinking gives it the tools to be about clarity and purpose as well. The two together can transform not just what gets built but why it gets built.

So how is VFT practised in Design Thinking? Recall that at the core of VFT is the principle that ideas should be pursued to fulfil one’s values. The implication for a creative individual is profound. Rather than brainstorming randomly, one can focus ideation on producing ideas that serve explicitly articulated purposes. The first step is to identify fundamental values that relate to the context in which one is generating ideas. For example, if a consultant is seeking ways to improve how they deliver reports, they might uncover underlying values such as clarity, client engagement, timeliness, and adaptability. These values then serve as criteria for generating novel approaches, rather than judging completed work retroactively.

Importantly, VFT distinguishes between *fundamental* and *means* objectives. Fundamental objectives are ends in themselves, what one truly cares about. Means objectives are pathways to those ends. For ideation, the distinction matters: values that are too abstract or means-focused tend to produce narrow, derivative ideas. But fundamental values help uncover broader and more novel solution spaces. Techniques for applying VFT to the Ideation phase include:

Wish Lists Reframed as Value Lists. A productive exercise is to begin with a wish list that articulates not what you want in a superficial sense, but what you *value* in the outcomes you are trying to achieve. For instance, instead of wishing for a more popular blog, the underlying value might be increased influence on industry thinking. That framing leads to broader idea domains, including formats other than blogging, like hosting discussion salons or publishing infographics.

Consequence Back-casting. Instead of imagining how an idea might work, VFT encourages imagining the outcome that would be most desirable and then working backward to what ideas could make it real. This “consequence-driven” approach encourages more ambitious and creative ideas by decoupling ideation from the constraints of current options.

Objective Expansion Trees. This involves taking a stated value and breaking it down into more specific components. For example, a goal of being more impactful at work might branch into values like communicating persuasively, making timely interventions, and influencing strategic direction. Each of these becomes a springboard for ideation. The benefit is twofold: more entry points for idea generation, and a clearer lens for evaluating the relevance of any new idea.

Cross-context Reapplication. Once fundamental objectives are stated clearly, VFT enables the transfer of ideas across domains. If reduced friction is a value in client on-boarding, it might suggest ideas borrowed from e-commerce UX. Because VFT focuses on values, not domain conventions, it supports lateral thinking across disciplines.

Beyond idea generation, VFT offers a rational basis for idea screening and prioritisation. Instead of filtering ideas using subjective intuition or feasibility alone, one evaluates them against the identified value hierarchy. This avoids the common trap of eliminating bold ideas too early simply because they are unfamiliar. For personal idea management, a VFT-informed scoring matrix can be used. Fundamental objectives serve as axes. For example, an entrepreneur exploring product concepts might weigh ideas against values such as scalability, personal passion, market differentiation, and executional simplicity. Weak ideas are exposed not because they feel wrong, but because they score poorly against what is truly valued. This makes VFT especially helpful for resisting external biases. Instead of chasing what others have deemed hot, the innovator centres decisions on what is valuable to them, maintaining both integrity and relevance.

VFT can also assist with iterative refinement. An initial rough idea can be unpacked in terms of how well it addresses each fundamental objective. Where there are mismatches or gaps, the idea can be modified or hybridised with others. This recursive use of VFT turns the model into a tool not only for generating ideas but also for improving them. The method invites ongoing reflection: What would this idea look like if it better satisfied my objective of X? Such a line of questioning invites not only improvement but also divergent thinking and broader exploration.

When VFT is internalised and almost becomes second nature, the process becomes a self-exploratory tool. Identifying your values with respect to ideation, e.g., originality over certainty, autonomy over reach, or craft over scale, helps clarify not only what ideas are good, but what ideas are good *for you*. In this way, VFT contributes to what might be called *epistemic self-trust*: confidence in your internal compass when navigating ambiguous idea spaces.

Thus, a primary promise of adapting VFT to personal ideation is deliberate creativity. Instead of hoping ideas emerge from chaotic brainstorming, individuals can create structured generative environments by anchoring their thinking in their most important values. VFT does not replace intuition, experience, or serendipity. Rather, it channels them toward outcomes more likely to feel meaningful, feasible, and distinctive. For any individual in knowledge work, consulting, or creative practice, VFT becomes a method for making the most of both their insights and their imagination.

Core Process Mapping

Business Process Reengineering (BPR) was one of the most ambitious management trends of the 1990s. Made popular by Michael Hammer and James Champy, it promised radical transformation through the complete redesign of organisational processes. For a while, BPR was everywhere: organisations launched transformation initiatives, rewrote workflows, and implemented new enterprise systems. Over time, BPR went out of fashion, and a host of other methods took over. However, none became as influential as BPR once was. Still today, BPR has a useful and effective core: the identification and understanding of main processes, those value-generating sequences of activity that define what an organisation (or individual) actually does. If BPR sought to rebuild these from scratch, we will instead reframe this practice as Core Process Mapping (CPM): a method not for enforcing total change, but for revealing the structure and gaps of workflows and user procedures as they actually happen, and then identifying opportunities for innovation within or between those processes. In this way, CPM emerges as a systemic diagnostic lens. It is no longer about forcing transformation or aligning to an enterprise platform. It is about seeing an organisation as a system of interactions, and then identifying where improvement, bridging, or novel intervention might matter most.

Traditional BPR saw processes as things to be flattened, accelerated, or automated. What it lacked was empathy, context, and nuance. Today's challenges, whether digital transformation, hybrid work, or cross-functional collaboration, demand clarity and creative fit. Core Process Mapping reclaims the structural insight of BPR but discards its mechanistic philosophy. Where BPR demanded revamping of the old, CPM seeks to understand what is essential. Its value lies in helping individuals and organisations answer questions like:

- What are our most value-generating activities?

- How do people and tools interact in real time to deliver outcomes?
- Where are transitions awkward, unclear, or wasteful?
- What assumptions are embedded in our existing flows?

The aim is not to reengineer anything for the sake of it but to see clearly what already exists, and from that view, identify the opportunities for change, improvement, or innovation. A useful comparison is customer journey mapping of traditional Design Thinking, also used in both UX and service design. These maps chart the stages a user or customer goes through when engaging with a product or service. Typically, they include touchpoints, emotions, and pain points, often visualised with emoticons, timelines, and comments. Such maps are designed to foster empathy and help identify friction from the user's perspective. While this approach is valuable, it is often too shallow. It captures the *surface experience* of the customer but fails to interrogate the *underlying operations* that shape that experience. A smiling emoji at a service touch point may obscure the fact that five manual processes and three workarounds are required to generate that moment.

Core Process Mapping is structurally deeper. It looks not only at the experience of individuals but at the flows, functions, and interfaces that produce that experience. Rather than only seeking to relieve pain points, CPM seeks to understand why they exist and how the surrounding processes might be redesigned, reconnected, or supplemented. In this sense, CPM becomes a meta-version of customer journey mapping: it incorporates not only user emotions but also systemic function. It zooms out to see the *machine* as well as the *moment*. For innovation, this is important: it is not enough to delight the user, we must also ensure the organisation can deliver that delight sustainably and coherently. The practice of Core Process Mapping involves a sequence of focused steps:

1. Identify the core process: Choose a critical outcome area, for example client onboarding, proposal generation, ticket escalation. The process must be bounded but significant.
2. Map actual steps, not theoretical ones: Capture how the process works in practice, not how it was designed. Include informal steps, workarounds, and undocumented transitions.

3. Visualise flows across roles and systems: A CPM is not a flowchart within a department. It is a systemic map showing how actors, tools, and artefacts move over time.
4. Highlight handoffs and delays: These often indicate friction points, misalignment, or missing functionality. Ask: What information is lost? What decisions are deferred?
5. Annotate gaps, duplications, and assumptions: Where do people repeat steps? Where do they fill gaps manually? What does the process assume that is no longer true?
6. Assess strategic alignment: Is the process producing the outcome it was designed to? Does it serve current organisational goals? If not, where are the misalignments?
7. Identify innovation points: These may include missing support functions (a role that should exist but does not), integration opportunities (a link between systems or units), or design openings (a better way to achieve the same goal).

The CPM becomes a thinking canvas, a mirror for both analysis and creation. Unlike mind maps or stakeholder matrices, which often hover at the conceptual level, CPM grounds the individual or team in actual, operational reality. While CPM may sound like an organisational tool (which its origin BPR was), it is especially powerful for personal innovators and consultants. Mapping a core process they are involved in, whether service delivery, writing, client on-boarding, or data reporting, can reveal several things: i) Where friction comes from: Often not from the task itself, but from the transitions and context shifts between steps; ii) Where effort is wasted: Duplicate actions, unclear responsibilities, or tool misalignment; iii) Where new roles or supports might fit: A missing cog or puzzle piece that could increase coherence, speed, or quality; and iv) Where strategic leverage lies: A single handoff point that, if redesigned, could improve outcomes across the process.

A consultant, for example, might discover that although client on-boarding technically begins with a signed contract, the *functional* process starts weeks earlier with informal expectation setting. By mapping that reality, they can introduce earlier scoping rituals or pre-engagement diagnostics, improving downstream alignment and satisfaction. One of the key reframes in CPM is to shift from control to curiosity. BPR was about making organisations run faster and leaner, often through top-down restructuring and IT enforcement. CPM is about understanding the system you are part of so that you can find better ways to contribute, reshape, or influence it. This mindset aligns with systems thinking: see the whole, identify interactions, and focus on feedback loops. It also aligns with Design Thinking: observe, map, and prototype better states. CPM brings these together, not to create a perfect process, but to surface the system and reveal its potential. Business Process Reengineering may be largely forgotten by now, but its underlying insight, that structure drives behaviour, remains true. In today's complex work environments, individuals need tools that help them navigate ambiguity not by simplifying it, but by revealing where complexity matters most. Core Process Mapping is such a tool. It allows individuals and organisations to see the living architecture of work, to locate gaps and misfits, and to design not for control, but for coherence and innovation.

Pretotyping

A Design Thinking approach, especially at the individual level, culminates in testing and iteration. But how do you ensure that your idea is not only well-executed but also worth implementing in the first place? This is precisely the focus of Alberto Savoia's work in his book *The Right It*. Savoia, an innovation expert and former Google engineering director, reframes one of the most neglected risks in innovation: the risk of building something nobody wants, uses, or values. His main thesis is both blunt and liberating: Most ideas fail not because they are poorly executed, but because they are the wrong it. His core insight aligns directly with a Design Thinking core principle: to avoid wasting effort, you must learn early whether your idea addresses a real and significant need. But he takes the principle further by introducing a set of practical methods under the umbrella of pretotype testing (where pretotyping stands for *pretendotyping*, meaning to pretend there is a prototype ready for testing). Whereas prototypes test *how* well a solution works, pretotypes test *whether* the problem is worth solving at all and whether the idea has intrinsic demand, much in line with the first of two cycles in the Openlab D3 process. For the personal innovator, this becomes an important tool. Before investing time or resources in perfecting a new workflow, service offering, or internal project, you can pretotype to ask: Does this idea even deserve to live?

Savoia distinguishes between two main categories of risk: execution risk and market risk. Execution risk is about whether you can build or do the thing well. Market risk, by contrast, is whether anyone wants it or will use it once built. In personal work, the analogue to market risk might be relevance risk. You execute a new template, report, or process flawlessly, but discover nobody values it. This is surprisingly common in knowledge work. A well-designed new dashboard that no one opens, or a streamlined report process that is ignored in favour of ad hoc communication, reflects not execution failure but problem misalignment. Savoia's principle is that you should test for right-it-ness early. The personal innovator can translate this to mean that before rolling out a new process, habit, deliverable format, or initiative, do a low-effort check to see whether it will meet a meaningful need. His emphasis on fast, low-cost, high-learning tests makes pretotype methods ideal for personal innovation as well as more traditional innovation tasks.

Pretotyping is defined as testing the appeal and actual usage of a potential product before building it. While traditionally applied to product design, its principles adapt neatly to personal innovators. His methods are designed to be quick and cheap, which dovetails with the Design Thinking prototype mindset, but they answer a different question: not Does it work? but Is it worth building at all? Below are several of his pretotype techniques listed, partly reinterpreted for personal efficiency.

The Mechanical Turk Test. Create the illusion that the system or process is already in place, and see how people respond. Suppose you are considering automating a reporting process, but before spending time coding it, you manually generate the report and send it out, seeing if people engage with it. If no one reads it, automation is irrelevant. This lets you test user engagement without building the back end.

The Fake Door Test. Offer an option that leads nowhere to gauge interest. A personal innovator might add a section to their internal newsletter that says, Click here to receive a weekly insight summary, but the link leads to a thank-you page. If enough people click, it justifies building the actual summary. If not, no need to proceed. This reveals demand before building.

The Infiltrator Test. Introduce the new element seamlessly into an existing workflow and observe reactions. For example, include a new visual format in one slide of a presentation without calling attention to it, and see if anyone notices, comments, or imitates it. If they do, that suggests interest or resonance.

The One-Night Stand Test. Try the new idea once, in isolation. For a new personal schedule format, try it for one day only and see how it feels. A team meeting facilitator might test a new structure just once and gather immediate feedback before committing.

The philosophy behind all these methods is what Savoia calls TTMFP: Time To Make First Profit. In this case, not monetary profit but knowledge profit. As a personal innovator, you want to profit from learning as fast as possible, to guide your actions intelligently. Each of these methods can yield quick insight: Is this idea worth pursuing? Or is it a polished solution to a non-problem? One of Savoia's most useful tools is the Initial Level of Interest (ILI), which quantifies how many people out of those exposed to your idea actually engage with it. For a personal innovator, this can translate to: of those who receive the new report format, how many open it or mention it? If you make a process improvement, how many colleagues adopt it? A high ILI indicates that the idea is likely right, that it resonates. A low ILI means it needs rethinking or may not be addressing a real need. This idea of early interest as a proxy for future success is especially powerful when your resources (time, energy, political capital) are limited. Before making a major push, invest minimally to see if there is any uptake.

A critical philosophical point in *The Right It* is that creators often fall in love with their solution instead of the problem. Savoia urges innovators to stay focused on the problem and to be ruthless in abandoning solutions that don't pass the test of relevance. This aligns directly with the Design Thinking principle of staying human-centred and needs-focused. A personal innovator applying Design Thinking must beware of becoming too enamoured with their clever dashboard, a new checklist, or self-improvement system if it does not actually address a core pain point. By falling in love with the problem instead, you preserve adaptability. You keep testing different solutions, always tied to the core need. This also avoids the sunk-cost fallacy:

you don't keep iterating a solution just because you already invested in it, you move toward what actually works.

Savoia's framework does not replace prototyping and testing in Design Thinking but complements it. His concept of pretotyping can be seen as an early filter, a *pre-prototype*, within DT (as it is used in the first cycle of D3). For the individual, it sharpens the decision of *which ideas are worth prototyping* in the first place. In other words, pretotyping answers Should I? before prototyping answers How should I? and testing answers Did it work? In practice, an effective first cycle innovation process might look like this:

1. Empathise and Define the problem.
2. Ideate possible solutions.
3. Pretotype the most promising ideas to see which have traction or relevance.
4. Prototype those that pass the pretotype test to expand and refine the ideas.
5. Test and iterate based on performance and feedback.

By embedding Savoia's insights into your Design Thinking practice, you increase the odds that your ideas not only work but also matter. In the world of personal innovators, where time and attention are scarce, applying right-it thinking ensures that your innovations are not just efficient, but effective. You are no longer optimising for speed alone, but for direction. Moving quickly, yes, but in the right direction. Next, we turn to the closely related topic of systems thinking.

But before we leave design thinking behind, here is a pro tip that often surprises people. It deals with getting the really creative ideas that all of us know are sometimes hard to come by. Many of them come when you are not able to immediately write them down, as if our subconscious minds are playing tricks with us. To win those games, do the opposite of what other books tell you: don't keep a notebook near the bathtub and don't bring one for your long walks. Let the subconscious mind think it has the upper hand, and then counter by developing a memorising technique for those great ideas. When you unpack the ideas you got, make sure you do it in reverse order, taking the last one first, since that is how they were stored in your memory. That way, you optimise your retention. The overall goal is to make your subconscious mind your ally, your hidden partner as a personal innovator.

3. Systems Thinking

Design Thinking and systems thinking emerged from different intellectual traditions but speak to a shared human impulse: to understand complexity and to intervene in ways that generate meaningful change. When taken together, they offer a complementary set of perspectives and tools that, when merged thoughtfully, produce a more holistic and robust approach to innovation. At first glance, DT appears agile, concrete, and human-centred, favouring prototyping, iteration, and learning through direct engagement with users. Systems thinking, by contrast, seems slower, more abstract, and focused on the interdependencies within larger wholes, seeking to map relationships, feedback loops, and patterns that often remain invisible. But these contrasts are exactly what make the merger so potent. Each approach compensates for the other's blind spots while amplifying its strengths. Design Thinking brings immediacy, action, and empathy; systems thinking brings perspective, structure, and rigour. Where Design Thinking may rush into intervention without fully grasping the larger context, systems thinking ensures that the context is not only understood but meaningfully framed. Where systems thinking may overanalyse and drift into paralysis by complexity, DT insists on engagement, iteration, and doing. The synthesis creates a dynamic interplay: thinking while doing, seeing the forest without losing the tree, and acting without forgetting to observe the consequences across the system.

One of Design Thinking’s greatest strengths is its grounding in the lived experience of real people. It does not seek to design in the abstract but to respond to the emotional, behavioural, and practical realities of users. It is empathy-driven and solution-oriented. This makes it highly effective for uncovering unmet needs, generating ideas that feel intuitive, and ensuring that innovations connect with actual human behaviour. However, this same human-centeredness can sometimes limit the scope of analysis. In focusing on individual users, a design team might overlook broader systemic drivers that influence those users’ lives. For instance, a team designing a financial literacy app for young adults might zoom in on interface usability and engagement strategies but miss the underlying structural issues, such as lack of access to credit, wage stagnation, or education policy, that are shaping the user’s financial stress. Without a systemic lens, the solution risks being a surface fix, treating symptoms rather than causes. This is where systems thinking steps in. It compels designers to zoom out and ask, What structures are producing the patterns we see? What feedback loops sustain the current situation? Are there delays or unintended consequences that we are not seeing? Systems thinking does not merely map the problem; it reveals leverage points, i.e. places where a small shift can produce disproportionate impact. These insights elevate the design challenge from how do we make this better for the user to how might we alter the system so that the user's challenge no longer arises or is significantly eased.

Conversely, systems thinking often struggles with operationalisation. Its diagrams and causal loop maps are powerful for understanding, but not always helpful for knowing what to do next. The complexity it unveils can be intimidating, and without a concrete method for intervention, teams may feel stuck. This is where Design Thinking offers a release. By grounding systemic insights in prototyping and user testing, DT provides a pathway from map to movement. A team might use systems thinking to identify that burnout among hospital staff is not just about personal resilience but is linked to organisational culture, shift scheduling, or policy incentives. But what then? DT helps by translating this insight into tangible experiments: a new break scheduling app, a redesigned shift rotation, or a micro-intervention to improve staff recognition. These are not final solutions but testable probes into the system. The loop then continues: test, observe impact, feed the learning back into the system model, and iterate. In this way, DT becomes the action arm of systems thinking, turning deep understanding into informed doing.

The merger also invites new ways of framing problems. Design Thinking typically encourages reframing challenges based on user insight. Systems thinking encourages reframing based on pattern and structure. When combined, the result is multidimensional reframing. A team might start with the question How can we make public transport more pleasant and end up asking: How might we redesign urban time perception, so waiting is not experienced as waste? Or they might move from how to improve job satisfaction to How Might We rebalance reward loops in knowledge work to sustain intrinsic motivation. These deeper reframings do not emerge from one method alone. They are born at the intersection of observing the user and observing the system that surrounds the user. Similarly, prototyping can become more strategic. In pure DT, prototypes are often seen as expressions of an idea to test desirability or usability. In a systems-integrated approach, prototypes can also be designed as system probes: interventions meant not only to test an idea but to generate insight into how the system behaves. A policy nudge, a new feedback mechanism, or a changed workflow might all be prototypes that reveal the system's response. The intent shifts from finding the best product to understanding the most effective leverage.

Feedback also takes on new meaning in the merged model. In Design Thinking, feedback is typically user responses to a prototype. In systems thinking, feedback refers to causal loops, i.e. how an output of a system becomes an input, influencing future behaviour. By holding both interpretations at once, teams become more attuned to unintended consequences. For example, a well-received app feature might be judged successful in a design mindset but could, over time, trigger unexpected system effects, perhaps increasing user dependency or shifting user expectations in a direction that destabilises other parts of the offering. Being aware of these potential loops from the beginning allows for more sustainable innovation. Moreover, the feedback collected during testing is not just used to refine the prototype but to revise the system model itself. The result is an evolving understanding of both the solution and the environment into which it is introduced.

Another benefit of the integration is resilience. Systems thinking often highlights that interventions fail not because they are wrong but because the system resists them. Complex systems have inertia. They push back. Design Thinking, with its iterative and adaptive style, is well suited to navigating this resistance. It encourages small steps, fast feedback, and learning through doing. Instead of launching a massive system redesign and hoping for adoption, a design thinker starts with a lightweight prototype and watches what happens. If the system resists, the prototype can be adjusted or re-scoped. This makes change less brittle. It also makes designers more humble. Many systems thinkers are fond of systems theorist Donella Meadows' insight that most leverage lies not in fixing the system but in shifting paradigms. DT may not claim to shift paradigms, but it is highly effective at surfacing them by exposing unspoken assumptions and inviting teams to imagine alternatives.

Together, they provide both the diagnosis and the therapy for complex challenges.

Perhaps most significantly, this fusion transforms the role of the designer. No longer merely a problem solver or user advocate, the designer becomes a systems navigator and change agent. The brief is no longer to make a thing better but to make a better system possible. This shift requires new capabilities: not just empathy but perspective taking; not just creativity but systems literacy; not just ideation but modelling. It also requires a new posture, one of humility and curiosity. Complex systems do not yield to control, but they respond to engagement. The designer in this new mode learns to dance with the system, to experiment within it, and to learn from its responses. This is a slower, deeper, and perhaps more profound form of design.

In practice, merging these approaches can take many forms. It might begin with a Design Thinking process enriched by a systems mapping exercise. Or a systems analysis might be concluded with a co-creation workshop rooted in design methods. Tools from both traditions can be blended: journey maps annotated with feedback loops, stakeholder maps extended into influence diagrams, and personas that include systemic roles. The methods are less important than the mindset. What matters is approaching problems with a willingness to look both close and far, to feel and to analyse, to act and to reflect. The merger is not a hybrid for its own sake but a recognition that complex human challenges demand both understanding and imagination, both rigour and empathy.

At a time when the problems we face are more interconnected than ever (climate change, social inequity, digital dependency, and institutional inertia), the need for joined-up thinking has never been more acute. Neither systems thinking nor Design Thinking is sufficient on its own. One gives us the map, the other the vehicle. One sees the landscape, the other explores the terrain. Only together do they allow us to chart a path forward that is not only desirable but feasible and sustainable. In this marriage, the promise is not only better solutions but wiser designers. Not only smarter systems, but also more human change.

In the earlier discussion, Design Thinking was considered not only a team-based method for innovation but as a powerful lens through which individuals can enhance their personal efficiency and creative effectiveness. The notion of working like a designer, iteratively, empathetically, and purposefully, was broadened to include personal innovators who seek better ways to manage knowledge, ideate, and solve problems. Yet one important dimension touched upon, but not fully explored, was systems thinking. This complementary approach adds cognitive depth to the design mindset, especially when it comes to anticipating long-term consequences, understanding interdependencies, and designing within dynamic contexts.

Systems thinking equips individuals to perceive their decisions not as isolated events but as part of a larger web of interactions. Unlike the reductionist view, which analyses discrete components, systems thinking focuses on synthesis: how parts function in relation to the whole. When applied to individual work, this outlook shifts the objective from task execution to strategic consequence navigation. A personal innovator drafting a proposal, for example, should not only consider the immediate outcome (approval or rejection) but also its ripple effects on departmental strategies, budget allocations, and even employee morale. Seeing such interconnections leads to better-aligned, more resilient actions.

Russell Ackoff, a systems theorist, warned that many performance improvement initiatives fail because they optimise a part of the system at the expense of the whole. This phenomenon is not limited to corporate settings. A personal innovator who focuses only on efficiency, such as responding to more emails or finishing more tasks, might unwillingly contribute to a culture of reactive busyness while missing the bigger question of what is worth doing. A systems perspective encourages reflection on whether one’s current inputs contribute to the desired system-wide output, which often includes non-obvious outcomes like knowledge transfer, emotional tone, or stakeholder confidence.

One of the most powerful personal insights from systems thinking is the realisation that one's worldview is not neutral. Systems thinking begins with metacognition, i.e. awareness of one’s own assumptions, mental models, and interpretative frames. As the physicist David Bohm remarked, thought fragments the world by default, unless we gain awareness of how our thinking shapes our perceptions. A systems-aware individual makes the invisible visible by surfacing assumptions and considering how their framing influences problem perception. Consider a consultant mapping out a client engagement. A linear view might focus on deliverables, deadlines, and scope. A systemic thinker, however, pauses to consider how internal politics, historical disappointments, stakeholder interdependencies, and power dynamics shape the context. By adjusting their framing, what Ackoff called changing the worldview, they are better positioned to design not just a solution, but one that is compatible with the system's capacity to absorb and sustain it. Mental models are often tacit, but systems thinking urges their externalisation. Diagramming tools like causal loop diagrams or influence maps can help individuals visualise complex relationships. This is particularly useful when navigating situations that involve multiple stakeholders, such as organisational change or multi-departmental projects. When made explicit, these models are not just thought aids but communication tools, allowing others to critique or refine the assumptions and causal hypotheses that underlie a plan.

The shift from analysis to synthesis is a defining move in systems thinking. Analysis breaks problems into parts; synthesis looks at how the parts relate within the system. For individual professionals, this means resisting the lure of isolated optimisation and instead focusing on patterns and relationships. For example, rather than asking how to speed up meeting preparation, a systems thinker might ask what structural or cultural dynamics make the meetings ineffective in the first place. The former is an efficiency question; the latter is an effectiveness question at the system level. Synthesis leads to better prioritisation. In a daily workflow filled with alerts, requests, and fragmented tasks, a systems-oriented individual considers not only urgency and importance but also systemic influence. Which tasks unlock others? Which address root causes? Which feedback loops are being activated? This framing is especially important in environments of high interdependence and complexity, such as product development, policy design, or consulting.

A distinguishing feature of systems thinking is its treatment of causality. Rather than simple linear causation (A leads to B), systems thinkers recognise non-linear, mutual, and delayed causality (A affects B, which affects C, which loops back to A). This shift alters how a personal innovator interprets success and failure. A presentation that initially seems well-received might trigger unforeseen resistance two weeks later. A decision to automate a task might reduce overhead now, but erode team learning over time. Mapping consequence chains helps anticipate such effects. This can be done through tools like system archetypes, which help individuals recognise common structural traps. For example, the fixes that fail archetype captures situations where a short-term solution inadvertently exacerbates the long-term problem. A systems-aware individual would avoid this by considering both immediate and delayed feedback loops and testing for reinforcing and balancing dynamics. Concrete methods include drawing feedback maps, using storytelling to trace sequences of influence, or conducting pre-mortems to imagine how a plan could fail. These techniques encourage a mindset of careful attention and humility, acknowledging that all interventions have side effects and that outcomes often diverge from intentions due to systemic complexity.

Systems thinking challenges individuals to set system boundaries thoughtfully. Every model simplifies; the question is what to include and what to exclude. A personal innovator applying systems thinking must decide which stakeholders, temporal horizons, and metrics are relevant. Narrow framing often leads to sub-optimisation. For instance, focusing only on customer metrics might ignore employee burnout, which later cascades into customer dissatisfaction. Once boundaries are set, systems thinking helps identify leverage points, places where small interventions can lead to large systemic shifts. For individuals, this might mean shifting from reactive tasking to influencing norms, incentives, or shared mental models. An employee might choose to initiate a change in meeting rituals, thereby improving collective decision quality. A consultant might alter how success is measured in their deliverables, thereby shifting client focus from outputs to outcomes. The goal is not to control the system but to co-evolve with it, nudging it toward desirable states through strategic, high-leverage actions. This aligns well with the character of Design Thinking, where prototyping and iteration are used not to deliver perfect solutions but to shape emergent futures. Several practical tools can be used to integrate systems thinking into individual workflows:

- *Causal loop diagrams*: Useful for mapping reinforcing and balancing loops around key behaviours or outcomes.
- *Rich pictures*: Visual narratives that capture both quantitative and qualitative elements of a situation.

- *Behaviour-over-time graphs*: Help visualise trends and lagging effects.
- *Leverage point analyses*: A structured method for identifying high-impact interventions within a system.

Importantly, these tools are not substitutes for judgment but aid in expanding perception. They help individuals pause, reflect, and act with greater situational awareness. A personal innovator applying systems thinking is not merely productive but generative. They operate not only within systems but upon them, constantly tuning their actions based on structural insight. This orientation fosters foresight, as individuals become attuned to downstream consequences, unintended effects, and hidden stakeholders. It also strengthens adaptability. When plans fail, systems thinkers revisit assumptions and feedback loops rather than blaming actors or doubling down.

Such an individual, equipped with both Design Thinking's human-centred creativity and systems thinking's structural foresight, becomes a more effective consultant, personal innovator, leader, or policy advisor. They generate solutions that are not only innovative but sustainable, not only efficient but meaningful, and not only timely but future-proof.

Cybernetics

Systems thinking and cybernetics share a deep and intricate lineage, each enriching the other in ways that are both conceptual and practical. While systems thinking offers a language and orientation for understanding interconnected wholes, cybernetics sharpens that understanding by focusing on control, communication, and regulation within those systems. The two traditions have evolved side by side, often overlapping, but with distinct emphases. Systems thinking seeks to map relationships and dynamics; cybernetics probes how those dynamics are stabilised or destabilised, how they are guided, and how they might be designed for adaptation and autonomy. When considered together, they form a more complete toolkit for grasping and engaging with complexity, one that is as useful in engineering and governance as it is in organisational design and personal reflection.

At the heart of cybernetics is the study of feedback: how systems respond to themselves through mechanisms that either reinforce or correct their current trajectory. This emphasis resonates with but also extends systems thinking. A systems map might show how variables are connected, how one factor influences another, and how loops emerge. Cybernetics asks: What kind of loop is it? Is it balancing or reinforcing? Is the feedback delayed or immediate? Does the system learn? In this way, cybernetics becomes a microscope within the systems thinking framework. It does not discard the big picture, but it drills into how that picture evolves over time, especially under stress. It brings temporality and control into sharper focus.

Stafford Beer's work stands as one of the most ambitious and elegant efforts to operationalise this union. His Viable System Model (VSM) is not merely a theory of organisational structure; it is a living cybernetic expression of systems thinking applied to real-world governance. The VSM is based on the idea that for any system to be viable, that is, to survive in a changing environment, it must be able to monitor itself, respond adaptively, and coordinate its parts while maintaining coherence. These are not abstract ideals but functional requirements. Beer's model divides systems into five interacting subsystems: operations, coordination, control, intelligence, and policy. Each plays a distinct role in keeping the whole viable. Operations are the doing parts; coordination ensures operations do not interfere with each other; control balances the needs of the parts and the whole; intelligence scans the environment and predicts change; and policy sets the long-term identity and purpose of the system. This layered structure echoes the human nervous system and is deeply inspired by biological organisation. It is also fully cybernetic. Each subsystem includes feedback loops, communication channels, and mechanisms for recursion (subsystems within subsystems), allowing the model to scale naturally.

What Beer adds to systems thinking is not just a diagram but a philosophy of governance that respects complexity without becoming paralysed by it. He argued that centralised control is inherently limited in complex systems. No one node can process all the information needed to guide the whole. Instead, viable systems must be distributed, autonomous at local levels but coherent at the whole level. This is where cybernetics shines: it explains how autonomy and coherence can coexist. Feedback loops, carefully designed and nested, allow parts to self-regulate while remaining aligned with the broader purpose. In Beer's work, this is not a metaphor.

It is a design principle. Whether in companies, governments, or even families, the goal is not to command from above but to construct a system architecture that enables adaptation, learning, and self-correction from within.

This contrasts sharply with traditional command-and-control thinking, which tends to treat systems as machines to be operated, not organisms to be cultivated. Systems thinking already challenges that view by emphasising interconnection and emergence. Cybernetics goes further by showing how command-and-control models often generate instability because they ignore feedback or apply it too late. A cybernetic approach accepts that control is always partial and delayed. It does not eliminate uncertainty but manages it. The skill is in tuning the feedback loops, making sure signals get where they need to go, quickly enough, and that responses are proportional. In a volatile world, the ability to dampen harmful oscillations, or to amplify weak signals of change before they become crises, is a source of resilience. This is one reason Beer's ideas have remained influential, even decades after their initial development. He saw what many still overlook: that design is not only about form but about flow, not only about roles but about reflexes.

Another key contribution of cybernetics, especially as Beer developed it, is its recursive nature. Systems are seen not as flat structures but as nested wholes. A viable system contains viable subsystems and is itself part of a larger viable system. This recursive insight mirrors real life. A team within an organisation needs to be able to manage itself, learn, and adapt. But so does the department above it, the organisation above that, and the ecosystem surrounding them all. Each level requires its own version of the same basic functions: operations, coordination, intelligence, and so on. This insight helps prevent the frequent error in systems thinking where the focus remains fixed at one level (organisational, societal, or global) without attention to how dynamics scale or distort across levels. Cybernetics provides a language for multi-level design, ensuring that autonomy and alignment are preserved as complexity grows.

Importantly, cybernetics also engages directly with the question of purpose. In classic systems thinking, purpose is sometimes implicit. It is inferred from the observed behaviour of the system. In cybernetics, especially second-order cybernetics, purpose becomes more active. The observer is part of the system. The modeller cannot stand outside the system being described but is implicated in it. This leads to a more reflexive, and arguably more honest, practice. Beer was early to explore this idea, recognising that models of organisation are not neutral but value-laden and that designers of systems must be aware of their own role in shaping what is seen and what is excluded. This reflexivity aligns with today's concerns about bias, agency, and power in system design. Cybernetics does not solve these challenges but provides tools to surface them: Who controls the feedback? Who decides what counts as noise? How is information distorted as it moves through the system? These are cybernetic questions, and they are ethical as much as technical.

Where systems thinking tends to stay in the realm of analysis and mapping, cybernetics drives toward design and enactment. It asks not only what the system is but how it can be steered. This does not imply authoritarian control but participatory design. In Beer's later work, especially in the Chilean Project Cybersyn, he envisioned a cybernetic form of national governance that was radically transparent, decentralised, and participatory. While that project was ultimately cut short when Pinochet ousted the Allende regime through a military coup, it remains a landmark case of systems thinking and cybernetics in action. Its control room, feedback dashboards, and real-time data flows prefigured today's national production on dashboards and adaptive graphics. More than the technology, it was the principles that mattered: governance as conversation, not imposition. The author remembers having a long discussion with Beer over the running of the Cybersyn project in his later days, and the takeaway was that it was as much a political vision as a realistic project, even though it actually ran on 500 telex machines across Chile and a central computer in the basement of the Chilean presidential palace in Santiago de Chile. It was like an ERP (enterprise resource planning) system for an entire nation, but it was as much a democratising information system for the people as it was a centralised control mechanism.

In practice, blending systems thinking with cybernetics enables a richer range of action. You can map the system, yes, but you can also simulate it, test interventions, design new structures, and build in feedback that allows the system to evolve. This is true whether you are managing a team, designing a policy, or rethinking an educational programme. You are not just diagnosing but rewiring. This demands a mindset that is both analytical and creative, humble and bold. It requires an appreciation for patterns without an obsession with prediction and for autonomy without loss of direction. In a world increasingly marked by complexity, volatility, and interdependence, these are the very qualities needed for wise action.

To engage with systems thinking is to see. To engage with cybernetics is to listen, respond, and design for resilience. Together they form a discipline of attentiveness, a way of being in the world that is both practical and philosophical. Stafford Beer understood this well. His models were not just technical artefacts but invitations to see organisations, and by extension ourselves, as dynamic, learning systems. Systems thinking gave him the lens; cybernetics gave him the means. That marriage remains one of the most powerful in the history of thought about complexity, and its relevance has only deepened. The more we live inside systems we cannot control, the more vital it becomes to learn how they can be guided, not with force but with insight, structure, and care. Next, we look at a tool from the systems thinking toolbox that helps uncover system dynamics, or rather the missing dynamics.

Missing Cogs and Puzzle Pieces

Within the context of systems thinking and design-oriented personal innovation, the concepts of missing cogs and missing puzzle pieces offer fertile ground for generating practical and impactful ideas. These metaphors direct the individual’s attention away from standalone solutions and towards structural fits within broader systems. They represent two complementary ways of identifying innovation opportunities: one by identifying what is *absent but expected* (the cog), and the other by identifying what is *absent but needed to integrate or enhance* (the piece). Both approaches rely on understanding the system's structure, actors, and interdependencies, and can be applied by individuals as scanning and ideation methods within their own domains of work.

The missing cog metaphor refers to a service, product, or intervention that is not currently present but would *naturally fit* within an existing system or workflow. Like a gear in a machine, its function is contingent on the rest of the system being in place. A cog innovation leverages existing infrastructure, relationships, and processes. It completes the mechanism rather than reinventing it. For personal innovators, scanning for missing cogs begins by mapping existing systems and asking what functions are implied or required but currently unfulfilled. The premise is: if the system already has components A, B, and C, what function X would make A–C run more smoothly or deliver more value? This approach is common in fields like UX, service design, and internal operations, where systems often evolve incrementally and leave functional gaps. Techniques for identifying missing cogs include:

Workflow mapping. A practical tool is to diagram the end-to-end process of how a service or product currently operates, including actors, handoffs, delays, and dependencies. Look for bottlenecks or awkward transitions, these are often signs of a missing cog. For instance, a freelance consultant who maps out their client on-boarding process may realise that although contracts and payment systems are in place, there is no structured knowledge handover at the project start. The missing cog could be a standardised project initiation canvas to be completed by the client. It fits the system, requires minimal education, and completes the loop.

Heuristic questioning. Use prompts such as What seems to be missing? or What do other systems of this kind usually include? These questions draw on analogical reasoning. If similar systems have an internal analytics dashboard and yours does not, is that a missing cog or a deliberate exclusion? This method helps identify *normative absences*, i.e. things that users expect but do not find.

Failure mapping. Analyse past breakdowns or inefficiencies and look for recurring omissions. For example, repeated miscommunication during project handoffs might suggest a missing cog in knowledge transfer. Creating a structured handover checklist might resolve this. The innovation is not radical; it is corrective and systemic.

Actor-role mismatch analysis. Map out who performs which roles in a system and determine whether any actors are doing work that should logically be done by a different role. This often signals a missing support structure (a cog). For example, if managers are manually compiling data reports weekly, the missing cog may be an automated data aggregation script or a dashboard tool, something the system needs but has not yet acquired.

Maintenance gap identification. Many systems function well at launch but degrade over time due to a lack of maintenance structures. A missing cog may not relate to creation but to *sustainability*. A personal innovator might identify that after initial team training, there is no refresher mechanism. Hence, the missing cog is a short, recurring maintenance brief or knowledge capsule. This kind of idea is often overlooked because it solves a long-tail problem, not a launch one.

The value of identifying missing cogs lies in alignment: the idea naturally integrates with current practices, enhancing flow or reliability. These ideas tend to be low-resistance and high-uptake because they fit the logic of the system. While missing cogs are about completing a function within a system, missing *puzzle pieces* represent innovations that link or synthesise existing entities. They do not replicate the logic of A or B but introduce C, a novel construct that ties them together. In this sense, the piece is not another cog, but an adaptor, converter, or scaffold. Missing pieces generate value by enabling synergy across silos or subsystems. Puzzle-piece innovations often emerge when two functions, tools, or departments operate in parallel but do not interact productively. For the individual, this perspective encourages looking between systems rather than inside them. Techniques for identifying missing puzzle pieces include:

Interface mapping. Document where two tools, processes, or departments intersect or fail to do so. Look at the seams. If your project tracking software and communication platform are completely decoupled, the missing piece might be a simple bot that posts project updates into the team chat. It is not a new platform, it is a bridge. This is the puzzle piece logic: it connects without replacing.

Double-user analysis. Identify two groups who use different systems for related purposes but would benefit from shared information or coordination. The classic case is sales and customer support. For a personal innovator in one of those roles, recognising that a feedback loop is missing can lead to the idea of a feedback sync summary, a light artefact that shares insights across departments. The innovation is the piece that carries context, not the tools themselves.

Layer design. Consider what kind of meta-layer could unify or enhance two existing systems. For example, you might identify that project retrospectives and onboarding are treated separately, yet insights from the former could improve the latter. The missing piece might be a knowledge capture tool that outputs digestible onboarding modules from retrospectives. The systems remain unchanged, but the piece synthesises them.

Triangulation scans. Use triangulation by identifying two areas you know and looking for what connects them. This method asks: what third element would bring coherence or added value? For example, you work with both marketing and analytics, but they rarely collaborate. A campaign insights dashboard that translates analytics into marketing-friendly language and actions could be a puzzle piece, bridging cognitive and linguistic divides. The technique is to observe two non-integrated functions and imagine what would *make them cohere*.

Value chain audits. Step back and look at the entire value chain of a process or service. Where are the discontinuities? Puzzle piece ideas often arise where two value chain stages rely on implicit, informal, or broken links. A personal innovator might discover that training and tool use are unlinked. People are trained once, but tools evolve. A missing piece could be an update digest that shares new tool features monthly with short guidance, thus linking two evolving stages of the value chain.

Puzzle piece innovations are structurally valuable because they reduce friction and increase system-level coherence. They often punch above their weight because they unlock latent synergies. However, they require more conceptual abstraction than cog ideas. They must see *between*, not just *within*. For the personal innovator, scanning for cogs and pieces should become a regular discipline. It involves viewing your work and environment not just in terms of tasks but as a living system with gaps and interfaces. Useful prompts might include:

- What should exist here, given the way things work?
- What would make these two tools/teams talk to each other?
- What is missing that everyone works around without naming?
- What acts as glue, not driver?

These questions create an innovation surface area, a mental space where pattern recognition, systems understanding, and user empathy converge. Cogs display flow within an existing context. Puzzle pieces display connections between them. Both offer high-leverage points for the personal innovator working inside complex knowledge ecosystems. Whether creating tools, refining processes, or reshaping service models, the ability to see what is missing is one of the most strategically creative faculties in systems-aware design.

We end this chapter with a reflection on how even experienced organisations can forget the importance of proper systems thinking. A smaller car manufacturer was planning its entry into the Swedish market. It had a seemingly perfect market proposition: a microcar, let us call it X, designed for two passengers and small enough to fit into half a parking space, or into locations where ordinary cars could not park. These kinds of spaces are often left unoccupied in cities, and urban buyers were the primary target group. When Project X began, the plan was for the car to be electric or hybrid. Well ahead of its time, just as the Toyota Prius was making its debut. However, before launch, Project X was converted into a petrol-powered vehicle. This made sense from a systems point of view at the time. An electric car would have required charging infrastructure, which few people were thinking about then. By switching to petrol, the car became simpler to manage in the prevailing ecosystem. Yet, in dropping the electric vehicle idea, the company also dropped its systems thinking. When the X car launched in Sweden in the early 2000s, it was sold through a network of sales offices in major cities, piggybacking on a well-established car dealership chain. However, it lacked an equally developed network of service workshops. Teaching a chain of workshops how to service and repair an entirely new car concept (or establishing a new one) takes time and money. And with few units initially sold, there would be limited billable service hours. Thus, a company that had shown clear systems thinking when X was envisioned as electric or hybrid became short-sighted over time. They forgot that any car, regardless of its energy source, requires maintenance and a steady supply of spare parts. After a promising initial reception, sales began to decline steadily as stories spread about the X car being difficult or impossible to service. In a misguided attempt to counteract this, the X car was known as “the car that does not need service,” a claim that only made things worse from a customer perspective.

4. Common-Sense Thinking

What is common-sense thinking (CST)? It is sometimes thought of as an all-purpose ability to make sound judgements in everyday situations, transcending any one domain of knowledge. In simple terms, it could mean using sound and prudent judgement based on a simple perception of the situation or facts. This goes besides formal education or specialised training. It is partly an innate capacity to draw on basic, widely shared knowledge and intuition to solve real-world problems. For example, what about knowing not to touch a hot stove or to save money for a rainy day? It does not require any academic training, but rather a general awareness learned through experience. Still, this is not CST. Common sense is thought of as *all the knowledge about the world that we take for granted but rarely state out loud*, i.e. the broad background understanding that any typical person has accumulated by adulthood. It includes a grasp of physical realities, social norms, and practical consequences. The concept of common sense encompasses all this, but much is common sense knowledge, i.e. the facts and relations that are shared static knowledge of a general nature. The stove example above is common-sense static knowledge, but does not require much dynamic thinking to be applied. So while certainly commonsensical, it is not what we mean by CST.

The relation of CST (process) to the broader concept of common sense (outcome) is similar to the relation between design and Design Thinking. Design is defined as the purposeful arrangement of elements to achieve a desired function, outcome, or experience, balancing constraints, requirements, and sometimes creativity. Design Thinking, however, is the process of design distilled from its many instantiations and codified as a method. So *design* focuses on creating for a known problem within defined constraints, applying conscious thought to achieve a functional solution. *Design Thinking*, on the other hand, approaches more general problems through human-centred inquiry, by uncovering underlying needs before proposing solutions. Analogously, CST is the core process when applying common-sense knowledge to a non-trivial situation.

Despite its name, common sense is not actually so common or easy to pin down. Psychologists and philosophers note that what counts as common sense varies with context, culture, and experience. Still, most would agree it involves some kind of ability to adapt one's thinking to a variety of real-world contexts, using a blend of prior knowledge, inference, and good judgement. In the sections below, we explore how researchers define this general-purpose reasoning ability, how it differs from other forms of intelligence such as street smart and how it manifests in fields like psychology, management, education, and decision-making. We also look at attempts to study and cultivate common sense thinking and compare it to related constructs such as practical intelligence, fluid intelligence, cognitive flexibility, and executive function.

The Nature of CST

It is important to distinguish between two kinds of CST. This distinction bears some resemblance to Kahneman's thinking systems 1 and 2 from his book *Thinking, Fast and Slow*, where 1 is the fast, precompiled knowledge and 2 is the slow, constructed knowledge that has to be assembled when needed. In a similar but not analogous way, there are two distinct ways of common-sense thinking. Let us call them System A and System B. System A is the everyday common-sense thinking that seems effortless and that most people display a good command of. Using System A thinking seems almost trivial and is done without most people ever consciously reflecting on it. This is not to suggest that System A operates without calculations or contemplation; that would be incorrect. Rather, these processes often occur rapidly and largely at a subconscious level. For example, if you drop a knife, your immediate reflex is to catch it. However, your consequential System A quickly projects the likely outcomes of doing so. If the knife has a sharp edge, System A sees the risk of injury and advises against catching it. It can be almost like a visual slide show or film with a few frames. If it is instead a table knife, you may get a little messy but not hurt by catching it. Rather, if the floor is scratch-prone, you prevent a dent if you catch or at least deflect the table knife. System B, on the other hand, are more complex realisations of where things would go if a decision is made in a specific manner or if events unfold in a particular way. While most people are reasonably good at System A CST, being a master of System B CST thinking is more unusual, even among people with high academic degrees and/or high IQ. Failure to distinguish those two different manifestations of CST is a major factor behind the sometimes confused discussions and debates on common sense and the lack of it in some contexts.

At its core, CST is the human capacity to make sensible presumptions about everyday situations and react with appropriate judgement. This means having a vast store of background knowledge about how the world works, from physical principles (e.g. that heavy objects can crush lighter ones) to social expectations (e.g. understanding other people's likely intentions and beliefs). Cognitive scientists often point out that a typical seven-year-old already possesses an enormous amount of this common-sense System A knowledge about objects, people, animals, and basic routines of life. Unlike encyclopaedia facts or academic theories, System A knowledge deals with general truths (water quenches thirst, friends expect loyalty, a dropped glass will likely break) that are broadly true across contexts. However, this knowledge is not automatically transformed into effective and efficient System B thinking.

System B is the reasoning part of common sense: the ability to use that background knowledge in a flexible, situation-appropriate way. CST System A is largely intuitive and fast. We draw almost immediate conclusions that feel obvious without any analysis. For instance, if we see someone shivering in light clothing, we immediately infer they are cold and ought to put on a jacket. Such inference appears effortless. We effortlessly understand narratives and predict likely outcomes in daily life because of an underlying web of common-sense understandings (sometimes called folk psychology and naïve physics in cognitive science).

It is important to note that common sense is not infallible. It can be biased or mistaken in novel scenarios. But as a cognitive ability, it is essentially our general-purpose tool for navigating life's myriad unstructured problems. Unlike formal logical reasoning, which uses abstract rules, common sense reasoning tends to be contextual, experience-guided, and pragmatic. It might tell us, for example, that if a deal sounds too good to be true, it probably is. This is a conclusion drawn from diffuse real-life examples rather than a mathematical proof. So common sense is the blend of general world knowledge and practical reasoning that allows humans to operate across domains with a baseline of good judgement.

Beyond Book Smartness

CST can be contrasted with the type of intelligence measured in school exams or IQ tests, sometimes called book smarts or academic intelligence. High IQ or academic prowess does not guarantee strong System B common sense in real life. Cognitive psychologists have long observed that some people who excel in analytical problem-solving can still make poor everyday decisions or show startling lapses in overall judgement. For example, Sternberg noted that cleverness is far from enough to excel at real-world tasks. He described observing many individuals with high IQ scores and advanced degrees who nevertheless made a mess of basic tasks in jobs or relationships due to a lack of practical sense. Other research backs up this distinction. Stanovich and others who study reasoning have found that standard IQ tests fail to capture many good-thinking skills needed for rational decision-making in daily life. Those tests focus mostly on abstract problem-solving and knowledge recall, which correlate with academic and professional success up to a point. But they fall short of the full set of skills that would come under the rubric of ‘good thinking. When researchers specifically test people’s reasoning and judgement in practical scenarios, high-IQ individuals do not necessarily outperform others. In other words, someone can be a brilliant mathematician or physicist, yet still lack what we call System B common sense. For example, in bad cases, they might be gullible, struggle with basic personal finance, or miss obvious social cues. Psychologist Bruce Charlton dubbed such cases clever sillies, suggesting that some very bright people overthink problems and override behaviours that are actually common-sense. They might devise convoluted solutions where a simple, sensible approach would do, indicating a disconnect between analytic intelligence and practical reasoning.

From a research standpoint, System B common sense aligns more with what psychologists label practical intelligence. Sternberg argues that practical intelligence is largely distinct from the analytical intelligence that IQ tests measure. Practical intelligence is about applying knowledge to real-world situations effectively, something Sternberg says depends greatly on tacit knowledge. The kind of know-how one picks up informally through experience. Unlike solving a geometry puzzle (where all needed information is given and the task is clearly defined), System B common sense problems can be messy and tacit. You have to know which details matter, draw on unstated background facts, and infer possible consequences. Such abilities are shaped by experience rather than textbook learning. It is been found that measures of tacit knowledge and practical intelligence correlate only weakly with traditional IQ. In practice, this means a person's score on an IQ test does not strongly predict how well they can navigate everyday challenges or employ good common sense. For instance, Sternberg and colleagues developed tacit knowledge inventories (questionnaires about how to handle realistic work and life scenarios) and discovered that performance on these had little relation to conventional intelligence scores. This provides evidence that common-sense thinking is a separate cognitive domain. One might be high in both academic smarts and common sense, but one can also be strong in one and weak in the other.

Thus, academic intelligence reflects the ability to learn, analyse, and solve difficult problems, often in specialised domains, whereas common sense intelligence (and thus thinking) reflects the ability to reason effectively across everyday situations. The academically bright individual might master calculus or symbolic logic, but the System B common-sensical individual excels at reading the room, spotting practical pitfalls, and making prudent decisions without needing a formula. In short, there is a lot more to being a good thinker than having a high IQ. Common sense in a System B sense encompasses that lot-more, i.e. the real-world judgement calls and adaptive thinking that standardised tests do not capture.

Street Smartness

Another term often mentioned alongside common sense is *street smart*. Street smart usually refers to practical savvy in navigating real-world challenges, especially in rough or unpredictable environments. It is related to traditional common sense but with much more emphasis on shrewdness and situational awareness. One way to differentiate them is that System A common sense is a baseline ability to make quick basic sound judgements (the common knowledge that any person should have), while street smarts is like a more elaborate version of that ability, often gained by dealing with difficult, high-stakes situations, although not being System B-level. Street smarts can be seen as a System A common sense trimmed by survival skills and experiences, particularly in unfamiliar or risky situations. The trimming is more about extending the knowledge base than the thinking. A street-smart person typically knows how to read social cues, avoid dangers or scams, and get by in the real world using resourcefulness and instinct. In essence, street smarts is common sense in action under challenging conditions. It is the quick intelligence you would want if you were lost in a city at night or negotiating a deal at a shady marketplace.

Both common sense and street smarts rely on the continuous accumulation of tacit knowledge, the unspoken lessons one learns from life experience. Tacit knowledge was defined by Polanyi as knowledge that *we know more than we can tell*. It is knowledge that is not formally taught or easily written down, yet it guides our actions. Sternberg's research emphasises that tacit knowledge is a key foundation of practical common sense. For example, through experience, a manager learns the unwritten rules of motivating employees, or a traveller learns to intuitively tell which neighbourhoods are safe. These lessons become part of an internal repertoire of knowledge. Tacit knowledge is often procedural (knowing how to do something) rather than declarative (knowing that a fact is true). Someone with rich tacit knowledge in a domain might not be able to articulate all their rules of thumb, but they just know what approach will likely work because they have internalised patterns from experience. This kind of know-how is what allows an experienced teacher to handle an unruly class using simple presence and understanding, or a seasoned doctor to make a quick diagnosis that an intern would miss. Tacit knowledge for practical intelligence tends to be acquired on one's own, without the support of formal instruction, and it remains unspoken and poorly conveyed relative to its importance for practical success.

In everyday terms, basic common sense could be thought of as the System A application of the broad tacit knowledge that nearly everyone is expected to have, while street smarts is specialised tacit knowledge for navigating particular real-world contexts. Both differ from theoretical knowledge in that they are context-dependent, learned informally, and often tied to action. An academically brilliant person might know the theory of combustion, but it is System A that advices you not to pour water on a grease fire. Similarly, a street-smart individual might not compute probabilities explicitly, but they have an instinct for when someone is trying to con them, drawn from years of dealing with people. Researchers have tried to capture this knowing-how aspect through situational judgement tests and scenarios, validating that it contributes to success at work and life independently of traditional IQ. All this underscores that common sense is rooted in experience-based intuition as much as or more than abstract reasoning.

Analytical Thinking

It is illuminating to contrast common sense with analytical thinking. Highly analytical thinking proceeds from explicit principles and rigorous steps, for example deducing a conclusion from premises in mathematics or following a scientific method. System B, by contrast, is often informal and driven by context. Humans often rely on heuristics (mental shortcuts or rules of thumb) that usually, but far from always, yield reasonable results. These heuristics are essentially distilled common-sense observations. For instance, one heuristic is the representativeness heuristic: judging likelihood by how representative something is of a known pattern (which is why a person might intuitively suspect a cheating scenario if something feels off). Such gut feelings are not fool-proof or strictly logical, but they are part of our common sense toolkit that works well in typical situations. Decision scientists such as Gerd Gigerenzer have argued that simple heuristics can be surprisingly accurate and useful, dubbing them fast-and-frugal ways of reasoning that often beat out complex analyses in real-world environments. In many cases, common sense aligns with practical logic, a kind of rough-and-ready reasoning that, while not rigorous, is well suited to everyday life's requirements.

As seen above, common sense also includes commonly held knowledge, which is something static and different from thinking, which is dynamic. History is full of examples where common-sense beliefs turned out wrong under scientific scrutiny (e.g. the once-common-sense idea that the sun revolves around the Earth).

Psychological Theories

Researchers from both psychology and cognitive science have proposed various theories and models to explain what underlies common-sense thinking. Here are a few key frameworks and constructs that relate to this domain-general reasoning ability:

Practical Intelligence. Robert Sternberg's triarchic theory of intelligence breaks intelligence into analytical, creative, and practical components. Practical intelligence is defined as the ability to adapt to, shape, or select environments to meet one's goals. It involves applying knowledge to real contexts and is measured via tacit knowledge and situational judgement. Sternberg and Wagner's studies showed that practical intelligence is distinct from analytical IQ. For example, a tacit knowledge test for business managers might ask what the best way is to handle an employee problem, measuring insight gained from experience rather than academic knowledge. They found that such practical know-how could predict success better than IQ scores could. In educational settings, Sternberg demonstrated that teaching and testing for practical intelligence (e.g. asking students to solve practical problems) can identify talents that standard tests miss. Practical intelligence is thus a psychological cousin to common sense. It formalises the concept as a measurable skill set based on real-world problem-solving and wisdom-in-action.

Fluid vs. Crystallised Intelligence. There is a well-established difference between fluid intelligence (the capacity to solve novel problems, think logically and see patterns in new information) and crystallised intelligence (accumulated knowledge and facts). Common sense draws on both in different ways. When encountering a new situation, one uses CST (fluid reasoning) to interpret it flexibly (like solving a new puzzle) but also pulls heavily from crystallised knowledge of similar past situations and general world facts. In fact, the resulting common sense could be seen as an interplay of fluid problem-solving with crystallised life knowledge. A person with high fluid intelligence might learn rules quickly, but without sufficient worldly experience (crystallised content) they could still make unwise choices. Conversely, someone with vast experience (high crystallised knowledge) but very low fluid reasoning might struggle to apply that knowledge in new ways. Traditional IQ tests emphasise fluid reasoning in abstract contexts, whereas common sense emphasises applying learning to concrete reality. In other words, intelligence must be put in the service of common-sense goals and values to be meaningful.

Cognitive Flexibility and Executive Functions. From a cognitive psychology perspective, executive functions (higher-order control processes managed by the frontal lobes) are important for CST. Executive functions include abilities like working memory (holding and mentally manipulating information), inhibitory control (suppressing impulses or irrelevant info), and cognitive flexibility (shifting one's thinking or approach as needed). Cognitive flexibility, in particular, is key to applying CST across domains. It allows a person to adjust to new rules or to see a problem from multiple angles. For instance, take an example from classical thinking: imagine driving in a foreign country. One must suppress the habit of driving on the familiar side of the road (inhibition), keep track of new signage (working memory), and flexibly adapt to different traffic rules and behavioural norms. This is a very practical mix of executive skills that manifests as using your cognitive abilities to avoid accidents. Some experts even colloquially equate strong executive functioning in daily life with having good common sense: the ability to organise oneself, make sensible decisions, and self-correct. Indeed, deficits in executive function (such as in ADHD or frontal lobe injuries) might appear as poor judgement or common sense abilities. The person might know the right thing to do but fail to do it, or act impulsively against their better knowledge. Thus, while common sense is not usually defined in neuropsychological terms, its exercise likely depends on an intact and active executive system enabling us to plan, foresee consequences, and adapt strategies on the fly. This aligns with the idea that metacognitive skills (thinking about one's thinking, as executive processes do) could support better CST.

Rationality and the Reflective Mind. Keith Stanovich has argued for a model separating intelligence from rational thinking. He notes that standard intelligence (the algorithmic mind) does not guarantee what he calls rationality, the ability and disposition to think logically when it matters, avoid cognitive biases, and make decisions that align with one's goals. In effect, he is pointing to a common-sense component of cognition: being able to reason through everyday problems and not be led astray by irrational quirks. He coined the term *dysrationalia* for the failure to behave rationally despite adequate intelligence. In a sense, that could be interpreted as a lack of CST capability. To capture this, Stanovich and others have developed tests of reflective thinking (like the Cognitive Reflection Test, which checks if people can override a gut response with a more reasoned answer). These tests reveal that many intelligent people still answer incorrectly on trick questions that require basic CST because they miss obvious considerations in a rush to intuitive answers. Stanovich proposes a Rationality Quotient to sit alongside IQ, essentially measuring common-sense reasoning and decision-making quality. Although this work uses the term rationality rather than common sense, it covers much of the same territory: judgement, avoidance of illogical thinking, and prudent decision-making in real-world contexts. The findings reinforce that rational common sense is an independent cognitive domain, one that can and should be measured and taught, because it impacts life outcomes significantly.

Each of these four frameworks, practical intelligence, fluid/crystallised intelligence, executive function, and rational thinking, sheds light on aspects of the elusive common-sense ability. They suggest that common sense is a composite of cognitive skills: part experiential knowledge, part flexible problem-solving, part self-regulation and rational judgement. No single theory (yet) fully encapsulates common sense, but together they paint a picture of a mental toolkit that allows individuals to learn from experience, adapt to diverse situations, and make sound decisions that pure analytic intellect or rote knowledge alone might not guarantee.

Scientific Thinking

The connection between CST and scientific thinking is deeper than it might first appear. Although science is often viewed as a domain of formal methods, systematic experimentation, and mathematical abstraction, its origins and everyday practice are rooted in a refined version of common-sense reasoning. Historically, the scientific method did not emerge out of nowhere as a set of strict procedures. Rather, it evolved as a codification of certain insights and practices that, when used consistently and critically, tended to produce reliable knowledge about the world. In this sense, the methods of science can be seen as being based on disciplined and institutionalised common sense.

Throughout the history of science, figures like Bacon, Galilei, and later philosophers such as Popper sought to articulate systematic ways of thinking that corrected the natural biases of ordinary human reasoning. Yet even in these efforts, the underlying goal was to refine and formalise the types of inference that had long been employed informally: making careful observations, testing explanations against evidence, and preferring simpler, more coherent accounts over complicated or ad hoc ones. Popper's notion of falsifiability, for example, can be understood as a formalised expression of the common-sense idea that a good theory should be testable against reality and should risk being wrong. Likewise, Kuhn's analysis of scientific revolutions reveals that paradigm shifts occur not purely through formal derivations but through communities of scientists gradually finding that older models no longer make sense of emerging phenomena. What Kuhn termed a crisis in a scientific paradigm is essentially a moment where the structured and institutionalised common sense embodied in a scientific worldview begins to break.

Moreover, scientific reasoning often relies heavily on what might be called educated intuition, especially at the frontiers where formal theory runs thin. Nobel Prize laureate Feynman emphasised the importance of *guessing* in scientific discovery, followed by rigorous testing. This sequence, first the informed conjecture and then the empirical scrutiny, mirrors the flow of CST: proposing plausible explanations based on prior experience and correcting them in light of reality. In some ways, the scientific method represents an institutionalisation of critical CST, where practices such as peer review, replication, and methodological scepticism serve to protect the community from individual biases while maintaining the adaptability and pragmatic spirit of common-sense reasoning.

Importantly, even the methods taught in scientific education rely on foundational cognitive moves that are not alien to everyday reasoning but are sharpened versions of them: observing patterns, inferring causes, proposing tentative explanations, and adjusting beliefs based on new information. Scientific literacy, at its best, is not the memorisation of facts but the cultivation of a refined common-sense view of nature or society. In this light, science does not replace or transcend common sense but seeks to correct its errors systematically while preserving its most vital strength: the capacity to make reasonable judgements in the face of uncertainty. Seen from this perspective, the history and philosophy of science can be read as a long project of making the common sense of the world more precise, more reliable, and more self-correcting without losing its essential character as a deeply human way of knowing.

Pólya's book *How to Solve It* offers an example of how refined CST underlies even the formal disciplines of mathematics and logical problem-solving. Although the book is situated within the context of teaching students how to tackle mathematical problems, the strategies Pólya outlines are recognisable as structured forms of intuitive reasoning, not rigid formal algorithms. Techniques such as guess-and-check, considering a simpler problem and working backwards from the goal, and draw-a-figure reflect deeply human, experience-based ways of thinking that predate and transcend formal schooling. Pólya's heuristics are essentially an attempt to make tacit common sense explicit and systematic, turning everyday problem-solving instincts into teachable methods.

Importantly, Pólya never claimed that his heuristics guaranteed success. Rather, he acknowledged that intelligent problem-solving is an art that requires judgement, flexibility, and an ability to adapt strategies to the context at hand. In this way, his framework mirrors the nature of CST. It is not about slavish rule-following but about intelligently navigating uncertainty, using a combination of accumulated experience, flexible reasoning, and practical intuition. Where formal logic demands strict proofs and certainty, Pólya's heuristics embrace provisional, adaptive reasoning. A form of critical CST applied within a disciplined domain.

Seen in this light, *How to Solve It* represents the same spirit found in the history and philosophy of science: the effort to refine, discipline, and enhance common sense without discarding its core strength, which is its capacity to make plausible, adaptive judgements under incomplete information. Pólya's work thus stands as a bridge between everyday reasoning and systematic inquiry, illustrating how even the most revered intellectual practices are, at their heart, elaborations on the general human faculty for seeing, guessing, testing, and adjusting that constitutes true common-sense thinking.

In modern cognitive science and educational theory, the ideas that Pólya anticipated have been developed into concepts such as adaptive expertise and situational judgement. Adaptive expertise refers to the capacity to apply knowledge flexibly and creatively across new situations, rather than merely reproducing learned routines. The researchers Hatano and Inagaki have distinguished between routine experts, who excel at applying standard procedures in familiar contexts, and adaptive experts, who can innovate and modify their methods when faced with novel challenges. This distinction mirrors precisely what Pólya sought to cultivate: not the mere execution of formal procedures, but the intelligent adaptation of reasoning strategies to fit the problem at hand.

Similarly, the study of situational judgement (the ability to size up ambiguous, real-world situations and choose effective courses of action) captures another dimension of refined common sense. In fields as diverse as management, education, and medicine, situational judgement tests are used to measure practical reasoning skills that go beyond pure analytical intelligence. These skills depend not just on knowing explicit rules, but on having an intuitive grasp of context, relevance, and human factors, all hallmarks of CST.

Pólya's heuristics, when viewed through this lens, can be seen as early efforts to scaffold the development of adaptive expertise and situational judgement in students. Rather than seeking to replace intuition with formulaic logic, he sought to elevate intuition to a higher level of conscious skill, fostering the ability to recognise patterns, shift perspectives, and improvise solutions based on a practical understanding of the structure of problems. In doing so, Pólya underscores a deeper truth: that the highest forms of expert thinking are not divorced from common sense, but are its disciplined extensions. They represent the culmination of human practical intelligence, honed through reflection, abstraction, and application, but still fundamentally rooted in the general cognitive capacities that allow us to navigate the ordinary complexities of life.

Thus, whether in the foundations of scientific inquiry, in the heuristics of problem-solving, or in the emergence of adaptive expertise, it becomes clear that CST is not a primitive stage to be outgrown, but the enduring bedrock upon which a lot of reasoning rests. Even the most refined intellectual practices remain, at their core, extensions of the general human ability to perceive, infer, and adapt to the complexities of the real world. Far from being a lesser form of intelligence, common sense constitutes the deep architecture of rationality itself. A universal resource that continues to shape both our everyday judgements and our greatest achievements.

Professional Domains

CST is highly useful in many professional and real-world domains, sometimes even more so than technical expertise. Here we explore how this general reasoning ability manifests in a few specific areas: consulting, education, and leadership.

In the realm of leadership, CST is often mentioned as an essential trait, sometimes under labels like practical wisdom, good judgement, or business sense. Many high-profile leaders are celebrated not just for their technical or creative brilliance, but for their down-to-earth judgement and ability to make sensible decisions amid complexity. In management theory, there is a growing realisation that effective leadership requires using common sense to guide decisions in addition to data and analyses. This includes understanding people's motivations, balancing short-term and long-term considerations, and knowing when an action simply feels right or wrong on a human level.

Recent research into common-sense leadership has tried to unpack what this means. In a qualitative study of senior executives, participants described common-sense leadership as a multi-faceted approach requiring flexibility, practical decision-making, and a moral compass. Leaders felt that common-sense decision-making often involves ethical judgement, for instance sometimes superseding organisational performance and profitability to do the right thing. In other words, a leader with common sense knows that purely following spreadsheet logic might not be wise if it violates basic human principles. These leaders emphasised integrating people considerations with business needs and using plain logic in communication. Common sense in leadership might manifest as an ability to cut through jargon and complexity to communicate a clear vision and to make decisions that align with both practical reality and core values. It is the opposite of getting lost in abstract strategy while ignoring on-the-ground facts.

It is said that good leaders have a gut instinct that filters all the data and gives them the answer, highlighting the role of intuitive judgement. That gut instinct often amounts to refined common sense, the product of deep experience and a sense of what usually works or fails in human organisations. Leadership training programs today often include scenarios and simulations to develop this judgement. Concepts like situational leadership implicitly rely on common sense: the leader must assess the situation's unique context and apply the appropriate style. In entrepreneurship, investors often say they back founders who have great business sense, a colloquial term for a form of practical intelligence. This can mean knowing your customer on a mental level or being able to pivot strategy when market feedback dictates rather than dogmatically sticking to a previous plan.

One interesting notion is *critical* common sense, denoting an advanced form of CST in leadership. This idea, discussed by some leadership scholars, suggests that while common sense is generally good, leaders sometimes need to question naive common sense, especially in unprecedented situations, effectively applying common sense to common sense itself. For example, it might be considered common sense in an organisation to always do things a certain way, but a wise leader knows when that old common sense no longer applies and a new approach is needed. This merges creativity with common sense. Leaders who excel seem to know when to trust the usual rules of thumb and when to break from them, a synthesis of practical intuition and adaptive thinking.

Thus, leadership across sectors consistently calls for keen CST. Whether it is a school principal handling a crisis or a big company CEO setting a strategy, those regarded as wise leaders tend to display sound judgement, adaptability, and the ability to relate decisions to real-world impacts. They use common sense when algorithms or pure theory run out. A common-sense approach in leadership is an often overlooked but vital complement to analytic and visionary skills. It grounds leadership in reality and earns trust from others who feel the leader gets it on a fundamental, human level.

Top management consulting firms (like McKinsey, BCG, and Bain) pride themselves on hiring bright analytical minds, but they also emphasise common-sense problem-solving as a critical skill. In fact, much of the famed consulting problem-solving approach can be seen as structured common sense. Anyone who has ever done a job interview with one of the tier-one management consulting firms knows how much emphasis they put on this kind of problem-solving ability. Consultants break down complex business problems into simpler parts and apply logical reasoning, but they also rely on broad, cross-domain thinking and practical judgement. Many case interview experts note that many, if not most, cases can be solved with common sense and some very basic business concepts. In other words, beyond knowing specific formulas, a candidate needs the general common-sense ability to sanity-check numbers, identify what truly matters, and apply simple logic to business scenarios. For example, if a case question asks why a company's profits are down, a common-sense approach might first separate potential causes (lower revenue vs. higher costs). This is a straightforward step that any reasonable person might take, definitely without an MBA.

Consultants themselves acknowledge the importance of intuitive reasoning. McKinsey consultants often tackle unfamiliar decision problems by remembering that, as with any other problem, common sense goes a long way in analysis. This points to a balance: common sense provided the initial guidance (basic, sensible factors to consider), and then a formal tool added clarity. In everyday consulting work, a lot of the heavy lifting is done by asking common-sense questions: What would a customer likely prefer? Where are we obviously losing money? Is this plan practically feasible on the ground? Such questions draw on a broad understanding of human and economic behaviour, not just textbook frameworks. Successful consultants are often those who combine analytical prowess with a strong dose of practical intuition. They can crunch data but also have a feel for which direction will yield a useful answer. The ability to see the forest for the trees, simplify complexity, and apply everyday logic to business problems is what makes consulting recommendations both intelligent and implementable. After all, a solution that looks brilliant on paper but violates common sense will not fly in the boardroom. Management consulting will be discussed further in Chapter 6.

In education, the role of common sense is a subject of frequent commentary. Teachers often observe that students may excel at exams yet falter at applying knowledge to real-life tasks, which is essentially a lack of practical reasoning or common sense. Educators therefore strive to develop not just students' academic skills but also their critical thinking and problem-solving skills, essentially trying to inculcate CST and judgement. For instance, a science teacher might push students not just to memorise formulas but also to use common-sense thinking to estimate answers. Does 500 kg seem like a reasonable weight for a balloon? Probably not. There is an increasing focus on real-world problem-based learning to bridge this gap. Some curricula include what can be called practical intelligence for school, lessons where students must use reasoning in everyday scenarios (budgeting a project, interpreting current events logically, etc. These efforts are informed by research like Sternberg's, which showed that teaching practical thinking strategies can improve students' adaptive skills.

On the flip side, educational psychologists have documented a phenomenon where high academic achievers may lack common sense. This can manifest as brilliant students who struggle with basic decision-making outside the classroom. The stereotype of the absent-minded professor embodies this: someone who is a genius in their field but forgets to lock their door or manage daily tasks. While it is a stereotype, it has some basis in observation. The structure of formal education rewards abstract reasoning and memory more than practical judgement, so it is possible to advance far academically without having any real-world problem-solving skills. To counter this, some universities include experiential learning (internships or team projects) where students face messy, unstructured problems that demand common sense.

There is also recognition that common sense has a cultural dimension in education. Educators talk about instilling common-sense values like courtesy, responsibility, and healthy scepticism. A student with common sense not only knows facts but also understands which facts apply in a given situation and has the good judgement to act on them appropriately. Developmental psychology shows that children begin forming common-sense understandings of both physics and psychology from a very young age. The role of schooling in general should be partly to enrich and fine-tune these common-sense notions, since quite often naive common sense is incorrect scientifically, and partly to ensure students carry their reasoning abilities outside the school context. So educators should view CST as an important life skill. The ultimate goal is to produce graduates who are not only knowledgeable but can also think on their feet and navigate novel situations wisely. But it is not only about navigating situations or generating solutions to problems, i.e. some active productive mode. It is as much the ability to recognise a good solution when one is presented. As an old saying, often attributed to Alfred North Whitehead, goes: Education is useful just in so far as it helps students to use their common sense.

Next, we will examine two domains where common-sense thinking (CST) is employed with distinct contexts and objectives: scientific research and management consulting. Each will be analysed with a focus on the personal innovator. Which methods are helpful for making personal improvements, discoveries, and innovations? The key message from these three chapters is that common-sense thinking can be trained. It is not a fixed, innate ability. While research suggests that general intelligence (IQ) is partially heritable, often estimated at around 50%, this figure is highly debatable. IQ is a polygenic trait, influenced by many genes, and studies typically rely on twin comparisons. Yet, even twins raised apart often share some environmental commonalities. Moreover, performance on IQ tests and similar can be significantly improved through training. One example, related to such tests, is the Swedish national scholastic aptitude test (*högskoleprovet*), taken by a large fraction of prospective university students and that measure general study skills and aptitude rather than subject knowledge. It has been shown to be possible to significantly increase the results of the test by training beforehand and retake the test. From mediocre, a prospective student can often rise to star-level just by appropriate training.

Most people never approach their full CST potential unless they actively invest time and effort in developing it. This fact alone should motivate readers to begin cultivating their CST abilities. In truth, not only the next two chapters, but the entire book is concerned with CST in one form or another. For instance, at the author's alma mater, KTH Royal Institute of Technology, all programmes are in engineering. However, the CST skills students develop as a by-product are valued across diverse industries, from financial institutions to management consulting. Though these CST skills are rooted in engineering and represent only a subset of the broader CST spectrum, they are substantial enough to make KTH graduates highly attractive even for roles far outside engineering.

5. Scientific Thinking

As we have seen, Design Thinking is widely known as a creative, human-centred approach to problem-solving. What is less often discussed is how much it has in common with the scientific method and how research techniques can bring greater rigour to the design process. In fact, DT can be seen as the scientific method adapted for the purpose of creating products, services, and experiences. This perspective is especially powerful for personal innovators aiming to boost personal efficiency and creativity. By treating one's ideas and prototypes as hypotheses and experiments, even a personal innovator can systematically learn what works. This chapter explores how scientific and research methods, particularly hypothesis testing and various types of validity (internal, external, construct, content, ecological, face, etc.), can be applied to the convergent stages of DT (Define, Prototype, and Test). We will show how anyone can use elements of scientific inquiry to make their personal design efforts more insightful and effective, without sacrificing creativity. The chapter does not assume the reader to be familiar with formal research methodology.

Design Thinking and the scientific method share a core cycle: observe, hypothesise, experiment, and learn. Both approaches value iterative improvement based on evidence. The key difference is context. Where the classic scientific method might test hypotheses with microscopes or in labs, DT tests ideas in the messy realm of human behaviour and everyday life. DT expands the scientific method to include empathic observation of people's needs and emotions, yielding qualitative insights as the basis for hypotheses about solutions. As a result, Design Thinking often begins with exploring what *is* (discovering user needs) before hypothesising what *might be*.

In practical terms, this means that before a designer defines a specific problem or solution to test, they spend time empathising, observing and interviewing users to gather data. This parallels how a scientist makes preliminary observations before formulating a hypothesis. Once initial insights are gathered, the designer defines the core challenge and forms a hypothesis about how to address it (for example, "I believe feature X will improve the user's productivity"). This hypothesis-driven mindset in design is directly inspired by the scientific method's emphasis on falsifiable propositions to be tested. The next steps, prototyping and testing, are essentially experiments. Just as a scientist sets up an experiment to test a hypothesis under certain conditions, a designer creates a prototype (an experiment in tangible form) and evaluates it with users to see if it confirms or refutes the initial assumptions. If the evidence doesn't support the idea, the design is iterated or even abandoned, much as a scientist would reject a hypothesis that data fails to support.

What Design Thinking adds to this scientific cycle is a strong focus on human factors and an openness to redefine the problem itself. Rather than starting with a rigid hypothesis in isolation, DT encourages co-creating the hypothesis through user insight. Scientific rigor tells us *how* to test, while DT guides us on *what* to test by ensuring we frame the right problem. By combining the two, individuals can systematically test their way forward toward innovative solutions, all while staying grounded in real user needs.

Adopting a hypothesis-driven approach in personal design projects means articulating your assumptions and then actively trying to validate or invalidate them. This approach has been popularised in the start-up world (e.g. the Lean Startup method), but it applies just as well to personal efficiency and creative endeavours. Instead of just brainstorming and implementing an idea, you frame it as an experiment: *If I implement idea X, I expect outcome Y*. This simple shift fosters a more rigorous mindset. It forces clarity on what you expect to happen (which sharpens the Define phase) and it pre-defines success metrics or criteria (which brings scientific objectivity to the Test phase).

For example, imagine you are designing a new morning routine to boost your creative output. A traditional approach might be to try a bunch of changes and see how it feels. A hypothesis-driven approach, by contrast, would have you clearly state: “I hypothesise that if I practice free-writing for 20 minutes every morning, then my number of new ideas per week will double.” Now this personal experiment can be tested. You can prototype the routine for two weeks and measure the outcome (count your ideas). Importantly, you'd also define what it means for the hypothesis to be supported (perhaps you indeed generate at least twice as many ideas) or not. This way, you engage with your personal efficiency challenge not just creatively, but scientifically. You eliminate guesswork and hone in on what truly works for you through evidence.

Embracing this mindset aligns with what entrepreneurs call *validated learning*. As Eric Ries describes in his book *The Lean Startup*, progress is achieved by business-hypothesis-driven experimentation and validated learning, favouring real customer feedback over intuition. In personal Design Thinking, your customer might be yourself or your target users, and the same principle holds: learn from real results, not assumptions. Each prototype or test yields data, whether quantitative metrics or qualitative feedback, that either supports your idea or not. Every hypothesis that is tested has the potential to generate new insight for future rounds of your product's development. In personal terms, every small experiment teaches you something about what increases your efficiency or sparks your creativity, informing the next iteration.

Importantly, a hypothesis-driven approach encourages a growth mindset. Negative results are not failures; they are informative. Just as a scientist might celebrate a disproven hypothesis for steering research towards a new direction, a designer can view a failed test as a useful pivot point. The Design Thinking process already embraces iteration (failure in a prototype simply leads back to redefining the problem or trying a new idea). By overlaying the scientific habit of hypothesis testing, you double down on learning: you always identify *why* something failed and what assumption was wrong, which guides your next steps with greater insight than mere trial-and-error. As John Michael Greer succinctly noted, Science, at its core, is simply a method of practical logic that tests hypotheses against experience. When personal design projects adopt this practical logic, creativity is enhanced by clarity of purpose, and efficiency is enhanced by not spending time on untestable or vague ideas.

Before delving into each stage of the Design Thinking process, it is important to introduce a toolkit of research concepts, particularly different types of validity. That will help ensure your experiments in the Define, Prototype, and Test phases are yielding reliable and meaningful insights. When we talk about *rigour*, much of it comes down to validity: Are you really testing what you think you are testing? Are your results to be trusted? We will see that identifying exactly *what* is being tested at each stage (a raw idea, a feature, an entire solution, etc.) will change which validity concerns loom largest and what best practices to follow.

In scientific research, validity refers to the credibility or believability of your findings, i.e. whether the results actually reflect the truth for the question at hand. Different types of validity address different questions about your experiment's design and interpretation. By understanding these types of validity, a designer can make sure their tests of a prototype or idea are rigorous and their conclusions are sound. Below are key validity types (adapted from research methodology) and their meaning in a design context:

Internal Validity. This is about cause and effect: did the change *you* introduced really cause the outcome you observed? High internal validity means you can confidently say X caused Y without interference from other factors. In a design test, internal validity is strong if the test is not biased toward a particular result and if alternative explanations have been minimised. For example, if you test two versions of an app interface to see which is more efficient, but you always show Version A to users first, any improvement in Version B might just be due to users learning the task on Version A first (a sequencing bias). In that case, the test lacks internal validity. To improve internal validity, one might randomise the order in which users see designs or use other controls to avoid favouring one outcome. In short, internal validity asks: Is my experiment itself sound? Or could the results be caused by something else (a confound) rather than my design change?

External Validity. This concerns generalisation: to what extent will the findings hold true in other situations or with other people? In design, an experiment has high external validity if the test conditions and participants are representative of the real-world use case. For instance, testing a prototype in a controlled lab with a handful of tech-savvy volunteers might yield results that do not generalise to the broader target audience or real usage environment. Nielsen Norman Group defines external validity in UX studies as the degree to which the participants and the study setup are representative of the real-world situation in which the design is used. If you are designing a site for seniors but only young adults participate in the test, the study lacks external validity and may mislead you about true user behaviour. Likewise, if you test a mobile app on a desktop simulation, you might not capture issues that would appear on actual phones. To boost external validity, one should use participants who mirror the intended users and simulate real usage contexts as closely as possible. External validity asks: Will my results apply in the real world outside of my test?

Construct Validity. This type of validity is about whether you are measuring what you intend to measure. In research, a *construct* is an abstract concept you care about (e.g., creativity, ease of use, or stress level), and construct validity means your method of measurement truly reflects that concept and not something else. In design testing, construct validity comes into play when you define metrics or criteria for success. For example, if your goal is to improve users' engagement with an app, you need to decide how to measure engagement. Is it time spent on the app? Number of pages clicked? Self-reported enjoyment? If you choose a metric like time spent, consider whether it actually correlates with engagement or if users could spend a long time because they are confused (which would mean you are inadvertently measuring confusion, not engagement, a construct validity problem). As another example, say you prototype a new brainstorming technique to enhance your personal creativity. You measure its success by counting how many ideas you wrote down. This has some construct validity for creative output, but does it capture creativity fully? Perhaps the quality or originality of ideas is part of creativity too. If your measurement overlooks those, the construct validity is limited. Achieving good construct validity involves carefully aligning your evaluation method with the true concept of interest, often using multiple indicators. The psychologists Lee Cronbach and Paul Meehl, who introduced the construct validity concept, emphasised that one must build a *network of evidence* that the test really represents the idea in question (e.g., depression isn't directly measurable, so one measures known symptoms). In design, this might mean that if you claim a design improves learning, you should measure actual learning outcomes, not just, say, user satisfaction, unless you have evidence that satisfaction equates to learning.

Content Validity. Content validity asks whether the *content* of your test or measurement fully represents the aspects of the concept or user needs you intend to cover. In practical terms, have you tested *all relevant parts* of the design or just a subset? For example, if you are testing a new productivity system for yourself, and productivity for you means managing time, staying focused, and achieving goals, then a content-valid test should examine each of these facets. If you only track, say, time spent working, you might miss other elements like quality of work or stress levels. Similarly, when user-testing a multi-feature product, ensuring content validity might involve creating tasks that touch on each key feature or use case, not just the easiest or most obvious one. A classic instance of content validity in design is when creating a usability test scenario: if your app has 5 main functions but your test tasks only exercise 2 of them, your evaluation's content validity is low. You are not seeing the full picture. To improve content validity, list out the important elements your solution addresses and make sure your prototype evaluation includes something for each. This prevents *omitted variable bias*, where missing aspects can skew results. In short, content validity is about breadth and coverage: *Did I test all the things that matter?*

Ecological Validity. Often considered a sub-aspect of external validity, ecological validity specifically refers to how realistic and natural the experimental setting and procedure are, in relation to the real context the design will be used. In other words, does your prototype test take into account the messy conditions of real life? A study in a sterile lab may lack ecological validity if, in real life, your users are distracted, multitasking, or in a specific environment. For instance, an app designed for drivers might test well on a computer by a calm participant, but in actual driving conditions (with noise, motion, and the need to focus on the road), the usability could plummet. That's an ecological validity issue. High ecological validity can be achieved by testing in context. For example, if you are designing a kitchen tool, test it *in a kitchen*, not just via a digital simulation. The Design Thinking process encourages considering the context of use, which aligns with seeking ecological validity (simulating real-world conditions makes results more credible). However, note that sometimes a highly natural setting can reduce internal validity (you cannot control all variables in the wild). Designers need to balance these, perhaps by doing some tests in controlled settings and others as field trials. Ecological validity asks: *Is my test environment and method true to life?* The more it is, the more confidently you can trust that users' observed behaviour would repeat in their everyday environment.

Face Validity. This is the most superficial type of validity: whether a test or experiment *appears* (at face value) to be measuring what it is supposed to. Face validity is not a rigorous statistical concept, but it matters in how participants and stakeholders perceive the test. In a design context, face validity could be considered when you present a prototype or questionnaire to users. Do they *believe* it is a valid attempt to solve the problem? For example, if you are testing a new creative brainstorming method but the tasks you give to people seem unrelated to creativity, participants might not take it seriously or might be confused, a failure of face validity that can undermine your results. Ensuring face validity might simply mean communicating the purpose clearly and using intuitive, relevant tasks so that testers nod and say "Yes, that makes sense" when they see the test. It is about the credibility of the test at first glance. While face validity alone doesn't guarantee actual validity, it can influence participant engagement. A test that lacks face validity might still yield useful data, but if users feel it is off-base, their behaviour or responses could be distorted. In personal experiments, maintaining face validity might equate to being honest with yourself: Are you really testing what you think, or are you kidding yourself due to bias? Having a friend review your plan can provide a face validity check: does the experiment *seem* like it would validly address your hypothesis?

These different types of validity are interrelated. For instance, a test can have great internal validity (very controlled) but poor external/ecological validity (not realistic conditions), or vice versa. Achieving all forms of validity to a high degree can be challenging, especially in practical design settings, but the idea is to be aware of trade-offs. As Campbell and Stanley noted, internal validity is the basic minimum requirement. If a study isn't internally valid, it is basically uninterpretable. However, a highly internally valid test that is so artificial no real user would ever experience those conditions is of limited value (lack of external and ecological validity). By being conscious of what you are testing at each step of design, you can emphasise the validity aspects that matter most for that step, and later complement it with other tests to cover the rest. We will now map these validity considerations to the convergent stages of Design Thinking (Define, Prototype, and Test) to see how a personal innovator can rigorously apply them in practice.

Define

The Define stage of Design Thinking is where you pinpoint the exact problem to solve or the exact goal to achieve. In the classic Stanford d.school model, this comes after the Empathise stage: you've gathered raw observations about user needs, and now you synthesise that into a clear problem definition and a Point of View. This stage is inherently convergent. From many possibilities and data points, you are narrowing down to a focus. Applying scientific rigour at this stage means formulating that focus in the form of a testable hypothesis and ensuring your defined problem has validity in terms of being the *right* problem.

One way to introduce hypothesis thinking here is to phrase the design challenge as a hypothesis about the user's need or opportunity. For example, instead of defining the problem as *Users need a faster way to do X* (which is a statement), you might say *We believe providing Feature Y will help users do X faster*. The latter is essentially a hypothesis that can later be tested. It combines a presumed solution with an expected outcome, making it explicit what will be evaluated. This approach is sometimes called *problem hypothesis* or *solution hypothesis*. It aligns with advice from hypothesis-driven design: a *Design Hypothesis* is basically an assumption that can be proven or disproven through research. By defining your project in those terms, you give yourself a clear direction for prototyping and testing.

However, before jumping to a solution hypothesis, you must ensure you've defined the *right* problem. This is where validity concerns already creep in at the definition phase. A classic pitfall is solving a symptom while missing the root cause or solving a problem that isn't actually important to the user. In scientific terms, this is like having a poorly defined construct: your whole experiment could become irrelevant because you framed the wrong hypothesis. In personal efficiency projects, for instance, one might define the problem as "I don't have enough time to exercise", and hypothesise a solution ("I will get up an hour earlier each day"). If it turns out the real issue was not time but low energy or motivation, then even a well-run experiment on wake-up times won't truly solve the underlying problem. The *construct* of the problem was wrong. This is analogous to lacking construct validity at the problem definition stage. The defined challenge didn't represent the true factors at play.

To avoid this, the Define stage benefits from triangulating multiple insights (much as a researcher uses multiple indicators to validate a construct). In Design

Thinking, tools like persona creation, journey mapping, or the Five Whys technique help ensure that the problem statement addresses fundamental needs rather than superficial wants. Essentially, you are validating the problem definition through further inquiry: *Is this really the user's pain point? How do we know?* Sometimes, quick Define-stage experiments can be done. For example, a survey or quick user poll to verify that a certain need is common or pressing. This is a way of checking the face validity of the problem: do target users acknowledge this is a problem when you describe it? If they respond with "No, that's not really an issue for me," then you know early on that your problem definition might lack validity.

An illustrative example comes from a healthcare context: a hospital team thought doctors' burnout was due to a lack of sufficient breaks (so their initial problem definition was around scheduling and leave). But by observing and interviewing (empathising) before finalising the Define stage, they discovered a deeper issue. Doctors felt *guilty* taking breaks for personal needs. The real problem was cultural, not procedural. If the team had defined the problem incorrectly (e.g. lack of leave time) and rushed to solutions, they would have tested interventions that wouldn't actually fix burnout. This underscores the need for construct validity in the Define stage: ensure the problem statement truly captures the essence of the issue.

One strategy is to phrase the define-stage hypothesis in *if-then* form, just like a scientific hypothesis. For instance: *If we increase doctors' permitted leave (assumed cause), then their burnout will decrease (expected effect).* Then critically ask: what *assumptions* underlie this? In the hospital case, the assumption was that lack of leave is the cause, which user research revealed was false. By identifying that assumption explicitly, the team was able to test it and revise the problem definition. This approach turns the Define stage into a mini-experiment in itself: you are testing the hypothesis Is this the right problem to solve? through research. Techniques such as open-ended user interviews or observational studies during Define help provide evidence for or against your initial framing. The *credibility* of your problem definition is at stake, analogous to internal validity in qualitative terms. You establish credibility by going back to the data (user input) and possibly doing *member checks* (asking users if your understanding sounds right). From a validity perspective, the Define stage is most concerned with construct and content validity of the project's focus:

Construct validity. Are you defining the right construct (problem/goal)? Address this by ensuring your definition is grounded in real user evidence and not just your assumptions. Use multiple sources of insight. If the construct is abstract (e.g. increase user delight), break it down into observable components (speed, ease, emotional satisfaction) so you know what you are actually targeting.

Content validity. Are you considering the full scope of the problem? Check that your Point-of-View statement isn't too narrow. For example, improve on-boarding experience. Does that include both the sign-up process and first-time use of the product? If user research showed pain points in both, make sure your defined challenge encompasses all relevant parts. Otherwise, your subsequent prototype might solve one part but leave another untouched (leading to only partial success).

External/ecological validity. This is about context. Ensure the problem you define is relevant to the actual environments and populations. Designers sometimes accidentally define a problem too much from their own perspective (e.g. a young design team might frame a problem in terms that don't resonate with older users). To validate external relevance, double-check the Point-of-View statement with a diverse sample of your intended audience or stakeholders. Does it ring true across contexts? If designing for multiple user segments, you might refine the definition or decide which segment you are focusing.

By the end of the Define stage, you want a clear, evidence-backed hypothesis about *what needs to be addressed*. For an individual working on a personal project, this might mean writing down: "I am addressing Problem X because I have observed Y and Z evidence indicating this is the key issue." This statement serves as your North Star going into Prototyping. And importantly, it is something you can later evaluate (did solving X actually produce the desired improvement?). A well-defined problem with high validity sets you up for success; as the saying goes, a problem well-defined is a problem half-solved, which is why DT spends about half of the effort in reframing the problem in a Point of View. In research terms, you've ensured you are *asking the right question* before investing effort in answers.

Prototype

The Prototype stage is where ideas turn into tangible (or at least perceivable) forms that can be interacted with. Prototyping is often thought of as a creative, generative step, but it can also be approached as designing an experiment to test your hypothesis. Instead of viewing a prototype as an early sample of the product, consider it as the apparatus of your experiment: it is the vehicle through which you will test whether your defined idea works as intended. This subtle shift in mindset has big implications for how you prototype and what you pay attention to.

When scientists design experiments, they carefully plan conditions and controls to isolate the effect they want to observe. Similarly, when designers create prototypes, especially in a hypothesis-driven way, they should think about isolating the key assumption they want to test. In a personal efficiency context, if your hypothesis is "Using a time-blocking schedule will reduce my distractions", your prototype might be a new calendar layout or an app with notifications turned off. The key is to structure the prototype such that, when you later test it, you can attribute any change in the outcome to the prototype itself and not extraneous factors. In practice, for a personal innovator, this could mean standardising other conditions: e.g. you try the new schedule on weeks when the workload is similar, and you avoid introducing two big changes at once (so you are not simultaneously changing your diet and your work schedule, for example. Otherwise, how to know which caused any improvement?). This is essentially applying the principle of controlled variables from science to your personal prototype trial, bolstering internal validity.

In team design scenarios, a hypothesis-driven prototype might be very specifically focused. For example, suppose a team defined a hypothesis that adding a social-sharing feature to their app will increase user engagement. A traditional design approach might have them build a fully polished social-sharing module as a prototype. A scientific approach might ask: Do we need a high-fidelity build to test this hypothesis, or can we create a simpler experiment? Perhaps a Wizard of Oz prototype would do, where a fake share button is added and if users click it, a designer manually simulates what would happen, or even just tracks that they clicked. This could validate interest in the feature without building the back end fully. The idea is to prototype at the level of fidelity necessary to test the hypothesis, and no further. This aligns with the Lean Startup concept of an MVP (Minimum Viable Product), which is essentially an experiment in the form of a basic prototype to test a business hypothesis with minimal effort. In a personal project, this translates to not over-engineering your solution before you know if it works. If you hypothesise that changing your study technique will improve recall, you could prototype it by trying the technique for one chapter of a textbook (a small-scale prototype of the method) rather than changing your entire semester's approach at once. This way, the prototype remains a low-cost, learn-fast experiment. The validity aspects during prototyping are:

Internal validity. Since internal validity is about ensuring the experiment isn't biased, think about how your prototype might inadvertently bias test results. For instance, if you prototype two versions of a design to compare (A-B test), make sure each version is presented fairly. Nielsen recommends counterbalancing order in such within-subject comparisons (half of the users try A then B, others B then A) to avoid order bias effects. If you are testing prototypes sequentially yourself (Week 1, I try Method A, Week 2, Method B for personal productivity), be aware of learning or fatigue effects (by Week 2, you might perform better simply because you are more practiced, not because Method B is superior). You might mitigate this by extending the trial or doing multiple alternating rounds. Additionally, ensure the prototype focuses on one big change at a time, if possible. If you change many variables in one prototype (e.g., new interface + new pricing model + new branding all in one go), and it yields better results, it is hard to know which change was the active ingredient. This is like an experiment with multiple independent variables. Not inherently bad, but it complicates interpretation. Often, early prototypes explore *breadth* (many changes) to get a general sense, but later prototypes might zero in on one change at a time for clarity. Being conscious of this helps balance creativity (trying bold ideas) with rigour (testing them in a decipherable way).

External/ecological validity. A prototype is, by nature, a simplification of the final product or solution. This can threaten ecological validity if the prototype is too abstract. For example, a paper mock-up of a mobile app can test layout and perhaps flow (and is great for quick user feedback), but it cannot simulate the timing delays, the feel of tapping, or the context of use (like sunlight glare on screen, or interruptions from a phone call) that a real app would have. So, what you learn from a paper prototype might not fully carry over to the field, i.e. limited ecological validity. To address this, as you refine hypotheses, you should also increase prototype fidelity in areas that matter. If your hypothesis is about the usability of navigation, a clickable digital prototype might be needed to get realistic behaviour. If your hypothesis is about whether users would even be interested in a feature, a simple storyboard might suffice because you are mainly after their *reaction*, which doesn't require full realism. Thus, match prototype fidelity to the hypothesis. It is common to start with very low-fi prototypes (sketches, storyboards) to test broad conceptual hypotheses (high content validity, covering all ideas, but low realism), then move to higher-fi prototypes (interactive, functional) as hypotheses get more specific and require more construct validity in measurement (like timing user tasks, which needs a working system). In each step, consider if the prototype environment is close enough to reality for the test at hand. Sometimes, designers even take prototypes *into* the real context. For example, a service design might be prototyped by role-playing in a real store. This boosts ecological validity at the expense of control. One memorable example: when IDEO was redesigning hospital patient experiences, they prototyped by simulating a hospital environment and even role-playing as patients and staff, to test ideas in a setting that felt real. The same technique has been used by Openlab students when prototyping concepts in their Master's course. This sort of prototyping blurs with testing, but it shows how thinking like an experimenter means sometimes you bring the prototype to the context instead of users to the prototype.

Construct validity. This might sound odd since construct validity usually refers to measurement, but here we can interpret it as ensuring the prototype actually embodies the concept you want to test. For instance, if your concept is a more social experience but your prototype only includes a single isolated feature (say, a comment box), is that truly representing social experience? Maybe not fully. Construct validity in prototyping could also mean that testers/users *perceive* the prototype as intended. If you design a new workflow to reduce stress, but the prototype (perhaps a mock interface) is so rough that users find it confusing or ugly, they might report high stress from using it, not because the concept is bad, but because the prototype quality interfered. That's like a measurement error. One way to guard against this is to communicate to users the purpose (We are testing the idea, not the visual design) and focus on the aspects relevant to the hypothesis. Or ensure the prototype is just polished enough in the critical areas that users can imagine the rest. Essentially, make sure the prototype allows you to observe the indicators of success you defined. If testing for faster task completion, a prototype needs to be functional enough to actually let users complete tasks to measure time. If it is too incomplete to do that, you are not measuring what you intended, i.e. a construct validity issue for the test.

Content validity. During prototyping, content validity is about exploring enough different ideas or variations so that you are not leaving out a potentially better solution. In early Design Thinking phases, we often generate many ideas (divergent ideation) and then select some to prototype. If you only prototype one concept, you might miss alternative approaches that target different aspects of the problem. The DT method suggests prototyping multiple options when possible, to compare and learn (this is like testing multiple hypotheses in parallel). For a personal project, say you want to improve your sleep, and you have two hypotheses: one is that a new bedtime routine will help, another is that changing your diet will help. You might prototype both interventions separately (on different weeks, or split aspects of your routine) to see which yields better results. That way, you cover more content in the design solution space before converging. It is like casting a wider net experimentally. Once you have a promising prototype, content validity also means making sure the prototype addresses all parts of the user's needs identified. If your problem definition includes speed *and* accuracy, check that the prototype is crafted to improve both, or consciously decide it focuses on one and you will tackle the other later. Otherwise, your test might show partial success and you might mistakenly think the whole problem is solved when only part was.

During prototyping, it is also valuable to plan how you will test the prototype later. This is designing the experiment protocol in parallel with the artefact. For instance, if you are making a prototype app, decide in advance what tasks you will ask users to do with it, or what data you want to collect. This ensures you build in any necessary logging or feedback mechanism. It also helps keep the prototype as simple as possible: *build only what you need to answer the question.* This focus can ironically boost creativity because it reduces the noise and lets you try bolder ideas on the core hypothesis. As the Furthermore team notes, hypotheses help focus teams and avoid protracted debates. Instead of arguing whose idea is best, you prototype both and test. For a personal innovator, this means your inner critic or indecision can be tamed by saying "I'll just prototype options A and B and see which actually works better," rather than getting stuck in analysis paralysis.

So, treating prototyping as experiment design encourages you to be *systematic in your creativity*. You are still free to innovate and tinker, but you always circle back to: *What am I trying to learn with this?* It ensures that by the time you reach the formal Test stage, your prototypes are ready to yield actionable insights, and you have not painted yourself into a corner with an untestable solution. The prototype stage sets the stage (pun intended) for the actual testing, so let us move to how to execute tests with rigour and interpret results through the lens of validity.

Test

The Test stage is where the rubber meets the road. You expose your prototypes to users (or to realistic conditions) and gather feedback and data. This stage is explicitly an experiment: you observe what happens when users interact with the prototype, and you measure outcomes against your expectations. To maximise learning, one must conduct tests in a way that yields trustworthy results. This is where all those types of validity we discussed become practically important. We will look at how to structure the testing sessions or trials, how to collect data (qualitative or quantitative), and how to ensure that the conclusions you draw are warranted. The aim is to avoid false positives (thinking an idea works when it really doesn't) and false negatives (dismissing a good idea due to a poor test setup), by being methodical.

Internal Validity. To have confidence in test results, eliminate or reduce biases and confounds in how you run the test. If you are conducting usability testing with users, be wary of how your facilitation might influence them. Something as simple as phrasing a question can bias responses. For example, asking Did you find feature X helpful? suggests that they should have found it helpful, potentially prompting a more positive answer than they truly feel. This is analogous to the example where asking Have you found the checkout difficult, primed participants to look for difficulties. A less leading approach could be: How did you find the checkout process? or better, if possible, observe without asking and only probe neutrally (Tell me what you think about this step). If you are testing your own personal behaviour (like trying a new routine), internal validity issues include things like the placebo effect or expectation bias. You might perform better just because you *expect* the new routine to help. To combat this, try to be blind to outcomes while testing (e.g., record data and only analyse after the test period, or involve a friend to objectively evaluate). While you cannot completely double-blind your own life experiments, you can put safeguards like not changing other factors simultaneously, as mentioned earlier, and being honest in recording outcomes.

For more structured testing, consider using randomisation where applicable. If you have multiple users and multiple prototype versions, randomly assign which user sees which version (or randomise order as discussed). This distributes any personal biases or abilities evenly, strengthening internal validity. In a personal A-B test (say you want to compare two diet plans for a month each), you obviously cannot randomise sequence (one has to come first); instead, you might randomise other aspects, such as which days of the week follow which plan if that's feasible, or randomly decide on the order ahead of time to avoid picking the order that subconsciously favours your preferred diet. These techniques ensure you are not stacking the deck.

Another threat to internal validity is confounding variables, i.e. other things changing that could affect outcomes. In user testing, confounds could be differences in user background knowledge, differences in how much help one user got from the moderator, etc. Good test protocol design mitigates these: give all participants the same introduction script, avoid answering help questions in a way that gives away solutions (or if you do assist one participant, note it and perhaps assist others similarly to keep consistency). In self-testing, confounds are even trickier since…life happens. If during the first week of trying a new workflow, you also happened to get new office equipment, that could confound results (was it the workflow or the ergonomic chair that improved your productivity?). Document any notable events during your test period and consider their impact. If something major occurred, you might decide to rerun the test or extend it to average out the disturbance.

External and Ecological Validity. When conducting user tests, try to make the test scenario as realistic as possible for critical factors. This is why contextual inquiry and field studies are gold standards in user research. Observing users in their natural environment using the product yields highly valid insights about real usage. If you cannot test in the wild, at least simulate key aspects. For example, if you are testing a fitness app, have users perform a mock workout with it, not just click through screens sitting at a desk. One approach to boost ecological validity is shadowing or diary studies: instead of one lab session, you might give users a prototype to use for a week in their daily lives and have them log issues. This sacrifices some control (you cannot see exactly what they do each moment) but gains realism and can uncover issues that only appear over time or in specific contexts (e.g., the app fails when there is no internet, something a short lab test might miss). In personal projects, ecological validity means testing your new habit or solution under normal conditions. If you only practice your new morning routine on vacation, success there might not translate to work days. Test it on a normal workday to see if it actually holds up when life is busy. Another aspect: include a variety of scenarios in your test. For a design, that might mean testing easy and hard tasks; for personal efficiency, try the method on both a good day and a stressful day. That helps ensure the result isn't a context-specific fluke.

Construct validity. By the time you are testing, you should have decided how to measure success or gather feedback. Ensure your measurement actually reflects the goal. We touched on this earlier, e.g., measuring engagement by time spent might be misleading. So perhaps you combine metrics: time spent *and* a satisfaction rating, to distinguish long usage because it is engaging from long usage because it is confusing. This combination gives a more construct-valid picture. In user tests, common measures include task completion rate, error count, time on task, and subjective ratings (like SUS – System Usability Scale questionnaires). Each measures a slightly different construct (usability has many facets). Many practitioners use standardised questionnaires post-test (such as SUS or NASA-TLX for workload), which have the benefit of being validated instruments. That covers construct validity because those tools were developed to measure exactly those concepts. If you create your own survey, pilot test it to ensure questions are interpreted correctly (face validity and content coverage). For qualitative outcomes, construct validity equates to *interpreting feedback correctly*. If a user says “I don’t like this feature,” probe why. Their reason tells you the construct. Without probing, you might guess the wrong construct (maybe you think they dislike the concept, but actually they just didn’t notice the button; a discoverability issue, not a fundamental concept rejection). Guba and Lincoln’s idea of credibility is relevant: ensure your interpretations match the users’ intended meaning (sometimes checking back with them, or having a colleague independently code the feedback). In personal testing, if your measure is something like productivity felt (a self-score of 1–10 each day), be mindful of mood and bias. You might supplement self-reports with objective measures (e.g., number of tasks completed). If both align, you have more confidence (convergent validity). If they diverge, investigate why. Maybe your definition of productivity needs refining.

Content validity. Plan your test tasks or test days to cover all critical use cases. For a broad-scope solution, one round of testing might not hit everything, so consider multiple rounds focusing on different aspects or a test that includes multiple tasks. For example, if your new app has features A, B, and C, don’t only test feature A extensively and ignore B and C; allocate some tasks to each (or if doing an open-ended test, ensure the scenario triggers the use of each feature). A method to ensure coverage is to use a matrix of user needs vs. test tasks to check that each need is addressed by at least one task. In personal experiments, content validity might mean if you are evaluating a new diet for health, you look at multiple indicators: weight, energy levels, blood levels, etc., not just one, to truly cover health. Or if you aim to improve creativity, you test both idea quantity and perhaps quality (maybe by later reviewing which ideas you pursued). Covering all bases prevents a narrow view of success. It also helps identify side effects: maybe your prototype solves one problem but creates another. If you only test the intended benefit, you might miss the new problem. So include general questions like What did you dislike or find difficult? to catch issues outside your main metrics.

Face validity. While running the test, consider participants’ body language and engagement. If you see that a user is visibly confused about the test’s purpose or the prototype’s purpose, take note. This often indicates a face validity issue with either the prototype or the test scenario. You might need to adjust on the fly (nothing wrong with a brief clarification, as long as you note that you did so). In longer studies or personal trials, maintaining motivation is important. If something about the test process is onerous or artificial (e.g., filling a daily five-page questionnaire), users or you might stop adhering to it. Better to have a simpler measure that people will actually do, a consideration balancing ideal data and practicality. A test with slightly less detailed data that is completed by all participants is more valid than a perfect data scheme that half of them abandon.

Reliability. Though the question focuses on validity, a quick word on reliability, i.e. the consistency of results. If you run the same test again, would you get similar outcomes? Reliability is improved by clear procedures and a sufficient sample size or repetitions. For a personal innovator, this might mean trying the prototype multiple times to see if results hold (not just one day), and for user studies, testing with more people to see patterns beyond individual quirks. Reliability underpins validity: a wildly variable result is hard to interpret. A rule of thumb says that testing with five users can find most usability issues. However, for measuring metrics, you often need more for statistical confidence. Decide based on what is at stake. If it is your personal project, you might iterate after a handful of trials because you can quickly try again. If it is a larger decision (like launching a product change), you may invest in more participants or an A-B test with hundreds for confidence.

After conducting tests, you will gather your findings and map them back to your hypothesis. This is when you identify what exactly was validated or invalidated. Perhaps your hypothesis was partially supported. For example, users liked the concept (qualitative feedback positive), but the prototype didn’t improve task time as expected (quantitative measure flat). This nuance is gold: it means the idea has promise (face validity with users), but execution needs work (maybe usability issues to fix). Or vice versa: the data showed improvement, but users didn’t *feel* more satisfied, indicating a disconnect between objective and subjective outcomes, again something to explore. In essence, a scientifically-minded designer doesn’t just ask Did it work or not?, but rather *In what ways* did it work or not, and *why*? This echoes the idea of learning from experiments rather than simply passing or failing. Each test result, positive or negative, is fed back into the Design Thinking cycle: you might redefine the problem or criteria (if tests reveal new insights about what users really care about), adjust the prototype, and test again. This iterative loop is exactly how scientists refine theories over experiments, and it is how design iterations converge towards truly effective solutions.

Finally, consider documenting your test process and results. In personal projects, keep a journal of what you did and what happened. This practice not only helps memory (so you don't later misremember how well something worked, a common but very avoidable bias) but also allows you to spot trends over multiple experiments. In a team, documentation of test protocols and findings increases the dependability of the research, i.e. someone else can review the audit trail and see that your findings are backed by systematic observation, not just anecdote.

Mapping Research Techniques

Throughout Define, Prototype, and Test, we've implicitly mentioned various research techniques. Here we explicitly map common research and scientific methods to each Design Thinking stage and discuss how to adapt them for personal or small-scale use. This serves as a quick reference to choose the right tool for the innovation.

Empathise Techniques

User Interviews. One-on-one interviews (structured or unstructured) are a primary way to collect qualitative data on user needs. Adaptation: If you are solving a personal issue, interview yourself via reflection or ask friends/family about their similar experiences to avoid blind spots. Ensure you ask open-ended questions and avoid phrasing that leads interviewees to confirm your existing assumptions (maintain openness to discover the real problems, enhancing the credibility of your findings).

Observation and Ethnography. Watching people in context (or observing your own behaviour as if you were an outside scientist) can reveal needs that users themselves might not articulate. For example, shadow someone going through the process you aim to improve (or record yourself doing it). This grounds your Define stage in real-world evidence, increasing ecological validity from the start.

Surveys and Questionnaires. These capture a broader sample of user attitudes or the frequency of problems. For personal use, you might not deploy a survey, but you could gather input from an online community or do a quick poll among peers (Have you experienced X? How do you handle it?). When designing surveys, use simple, clear questions (tested for face validity) and ensure you cover all relevant topics (content validity) without making it too long (or users will not respond reliably).

Data Analysis. If quantitative data exists (website analytics, personal trackers, etc.), analysing it can help define the problem (e.g., I spend on average 2 hours on social media daily. Maybe that's what I need to address in my efficiency plan). In design, if you have product data, look for pain points (where users drop off, etc.). This is like an observational study at scale.

Define Techniques

Point-of-View Statements and How-Might-We Questions. These are classic Design Thinking outputs to frame the problem creatively. To add scientific rigour, treat them as hypotheses to verify. For instance, a POV statement *Users [need] a faster way to on-board because [insight] they get frustrated with lengthy forms* can be checked by confirming that frustration exists and is due to forms (perhaps through the research above). The HMW questions can be prioritised by which seem most supported by evidence.

Theory of Change diagrams or Logic Models. Borrowed from social science and programme evaluations, these map out how a proposed solution is supposed to lead to desired outcomes (listing assumptions along the way). Creating a simple logic chain for your problem forces you to identify assumptions that you can later test. For example, providing mentorship (solution) → increased student confidence (immediate outcome) → higher graduation rates (long-term outcome). Each arrow is an assumption. This technique ensures you have internal consistency in your defined problem-solution reasoning, and highlights which link to test first.

Prototype Techniques

Low-Fidelity Prototyping. Sketches, paper prototypes, storyboards, and role-playing. These are quick and cheap, great for testing concept understanding and basic interactions. Adaptation: If your domain isn't a physical product (e.g., a new workflow or service), you can still paper-prototype by mapping out steps on sticky notes or index cards. For instance, when designing a new morning routine, you might draw a timeline of activities and simulate it mentally or with a friend acting as a user. Low-fi prototypes are best for exploratory tests focusing on gross feedback (face validity: do people get the idea, do they foresee value or issues?).

Wizard of Oz. A prototype technique where you fake the functionality behind the scenes. E.g., show a user an app interface that appears automated but you are actually controlling responses manually. This is useful to test reactions to a feature without fully building it. For a personal innovator, consider *role-playing* scenarios: want to test a difficult conversation or habit, walk through it in a pretend setting to see what might happen. The key is that the user (or you) experiences something *as if* it were real. This technique can yield very insightful qualitative data early, checking if the concept resonates and how users behave, while maintaining some internal validity (since the user doesn't know what is real or fake, their behaviour is natural).

Experimental Prototypes (A-B testing). In digital products, this can be done by coding two versions and randomly splitting users. For small projects, you might simulate an A-B test by alternating conditions. For example, on your personal blog, publish with two different layouts on different weeks and see engagement metrics. Or if trying to improve meeting productivity in your team, run two meetings with different formats and gather feedback or measure outcomes. The rigorous part is treating it like an experiment: change one key variable at a time and compare. A-B testing provides strong internal validity if randomised, and with sufficient numbers, can give statistically significant results. Ensure you have defined what metric indicates better beforehand (to avoid cherry-picking after the fact).

Computer Simulations or Models. Sometimes, especially in engineering design, one might create a simulation rather than a physical prototype to test an idea. In personal efficiency, a simulation might be harder, but one could use tools like calendars or spreadsheets to model scenarios (e.g., simulate different budget plans or schedules). This is more analysis than design, but it can inform prototypes to focus on viable options. For validity, remember that simulations rely on assumptions. Ensure those assumptions are based on reality (external validity) or plan to verify them later.

Test Techniques

Usability Testing. Watch users (or yourself) use the prototype to perform tasks, without intervention except where truly stuck. Collect both observational data (where do they stumble? how long do tasks take?) and subjective feedback (what did they feel? what did they like/dislike?). Jakob Nielsen’s heuristic of five users can catch the majority of usability issues, especially when users are relatively homogeneous. If you find the same confusion or error with multiple people, it is a strong sign to iterate on that aspect. To apply to yourself: you might video-record your own usage of a new tool to catch moments of friction you didn’t notice in the moment, or keep a diary of difficulties each day using a new method.

Field Trials/Pilots. Give the prototype to users to use in their own environment for a period (a beta test). This tests real-world uptake and uncovers issues of context (high ecological validity). For example, if you design a new recipe workflow, have a friend actually try it for a week in their kitchen and report back. Or release a beta version of an app to a small group. The data from pilots can be both quantitative (usage logs, performance metrics) and qualitative (interviews, surveys post-trial). Transferability is the qualitative analogue of external validity. By running a pilot in a different context or with different users, you assess whether the solution transfers well. If not, note what conditions it seems to require.

Diary Studies. Particularly useful when the experience unfolds over time or you want users’ evolving impressions. Participants agree to record their thoughts/behaviours at certain intervals or when triggers happen. For example, after each day of using a productivity app, the user writes a short entry about how it went. This yields longitudinal data that single-session tests cannot. It can also highlight variability: did the solution consistently help, or only on certain days? For personal use, maintaining a diary during your experiment is effectively the same, and it improves the dependability of your self-research (you are not relying on memory alone).

Expert Review. Sometimes, instead of (or in addition to) user testing, you might consult experts to evaluate the design (e.g., a UX expert doing a heuristic evaluation, or a mentor giving feedback on your new strategy). While not a direct experiment, this can catch issues based on established principles. It is like a theoretical validation. For instance, an expert might point out that your prototype violates a known usability heuristic (predicting users will be confused), which you can then test specifically. This can save time by refining prototypes before broad testing. However, be mindful that expert opinion is not always proof. It should complement, not replace, empirical testing with real users for external validity.

Data Analysis of Test Results. Once tests are done, employing statistical analysis (if sample size permits) can formalise the validation. Even simple stats like average task times before vs after, or basic thematic coding of feedback, help structure the findings. If you have enough data, you might do a t-test or chi-square to see if differences are significant (many accessible tools exist, some without coding). For personal experiments, one might graph the data (e.g., a line chart of productivity scores over days) to visually inspect trends and variance. If an improvement is within the normal day-to-day fluctuation range, it might not be a real improvement (no signal beyond noise).

All these techniques should be chosen and tailored with an eye on what hypothesis you are testing at the moment. In the early stages, more exploratory, open-ended methods are used (observations, open interviews) to generate hypotheses. In later stages, more structured, confirmatory methods (A-B tests, timed tasks, questionnaires) are used to validate or invalidate specific hypotheses. Design Thinking is inherently iterative and non-linear, so you might go back and forth. For example, test results might send you back to do more user research or to prototype a different idea. That is expected and one of the key ingredients in Design Thinking. You iterate back to revisit previous stages all the time, and a stage never closes behind you. Consequently, the Test stage often feeds back into the other stages: findings can redefine the problem, inspire new ideas, or suggest tweaks to prototypes. This iterative looping is akin to how in science an experiment might lead to new questions and further experiments, refining the theory each time.

Creativity for Personal Innovation

Bringing scientific methods into Design Thinking is not about making the process rigid or stifling creativity. Rather, it is about injecting clarity and learning into each step. By defining clear hypotheses, you focus your creative energy on ideas that matter. By considering validity, you ensure the insights you gain are real and not illusions. This symbiosis allows individuals to innovate efficiently: you still brainstorm and imagine freely (divergent thinking), but when it is time to converge (Define, Prototype, Test), you do so with disciplined methods that maximise your chances of finding truly effective solutions.

For an educated layperson or someone in the self-improvement realm, the key takeaway is that you can treat your personal growth or projects as a series of experiments. You become both the designer *and* the researcher of your life. If something doesn't work, you haven't failed, you've learned, as long as you set it up in a way to know *why* it didn't work. Adopting this mindset can also reduce the emotional sting of things not working out; it is not a judgment on you, it is just a hypothesis that didn't pan out. Thomas Edison exemplified this experimental perseverance in inventing the light bulb, reportedly testing hundreds of materials until finding one that worked, famously saying, "I have not failed, I've just found 10,000 ways that won't work." He was effectively applying hypothesis testing (each filament material was a hypothesis) and he treated each test as a learning iteration, not a personal failure.

On the other hand, scientific rigor reminds us to be sceptical of our own ideas (in a healthy way). It encourages us to seek falsification, to test in ways that could prove us wrong, not only those that confirm our hopes. As discussed by philosopher Karl Popper, a hypothesis must be falsifiable to be meaningful; similarly, in design, if you only ever seek confirmation (Hey, users, isn't this feature great?), you will miss the truth. So we ask neutral or even devil's-advocate-style questions: What might be the reasons this solution *wouldn't* work? Let us test for those. Remember that you never test to prove your point, but always to get feedback, both positive and negative. Paradoxically, by looking for potential failure points, you make the solution stronger and more creative, because you will uncover flaws and can address them with novel ideas.

It is worth noting that not every design effort needs laboratory-grade rigor. Depending on the stakes, you can dial the depth of the method up or down. For a low-risk personal experiment (trying a new recipe), you might just do one trial and eyeball the result. But for a major life decision or a product launch, you'd be wise to increase the rigour. Maybe run multiple experiments, gather lots of data, get peer review on your thinking (friends as sounding board), etc. The principles remain the same, only the extent varies. Even a personal project can achieve *trustworthiness* by being credible (aligned with evidence), transferable (lessons apply elsewhere), dependable (the process is documented and logical), and confirmable (conclusions are grounded in data, not just personal whim). Those qualities make your personal design outcomes more robust.

In conclusion, applying scientific and research methods to Design Thinking's convergent phases enables a powerful combination of inspiration and evidence. You define problems not just by intuition but by insight, prototype solutions as purposeful experiments, and test them in ways that yield reliable knowledge. By identifying what exactly is being tested at each step, be it an idea's desirability, a feature's usability, or a system's effectiveness, you can focus on the relevant validity concerns and tailor your approach accordingly. This leads to methodological best practices naturally slotting into your workflow: you will randomise without thinking much about it, phrase survey questions carefully, or choose test participants thoughtfully, simply because you know *why* it matters. The result is personal and creative innovation efforts that are more likely to succeed and, importantly, more likely to teach you something valuable even when they don't. Just as Design Thinking brings a human element to problem-solving, the scientific method brings a truth-seeking element. Together, they equip you to tackle challenges in a way that is both imaginative and reliable. This is a true best of both worlds for the curious and creative mind striving for efficiency and breakthrough.

Before we move on, a brief reminder. The scientific methods and skills discussed in this chapter, and their relevance to the everyday practice of a personal innovator, should not be confused with the cultivation ideal, often promoted by proponents from the humanities. That ideal holds that broad and deep knowledge of the classical subjects, such as literature, philosophy, and the arts, is the most important kind of knowledge. From a CST perspective, such claims lack validity. They overlook the dynamic, operational nature of knowledge that characterises effective personal innovation. Being trained in the cultivation tradition certainly does no harm. But adherence to a centuries-old model of intellectual refinement does not, in itself, strengthen or enhance the kind of common-sense thinking that matters today.

6. Consulting Methods

Management consultants are renowned for their problem-solving prowess, structured creativity, and efficiency, skills that should be of much interest to a personal innovator as well. This chapter delineates the tools, techniques, cognitive models, and habits these consultants use to reason through problems, analyse data, structure solutions, and innovate, forming a playbook for the personal innovator's creative work. We will explore core principles such as MECE structuring, issue and hypothesis trees, 80/20 prioritisation, hypothesis-driven problem solving, SCQA (Situation-Complication-Question-Answer) logic, and the Pyramid Principle, among others. Additionally, we highlight day-to-day working practices and mental habits: rigorous time management, efficient note-taking, knowledge reuse, and fostering a culture of critical thinking (including the obligation to dissent). Each section provides real-world examples, demonstrating how management consultants cultivate clarity, insight, and leverage in their workflows. The author worked for more than 20 years as an IT and management consultant before changing sectors to academia, and knows how useful such skills are. This chapter is a dive into how consultants such as I think and work internally to solve problems creatively and systematically.

Consulting firms build their success not just on business savvy, but on well-honed problem-solving methodologies and a mindset of structured creativity. Barbara Minto, McKinsey's first female MBA hire and author of *The Pyramid Principle,* said that the pyramid is a tool to help you find out what you think, emphasising how structure can unlock clarity. In the late 1960s, she pioneered the concept of MECE (Mutually Exclusive, Collectively Exhaustive) grouping, a way to break down information and ideas into distinct, non-overlapping categories that together cover all possibilities. MECE thinking underpins how consultants structure problems and solutions to ensure no important aspect is overlooked while avoiding confusion from redundancy. Beyond structuring information, management consultants leverage cognitive frameworks to approach problems systematically. They define problems clearly, disaggregate complex issues into manageable parts, formulate hypotheses, and then gather evidence to prove or disprove these hypotheses. This hypothesis-driven problem solving (sometimes called *answer-first*) contrasts with unguided analysis; it focuses efforts from the start on what is likely to matter most. It is an iterative approach: as data come in, hypotheses are revised or rejected, and new hypotheses may arise.

Importantly, consultants must balance data and intuition. Such a process is heavily fact-based, yet experienced consultants acknowledge the reality that decisions often rely on an interplay of analysis and experienced judgment. Consultants are trained to trust data but also to recognise when to apply the 80/20 rule (Pareto principle) to find quick insights, when to avoid "boiling the ocean" with analysis, and when to step back and ask *So what?* about the data. This blend of analytical rigour and pragmatic insight defines the consultant's efficiency.

Furthermore, top consultants develop habits around time management, communication, and knowledge management that set them apart as *personal innovators*. They manage long workweeks through prioritisation and personal rules for work-life boundaries. They take structured notes and maintain fact packs for ready reference. They actively reuse knowledge by tapping into databases or past project documents, living by the mantra *Don't reinvent the wheel*. They also practice clear and hypothesis-led communication, using storylines and frameworks like SCQA (Situation, Complication, Question, Answer) to present ideas in a logical flow. Even in everyday team discussions, there is an emphasis on structured thinking: internal communications often mirror the clarity expected in client presentations.

So the internal workflow of an elite consultant is a master class in structured problem-solving, efficient analysis, and clear communication. In the sections that follow, we peel apart these elements, starting with the foundational MECE principle, and illustrating each with real-world practices and examples. Whether working solo on a memo or collaborating in a team room covered in whiteboard sketches, consultants anchor their creativity in proven techniques that anyone can learn from and apply to become a more creative and efficient personal innovator.

MECE Thinking

Mutually Exclusive, Collectively Exhaustive (MECE) is often the first principle taught to new consultants at McKinsey and has since permeated the approach of many other consultants as well. Coined by Minto during her tenure at McKinsey in the 1960s, MECE is a way of structuring information so that no important element is missing (collectively exhaustive) while avoiding overlap between elements (mutually exclusive). This logical structuring is the backbone of issue trees, storylines, and even slide presentations.

Purpose of MECE: By organising issues or ideas in a MECE way, consultants achieve maximum clarity. Each item in a list or each branch of an issue tree addresses a distinct aspect of the problem (avoiding confusion) and taken together, the items cover the entire scope of the question (avoiding gaps). Minto's insight was that grouping ideas this way forces the problem-solver to *pull out of their head information that they weren't aware was there* until the thinking is crystal clear. Imagine you are tackling the problem: How can we increase the profitability of Company X? A non-MECE list of issues might randomly mix cost problems, revenue ideas, market conditions, etc., with potential overlaps or missing pieces. A MECE approach, by contrast, starts with distinct buckets, for example Revenue-related factors, Cost-related factors, Market/Industry factors, each of which can then be broken down further. If *Revenue* and *Cost* cover all internal financial levers, and *Market factors* cover external influences, you have a collectively exhaustive set at a high level. Within each, you ensure sub-issues don't overlap (e.g., increased sales volume vs. increased price are separate revenue drivers). Consultants often test their issue lists: *Is each item separate and distinct? Together, do they cover everything?*

Example. Start with the problem: We need to sell more X. A MECE issue list could be: (1) *Improve sales to retail outlets*, (2) *Improve marketing to consumers*, (3) *Reduce unit cost to enable price cuts*. These three don't overlap and plausibly cover all areas (sales process, demand generation, cost competitiveness). If one were to add Reengineer X production process, this would break MECE because production process changes are actually a subset of reducing unit cost (they overlap with item 3). Instead of adding a redundant fourth category, that idea belongs as a sub-issue under costs. This example shows how MECE forces careful thinking about where each idea belongs.

MECE Beyond Lists. Minto's Pyramid Principle, an approach to writing and presenting, is deeply rooted in MECE. Each grouping of ideas in a pyramid must be MECE for the logic to hold. For instance, when writing a report, a consultant might have three key findings supporting the main recommendation. Those findings should be MECE (covering different domains of the problem without redundancy). If any finding overlaps with another, it indicates a muddled structure, which can confuse the audience. It is common practice for consultants to iterate on the grouping of ideas during analysis and again while preparing slides to ensure a MECE flow.

Universality. While MECE emerged at McKinsey, the concept is now part of the general lingua franca of consulting. MECE thinking is universally applied at many other top firms. The goal is always the same: structure problems and solutions so nothing important is missed and nothing is counted twice. In practice, consultants use MECE not dogmatically, but as a useful guideline. There are situations where perfect mutual exclusivity is hard to achieve or not worth agonising over, but starting with a MECE mindset helps break large, fuzzy problems into clear components. MECE's real power is how it encourages complete yet concise thinking. By applying MECE discipline, personal innovators can mirror consultants' ability to organise information, whether planning a research report, writing an email, or structuring a personal project. The result is thinking that is both comprehensive and clear, a foundation for creative problem solving since it gives you a full map of the issue terrain on which to innovate.

Issue Trees

One sign of trained consultants is their use of issue trees and hypothesis-driven problem-solving to break down complex problems and direct their analysis efficiently. This approach is often summarised as moving from *problems* to *hypotheses* to *analysis*, rather than just diving into analysis without a focus. An issue tree (or logic tree) is a hierarchical diagram of questions or issues branching out from a central question. It visually disaggregates the problem into MECE sub-problems. The idea is to create issue trees early in a project. Issue trees ensure a structured, thorough exploration of the problem space. By laying out all possible drivers of a problem, consultants can discuss and decide where to focus. It is also a tool to communicate the problem structure to team members or clients, showing here's what we think influences the outcome. There are generally two kinds of trees. Diagnostic (Information) Trees are used when you need to understand what is going on. For example, Why is our market share declining? might branch into issues like *competitor actions*, *customer preferences*, *sales execution*, etc. Decision (Solution) Trees are used to answer What should we do? For example, How to increase market share? branches into *options to increase share*, such as *pricing strategies*, *marketing investment*, *product improvements*, etc., each of which can be further branched.

Example. If profitability is down, the first branch of a profitability issue tree might be *Revenue* vs. *Cost* (MECE drivers of profit). Revenue might branch into *price* and *volume*; Cost might branch into *fixed* vs. *variable*, and so on. The tree helps the team enumerate all the levers affecting profit in a structured way, so they can target the analysis. The method emphasises using frameworks (often existing, generic issue trees) to avoid reinventing the wheel; a McKinsey team will download past similar studies to see how issues were framed.

Branches of an issue tree must be MECE. This is where MECE thinking becomes practical: if branches overlap, you might double-count or misallocate work; if they have gaps, you might miss a critical analysis. Consultants check their issue trees by asking: If we answer all these sub-questions, do we answer the main question? And does each sub-question answer a unique part of it? Use issue trees at the start of a project or problem, especially if data is sparse or you need to organise your approach. This ties back to Design Thinking, but perhaps not the way you expected. In the ideal DT situation, there is much data to collect. Many users stand in line to be empathised with, interviewed, and immersed with in various ways. However, the cold reality is not always that sweet. Data might be scarce or users might not be available for one reason or the other, but the innovation and problem-solving process cannot stop. In such cases, management consulting methods often represent not only the most efficient way forward but sometimes also the only one. BCG, for example, teaches new consultants a sequence: after defining the problem, Step 2: Structure the problem using issue trees. This can be done individually or in a team brainstorming session. Bain similarly encourages breaking down cases into issue trees in training, though Bain also places a heavy focus on starting with a hypothesis (their *answer-first* approach, which we address next).

Hypothesis-Driven Problem Solving

While issue trees map out what to consider, hypothesis-driven problem-solving is about where to start testing. Instead of analysing everything, consultants form an initial hypothesis (an educated guess of the solution) and focus their research and analysis to validate or refute it. This approach is deeply ingrained in McKinsey and BCG cultures, and Bain explicitly calls it *Answer-First* thinking. Complex business problems often have a breadth of data and factors; trying to gather and analyse all of it would be inefficient or "boiling the ocean". A hypothesis provides a focusing lens. It is based on prior experience, industry knowledge, or early data indicators. By hypothesising a likely answer, consultants can direct their limited time toward proving or disproving the most important assertions. As an iterative process, if the first hypothesis is wrong, the data will show it, and one can pivot to a new hypothesis.

A good hypothesis is *specific, testable, and relevant*. Strong hypotheses often invite debate, are not obvious, matter to the outcome, and are provable with data. For example, instead of hypothesising vaguely that Company X should improve sales, one might hypothesise that Company X can increase sales by 15% by optimising its online channel marketing, which currently underperforms the industry by at least 20%. This hypothesis can be tested by analysing channel data, industry benchmarks, and running a pilot or modelling the impact of improvements. Sometimes, consultants create a hypothesis tree instead of an issue tree. This is similar in structure but phrased as hypotheses at each branch rather than questions. Each of those can branch further into sub-hypotheses or specific analyses. The hypothesis tree essentially lays out the argument you will try to prove with data. The hypothesis-led approach is not linear. Consultants iterate between hypothesis and data:

Based on experience and any initial data, propose a possible answer. What data would support or refute this hypothesis? Design analyses accordingly. Quickly gather data (by market research, financial analysis, expert interviews, etc.) and check your hypothesis. If evidence supports it, strengthen it; if evidence contradicts, discard or modify the hypothesis; then repeat. This is a mini-loop in the Design Thinking methodology. Coming from radically different cultures, design thinkers and management consultants have discovered similar ways of working, albeit with differing terminologies. These similarities function as useful triangulations in finding the best practices for the personal innovator.

The *test to destruction* mindset is often cited at McKinsey. Teams are encouraged to beat up their hypotheses with data to ensure only solid arguments survive. This scientific approach, akin to forming a theory and attempting to falsify it, allows consultants to learn and fail fast. This means once a hypothesis is clearly supported or refuted, they move on, embodying efficiency. Bain & Company explicitly uses the term *Answer First*. The idea is that even when you don't have all the information, articulate a tentative answer as a starting point, then drive your analysis to validate it. Bain often pushes teams to state their current answer early in an engagement (e.g., We believe the client should do X because of Y) and treat everything else as working to substantiate that story. This doesn't mean Bain ignores data. Rather, they're using the hypothesis to focus data collection on what matters. Again, this ties in with Design Thinking. Formulate a hypothesis (or Point of View) and then prototype and test it to learn whether it was a good one or not. Also note that Bain uses the story as a format, something encouraged a lot in DT. BCG also embraces hypothesis-driven work. BCG case teams are known for systematically pushing one level deeper. BCG's materials often focus on the structured process but implicitly include hypothesis testing at each stage. The streamlined hypothesis-led approach is sometimes called the BCG and McKinsey go-to method in the business.

A danger of hypothesis-first is *confirmation bias*, only seeing data that fits your guess. Consultants are warned about this; hence, the emphasis on genuinely testing to *disprove* the hypothesis too. If not carefully managed, one might ignore inconvenient data. The firm cultures mitigate this with practices like team debates, an obligation to dissent (at McKinsey) to challenge assumptions, and insisting on evidence for every storyline assertion (show me the backup). In fact, a culture of healthy debate around hypotheses is vital, which we'll touch on in the critical thinking section.

Example. Imagine a McKinsey team working for a bank wanting to cut costs. They might start with an issue tree of bank cost components (branch operations, IT, corporate overhead, etc.). Using prior knowledge, they form a hypothesis: Branch operations can be optimised to reduce total costs by 15% via staff scheduling improvements and automation. They design analyses: benchmark branch staffing vs. transaction volumes, pilot an automated system in a test region, etc. Meanwhile, other hypotheses (IT or overhead cuts) might also be in play, perhaps tackled by different sub-teams. As data comes in, the branch hypothesis might prove that even bigger savings are possible, or perhaps it saturates at 5%, and IT shows more promise, shifting focus. Throughout, the issue tree ensures all areas are at least checked, and the hypothesis focus ensures the analysis is efficient, not aimless.

Consultants choose their approach based on how familiar the problem is. If they or the client truly don’t know where the answer might lie (new problem type, new industry), they might start with a broad issue tree to map it out, then prioritise. If they have a strong intuition (perhaps this client's problem is similar to another case), they might start with one or a few hypotheses and go straight into targeted analysis. Oftentimes, they’ll use both: build an issue tree, pick the branches likely to have the biggest impact and then hypothesise within those branches. The interplay is dynamic. So issue trees give consultants the what-could-we-look-at map, and hypotheses give the what-should-we-look-at first focus. Combined, they symbolise structured yet creative problem solving: structured in laying out the terrain, creative in jumping to an answer and seeing if it holds. This approach can save tremendous time and often yields early insights that guide the team (or an individual’s work) long before every data point is known. For a personal innovator, adopting a hypothesis-driven mindset means that before diving into research or analysis, stop to brainstorm a likely answer and a plan to test it. It is a way to bring scientific rigour to business problems and ensure efficient use of time.

The Pareto Principle

One of the rules to live by in consulting is the 80/20 rule. Often invoked in the context of time management and analysis, the 80/20 rule (also known as the Pareto Principle) states that *80% of results come from 20% of the effort or causes*. In a consulting context, this principle is a guiding light for prioritisation: focus on the critical few, not the trivial many. This is mirrored in Design Thinking but for a somewhat different reason. There, the trivial many are also disregarded, but for the reason of finding the interesting outliers that drive innovation, rather than pointing to how to solve or rectify the problem in the consulting case.

The 80/20 idea wasn't invented by consultants. It is credited to economist Vilfredo Pareto and observed in many contexts (for example, 20% of customers generating 80% of sales). Consultants adopted it as a mantra for efficiency. At McKinsey, it is commonly said that 20% of the analysis will get you 80% of the answer. The key is to identify *which* 20% of issues or data deserve your attention. That is found by probing the data at hand (or collecting more). Don't analyse every possible set of information. Start with sets that likely yield insight (by region, by product, by customer segment, whichever is likely to have variance). Be comfortable with rough estimates if they're directionally sufficient. A back-of-envelope calculation might eliminate whole branches of an issue tree as unimportant, saving weeks of work.

80/20 is not about being sloppy or ignoring analysis. It is about prioritisation and recognising diminishing returns. Consultants are trained to ask: Is this deep dive going to change the recommendation meaningfully? Or have we hit the point of marginal returns? They use 80/20 to justify stopping an analysis when further detail won't add value commensurate with the time required. For example, after building an issue tree or list of hypotheses, consultants apply the 80/20 lens to decide where to focus first. They consider *impact* and *effort* (or feasibility). A common tool is the familiar Wow-Now-How diagram from Design Thinking, a 2x2 matrix plotting issues on high/low impact vs. easy/hard to implement or analyse, that is equally useful in management consulting. The top-right quadrant (high impact, easy implementation) is tackled immediately, while the bottom-left (low impact, hard implementation) might be dropped. This is a structured way of doing 80/20 prioritisation. Bain consultants often mention 80/20 as part of their toolbox for pragmatic consulting. Bain's culture historically emphasises practical, quick-and-dirty analysis if it yields a usable answer faster. BCG, too, values insight over academic thoroughness. The idea is to first get to a roughly right answer that captures the big drivers (80/20), then refine it if needed, rather than aim for perfect precision from the get-go. In recruiting, candidates who spend too long on minor details are dinged. Interviewers look for 80/20 thinking: Did the candidate focus on the biggest issue? Did they structure their time, diving deep where it matters and summarising or skipping less important areas? This mirrors on-the-job expectations.

While 80/20 is powerful, it can be misapplied. Not every situation is a neat Pareto distribution. Sometimes, what appears to be the 20% might rely on something in the 80% you would ignore. Also, if used as an excuse, one could prematurely drop analysis that is actually important (the remaining 20% can matter if, say, a safety issue or compliance risk lies there). Consultants guard against this by periodic comprehensive checks. For example, once they have a recommendation, someone might ask, Are we sure we are not missing a piece of the puzzle outside our 80/20 focus? Moreover, truly collectively exhaustive thinking (from MECE) is needed to verify that what was set aside isn’t critical.

For any personal innovator drowning in information or tasks, asking *What 20% of this will contribute to 80% of my desired outcome?* is a powerful focusing question. Consultants use it to allocate their time: e.g., spend the most time on the analysis that yields the big recommendation, not on perfecting a chart’s formatting. They use it to allocate team resources: e.g., put your best analyst on the most complex, high-value analysis, not on something secondary. And they use it to allocate client effort: focusing client interviews on the most insightful experts, not every employee, etc. The 80/20 principle in consulting is about maximising impact per unit of effort. It complements hypothesis-driven thinking (which presumes your first hypothesis is about the high-impact drivers) and MECE (which helps ensure that focusing on 20% doesn’t omit something essential). It is a mindset of working smarter, not just harder, which is important in an environment where hours are long but results, not toil, are what count.

Don’t Boil the Ocean

Hand in hand with 80/20 prioritisation goes the consulting adage: Don’t boil the ocean. This phrase encapsulates the importance of not trying to do too much at once or not over-analysing to the point of wasted effort. It is about being selective with analysis and data in order to solve the problem effectively and efficiently. Just as boiling the entire ocean to get a bit of salt is an exercise in futility, trying to analyse all information in hopes of solving a problem is impractical. Consultants use this phrase to caution against overly broad scopes or analysis paralysis. They recognise diminishing returns; additional data analysis could consume more time and energy than it yields. McKinsey’s approach (and by extension BCG and Bain) is to gather enough facts, not *all* facts. *Enough* means sufficient to prove or disprove the hypothesis or to support an analytical conclusion. This reflects a disciplined approach: have a clear question for each analysis and stop when that question is answered. Consultants often set small analysis work streams with a specific scope (e.g., What is the revenue by product and region for the last five years, and what is driving growth?). Once answered, they move on, rather than endlessly interpreting the data in new ways without a purpose. Instead of boiling the ocean, consultants focus on key drivers of the problem. In any business or system, usually a few factors drive the majority of outcomes. Identifying these early is important. For example, in declining widget sales, the key drivers might be *retail distribution reach, product pricing, and competitor innovation*. If you agree that those are key, you might (initially) put less attention on, say, macroeconomic trends or fringe customer segments.

This relates to an engineering concept: the Square Law of Computation. The complexity of solving many algorithmic problems increases quadratically with the number of components, so you simplify heavily by focusing on the main components. By analogy, consultants narrow the problem scope to avoid an explosion of their analysis. For example: many factors (weather, consumer confidence, raw material prices) might affect sales of X, but if the data or judgment says the top three factors are, say, *distribution, price, and product features*, then the team will ignore the rest to make the problem tractable. Ignoring the rest doesn’t mean literally never looking at them. It means in initial focus. Often, each rest factor is parked as a low-priority branch of the issue tree. If something in the analysis later contradicts expectations, consultants can revisit it. For instance, if distribution, price, and features don’t explain the sales decline fully, maybe the team revisits macro trends to see if an economic downturn is contributing more than thought. The point is, they don’t start by analysing *everything* (boiling the ocean); they start with the likely key drivers.

Example. Consider a BCG project on improving a factory's efficiency. Boiling the ocean would be trying to analyse every machine, every process, all at once. Instead, the team identifies a few bottlenecks or key cost drivers (say, two production lines that account for 60% of downtime, and one raw material that's 40% of the cost). They focus data collection and observation on those. They might still have a broad map of the factory process (issue tree of sorts for operations), but they decide, e.g., maintenance practices and scheduling in Lines A and B are where to deep dive first. If those yield big insights (e.g., poor preventive maintenance causing most breakdowns), they might not need to look as hard elsewhere. If they didn't find much, they had to move to the next driver on the list.

Management consultants are trained to define the scope of work sharply (recall the earlier section on defining the problem). This means explicitly stating what is out of scope, so you don't accidentally boil that part of the ocean. If a client asks for a cost reduction plan for a division, maybe the scope is only that division, only internal cost levers (e.g., excluding outsourcing for now if that's considered external), and only within a three-year horizon. Writing this down and getting stakeholder agreement gives cover to not analyse, say, expansion options or macroeconomic scenarios beyond three years, etc. Consultants often get this sign-off in a charter or an early workshop. Another method is self-imposed (or manager-imposed) time limits on analyses. For example, let us spend one day pulling data on X and see what we get. This encourages finding quick indicators rather than delaying until perfect data is acquired. Often, a junior team member could chase data for weeks if unchecked; a savvy engagement manager will time-box it, forcing an 80/20 data grab and then deciding if more is needed. The motto is Do enough analysis to be confident, but not so much that you are polishing instead of problem-solving. So don't boil the ocean is about focus and efficiency in analysis. It complements the 80/20 rule: while 80/20 says to focus on the most impactful 20%, the concept says to consciously decide which analyses not to pursue. It is a negation principle: avoid analysis for its own sake or open-ended research with no clear question in mind. Consultants internalise this, learning when to say We have enough to move forward or This angle isn't worth exploring. For personal innovators outside consulting, the lesson is similar: define your scope, don't let curiosity or perfectionism lead you to research everything, and identify the key factors of your problem so you channel your energy where it counts. The ocean of information in the digital age is vast; the consultant's mindset is to boil just a pot of it to get a useful amount of salt, and no more.

The Pyramid Principle

Solving a problem is only half the battle for consultants; communicating the solution effectively is the other half. Top-tier consultants are trained to present ideas in a structured, persuasive manner. Two closely related frameworks play a central role: The Pyramid Principle (structured thinking and writing) and SCQA (Situation, Complication, Question, Answer) for introducing a story. These techniques, originally developed by Barbara Minto at McKinsey, are now standard tools at McKinsey, BCG, and Bain for internal and client communication.

The Pyramid Principle is a method of organising information like a pyramid: start with the main point (the apex) and support it with layers of arguments and data underneath, grouped logically. The key rules:

- Start with the answer (governing thought), then
- Group and summarise supporting points (each group's summary is a support for the main point).
- Ensure ideas in each grouping are logically the same and MECE.
- Order points for logical flow (either deductive or inductive sequences).

In practice, when a consultant writes a slide deck or a memo:

- The top line (title or initial statement) is the main recommendation or conclusion.
- Then come a few bullet points or sections that are the primary supports (the first-level pyramid grouping).
- Each of those might be elaborated with sub-points or data (second-level details).

This way, if an executive reads just the titles of slides or the first sentence of each paragraph, they grasp the story. There are two advantages. i) *Clarity and Logic*: By forcing thoughts into a pyramid, any disconnects or gaps become evident. If an idea doesn't fit into the structure, maybe it is extraneous (should be cut) or the structure needs adjusting. Minto noted that writing in pyramid form forces you to pull out of your head information and shape it until the thinking is crystal clear. ii) *Efficiency for Audience*: Executives are busy. The pyramid style gives them the answer first and allows them to drill down only where needed. In contrast with a narrative hero's journey approach, consultants don't slowly build up to a conclusion like a drama, they give it up front. The supporting points are like: if you believe these, you will believe the main point.

SCQA (Situation, Complication, Question, Answer) is a specific way to start the pyramid or any communication:

- **S**ituation: Establish the context or facts – the status quo or starting point.
- **C**omplication: Describe what has changed or why the status quo is problematic and why the issue or tension needs resolution.
- **Q**uestion: Formulate the question that logically arises from the complication.
- **A**nswer: State your answer to that question.

For example, suppose a consultant presents: *Situation:* Our client grew 5% annually in a stable market. *Complication:* Recently, growth stalled and competitors are outpacing them. *Question:* How can our client reignite growth to maintain market leadership? *Answer:* They should implement a digital sales strategy targeting younger consumers, which we estimate can add 8% to annual growth. SCQA effectively sets the stage and then immediately gives the solution. McKinsey suggests using SCQA to frame the introduction of any report. It is essentially how you introduce the pyramid's main point. Consultants use SCQA at the start of documents, but also in everyday thinking. If a partner asks, Why are we doing this analysis, a consultant might reply in mini-SCQA form: e.g., Well, currently (Situation), we have only a rough idea of customer profitability. The issue is (Complication) we suspect a small segment drives most profit, but we don't know which, raising (Question) which customers to focus sales efforts on. So (Answer) this analysis will identify the top 10% most profitable customers and how to target them. This trains clear thinking. SCQA is also recommended for job interviews, emails, and really any persuasive communication. It is about contextualising your answer. Many firms internally talk about the *storyline* of a presentation, and SCQA is the way to set up that storyline.

Example: McKinsey is known for their document-style PowerPoint presentations that read almost like a report (sometimes called McKinsey's Way of presenting). Every page has a title that advances the argument. The logic flows top-down. If you flip through only the titles, you get the narrative. This approach became a signature that other firms also emulate.

So the Pyramid Principle and SCQA are about communicating ideas in a logical, top-down fashion. They help a consultant turn complex analyses into a coherent story where the audience immediately grasps the key message and can follow the supporting logic easily. These techniques are invaluable for any personal innovator: structuring an email, a report, or even a difficult conversation using SCQA and pyramid logic can vastly improve clarity and persuasiveness. They force the writer/speaker to think through the logic (making one's own reasoning clearer) and make it easy for the recipient to understand and remember the points. In essence, they translate the rigorous problem-solving the consultant did into equally rigorous communication of the solution.

Time Management

The demanding workloads at top management consulting firms force consultants to develop strong time management and personal productivity habits. While each individual finds what works for them, there are common techniques and cultural norms that help consultants stay efficient and avoid burnout. Here we explore how top-tier consultants manage their time, prioritise tasks, and maintain effectiveness, as well as strategies for work-life balance in a tough environment. In the next chapter, we will have a look at a general innovation-supporting way of time management. Here, we look specifically at a few tips on how top management consultants manage their time through a set of techniques.

Prioritisation and Task Management

Daily Prioritisation. Consultants often start the day (or end the prior day) by listing out tasks and prioritising them. Given multiple work streams and client needs, they constantly ask: *What is the most critical thing I need to get done today?* This echoes 80/20 thinking on a micro scale. For example, if a client meeting is tomorrow, preparing for that takes precedence over a less urgent analysis. Tools like the Eisenhower Matrix (urgent vs. important) are implicitly applied.

Big Rocks First. A common time management lesson (not unique to consulting) is to do big, important tasks (big rocks) first, then fill the rest of the time with smaller tasks (gravel, sand). Consultants may not explicitly refer to this analogy daily, but the concept is present. For instance, schedule a block in the morning to work on the primary analysis while energy is high, then handle routine emails or slide formatting later in the day.

Work Planning. On projects, managers create a work plan. This includes major tasks, owners, and timelines. For a personal innovator, having clarity on who's doing what by when helps manage their own time. They know the critical path tasks (things that, if delayed, will delay the whole project) vs. parallel tasks. Good consultants look ahead. If they have a heavy analysis due next week, they might start gathering data now because data often causes delays.

Time Boxing. Because tasks can expand if given too much time (Parkinson's Law), consultants frequently time-box activities: e.g., *I will spend two hours drafting this section, then I must send it for review*. This aligns with the don't-boil-the-ocean mentality. One might block, say, 90 minutes to brainstorm a structure for a document (instead of doing it indefinitely) or half a day to pull benchmarking data. If the time is up and the output is good enough, they move on unless it is truly insufficient, in which case they reassess priorities.

Brainstorming and Creative Problem-Solving

Despite their structured methods, top consultants also excel at *creative thinking* when facing novel challenges. Brainstorming is a primary technique they use to generate ideas, often in a team setting, but also individually. However, unlike free-form brainstorming, top consulting firms often bring structure and rigour even to creative sessions to make them productive. Let us look at how consultants brainstorm and employ other creativity tools, and how they balance out-of-the-box thinking with their analytical frameworks. Brainstorming the McKinsey Way means setting up the work context for optimising the outcome. *Setting*: Usually after initial research and before heavy analysis, the team gets in a room (often a conference room with a whiteboard or flip charts, plenty of sticky notes, etc.). They allocate a substantial chunk of time (often 2+ hours, sometimes on weekends to avoid interruptions).

Preparation: Before the first brainstorm, consultants do their homework. This means brainstorming isn't starting cold; it is informed by data and initial thinking.

During Brainstorming: They emphasise leaving preconceptions at the door. Even if initial hypotheses exist, they are *tested to destruction*. The idea is to be willing to throw out any assumption if the team thinks of a better one. Everyone is encouraged to contribute. Hierarchy is downplayed in these sessions (especially with the obligation-to-dissent culture at McKinsey, a junior can challenge a senior's idea). Similar to DT, no idea is immediately shot down (the classic brainstorming rule: defer judgment). Instead, ideas are captured and then logically debated. If someone says an offbeat idea, others might ask how it could work or what would need to be true, rather than saying that's wrong. Tools like whiteboards, post-it notes, and markers are extensively used to visualise thoughts, group ideas, draw quick charts or issue trees, etc. They might explicitly reframe the problem in different ways to spur thinking. What if we approach this from a customer perspective vs. a product perspective?

Output: They aim to come out with something tangible. A refined set of hypotheses to pursue (perhaps a reordering or new ones). A better-structured issue tree than they started. Key questions to answer or new analysis ideas. For example, We hadn't thought about looking at customer churn by cohort; let us do that. Sometimes, truly innovative solutions emerge that weren't on anyone's initial radar.

Follow-up: Immediately after, teams often assign owners to follow up on each promising idea (who will test which hypothesis, or research an outlandish idea's feasibility, etc.). Brainstorming is integrated into the overall structured approach: it is used to ensure creative coverage of solution space, and then they funnel back into structured analysis.

Techniques to Spur Creativity

Consultants may implicitly use or explicitly introduce various creativity techniques. *Analogies* and Stealing like an artist: They ask Who else has solved a similar problem? If, for instance, a hospital client has an issue, the team might recall something from manufacturing (like lean principles) and try applying an analogous solution. Cross-industry analogies often spark fresh solutions. *Hypothesis* What-ifs: Even if an initial hypothesis seems wrong, they explore What if it were true? How would that solve the issue? Sometimes a seemingly wrong idea leads to a piece of a solution in a modified form. *Zero-based thinking*: Pretend constraints don't exist or start from scratch. If we were a new entrant, how would we design this process? This can break a fixed mindset of improving the current state and instead find transformative changes. *Worst Idea or Anti-solutions*: Generate the worst possible ideas (or how to worsen the problem), called *start reversed* or *undesign* at Openlab. This can be fun and sometimes inversion leads to good ideas by flipping the bad ones.

80/20 in Brainstorming

Interestingly, while brainstorming opens up possibilities, consultants still use MECE and 80/20 afterwards. They might generate 20 ideas, but then cluster and vote on the top three to pursue. There is recognition that not all ideas can be chased (time constraints), so prioritisation kicks in here too: which ideas seem most promising (impact vs. feasibility, maybe a quick gut-check matrix)? This differs from a Design Thinking approach that tries to follow up on all generated ideas before discarding them. The time pressure is not as high in a DT process.

Collaborative Creativity vs. Solo Thinking

Consultants alternate between team brainstorms and individual thinking. After a group brainstorm, each member might go off and work out the details of one idea (solo creative work). Sometimes individuals brainstorm on their own first (jot down ideas), then bring them to the group. This can avoid groupthink and ensure quieter members have ideas to share. Tools like issue trees are themselves creative structures, building them is a creative act in finding how to break down a problem. BCG historically prided itself on being thought partners with clients, which often means brainstorming with clients in workshops to get their perspectives and co-create solutions. These require facilitation techniques to get client managers to open up as well, not just consultants dominating.

Brainstorming Etiquette and Roles

Usually, someone facilitates (often an experienced consultant) to keep it productive: ensuring everyone speaks, moving on if stuck on one idea too long, summarising periodically. A scribe capturing on a whiteboard or electronically is key; others can then focus on ideation. They often revisit the problem definition at the start (We are trying to solve X, let us confirm that's right), because a great brainstorm on the wrong problem is wasted.

Example. A Bain team working on improving customer experience might gather: What are ALL the things we could do to delight customers at key touch points? They had thrown ideas: some conventional (improve call centre, redesign website, etc.) and some wilder (surprise gifts for loyal customers, create a customer community app, etc.). They had listed maybe 30 ideas across the journey. Then they group them (digital vs. physical, cost level, etc.) and narrow them down through a quick discussion of impact and feasibility: perhaps focusing on five ideas to prototype or analyse further, very similar to the first double diamond in D3. Bain, being action-oriented, might directly pilot a couple of ideas to test customer reaction rather than do heavy analysis on all, which is also creative regarding method (test & learn).

Lately, Design Thinking has influenced consulting. Bain even officially admits to using DT as a tool. DT's brainstorming (like using empathy maps, etc.) might be used especially for customer-centric or product innovation projects. BCG and McKinsey have acquired design agencies or built innovation centres (McKinsey's Digital Labs and BCG's Platinion), which incorporate creative brainstorming with designers and developers around. These still incorporate structure but allow more free imagination, especially in the early concept stages. If a team is large or some are remote, they might use brainwriting (a classic DT technique in which everyone writes ideas on post-its and then shares) to have a volume of ideas before discussion.

Thus, while consultants have a stereotype of being rigid or overly analytical, the reality is that they dedicate deliberate time and methods to ensure creative, divergent thinking occurs before converging on solutions. The combination of a free-thinking brainstorming phase with a rigorous analytical phase is akin to designing a solution space widely and then engineering the chosen design precisely. It is in these brainstorming sessions and creative hypothesis generation that sometimes the most innovative client recommendations (new business models, a surprising cost-cutting approach, etc.) are born, which does not come as a surprise for a devoted DT practitioner. And even outside of brainstorming, consultant trains themselves to think flexibly. For example, are we framing this the wrong way? Let us flip it around. Such habits are creative and often encouraged as validation tools, not only creation tools.

For any personal innovator, adopting a consultant's approach to creativity means setting aside time for idea generation, doing your homework to inform creativity, using structure to explore all directions (not just random ideation), and then critically selecting ideas to pursue further. It is about disciplined creativity, an apparent oxymoron that consultants manage to make a reality.

The HEA Cycle

One way to synthesise many of the problem-solving practices of consultants is to look at the iterative cycle of Hypothesis → Evidence → Action (or Recommendation) and back that they engage in throughout a project. This is an indirectly used model in firms, and it is a useful lens to describe how consultants continuously refine their thinking and drive towards solutions.

1. Start with Hypothesis (Hypothesis-Driven Approach): As discussed, consultants begin by hypothesising potential answers or paths. For example, hypothesis: Our client's profitability can be improved primarily by streamlining their product portfolio.

2. Collect Evidence: Next, they seek data or qualitative evidence to test this. This might involve:

   - Gathering quantitative data (financials, market stats).
   - Conducting interviews for expert opinion.
   - Running analysis (e.g., profitability by product).

Importantly, they design specific analyses to test the hypothesis. This is where issue trees help break down what evidence is needed under each hypothesis.

3. Analyse and Check So-What: After evidence collection, consultants analyse it and constantly ask the So-What? question:

    - *What insight does this evidence give?*
    - *Does it support or refute the hypothesis?*
    - *If it supports, so what? What does it mean for the recommendation? If it refutes, so what? Do we pivot our hypothesis?*

    The so-what analysis forces translating data into implications. BCG folks often emphasise that data itself isn't an answer; the team must extract the meaning (the insight that matters to the client). For instance, data might show that 10% of the products have negative margins (evidence). The so-what could be: these products are dragging overall profitability down. Hence, the hypothesis of trimming the product line is validated.

4. Refine/Iterate Hypothesis: Based on evidence, the hypothesis may be refined: maybe it is not just streamlining products but also raising prices on certain remaining products. Or if evidence contradicts, they form a new hypothesis. This cycle repeats, which is what *iterative problem-solving* means. Hypothesis-driven doesn't mean one hypothesis and done; it means you always have a working hypothesis that evolves.

5. Develop Actionable Recommendations: Once confident in a refined hypothesis, consultants translate it into concrete actions (the recommendation):

    - This means figuring out the *how*. If the solution is to streamline a product portfolio, action steps could be: identify which products to cut, how to cut them with minimal customer upset, etc.

    - It also means verifying *feasibility*, often through additional evidence gathering like pilot tests or discussions of organisational capability. The hypothesis might tell you what to do, but you need to confirm that you can.

    This leads to the final recommendation(s) to the client, backed by evidence and analysis that was gathered.

6. Implementation Considerations: Though classic strategy consulting often stops at recommendations, in practice, consultants are increasingly involved in outlining implementation. They may hypothesise potential implementation challenges (Hypothesis: sales team will resist eliminating products because of relationships with certain customers) and gather evidence on that (maybe interviews with sales managers) to prepare mitigations.

Continuous Synthesis and Feedback. At every cycle, consultants synthesise what they have learned and communicate within the team (and often with the client in updates). This synthesis uses the pyramid principle to articulate the interim answer at that point. McKinsey encourages making an initial storyboard or “week 1 draft” of the final report early and updating it as you go. That is essentially forcing continuous synthesis: by trying to write the final story from day 1 (knowing it is tentative), you see what pieces of evidence you need and you refine it as evidence comes. This avoids the trap of finishing the analysis and then scrambling to figure out what it means; instead, you are always saying Right now, our answer is shaping up to X because Y, but we still need to confirm Z.

Agile Thinking. This cycle is akin to an *agile* approach in software: iterate on hypotheses (features), test (develop) them, get feedback (data) and refine. Consultants, especially in digital and operational projects, explicitly use agile project management now, but even in strategy, their iterative hypothesise-and-test approach is agile in spirit. Tools for evidence gathering:

- They use both primary research (interviews, surveys) and secondary research (reports and databases like Capital IQ).
- In analysis, they often do quick models to see if an idea holds water.
- Benchmarking as evidence: how do peers do it? If the hypothesis is that they spend too much on SG&A, benchmarking against peers’ SG&A is evidence.

Hypothesis Logs: Sometimes teams maintain a hypothesis tracker, listing each hypothesis, its current status (proven, disproven, in progress of testing), and evidence references. This ensures no hypothesis gets lost and everyone knows where things stand. It is also helpful in final documents: each key point in a recommendation ties to one or more pieces of evidence, which came from specific analyses or research.

Critical Thinking

Beyond formal processes and tools, top consultants cultivate certain cognitive habits and cultural norms that underpin their problem-solving approach. Two interrelated aspects stand out: a rigorously critical mindset (never taking information at face value, always probing for logic and evidence) and a culture that encourages constructive dissent to surface the best ideas and challenge assumptions. These translate into daily habits that make individual consultants more creative, careful, and credible in their work.

Critical Thinking as Default

Consultants are trained to be sceptics in the best sense. *Fact-Checking Instinct*: A consultant hears a claim (from a client or data source) and automatically considers, What is the evidence? Does other data or experience confirm this? This is partly why they rely on facts so heavily; they won't assume something is true just because the client believes it (politely, they'll validate). Eric Rasiel notes that hiding from facts is a prescription for failure… You must not fear the facts. Hunt for them, use them. This instils a habit: always seek the factual basis. *Devil's Advocacy*: In team discussions, someone will often play devil's advocate: What if the opposite is true? or What could go wrong with this reasoning? This isn't negativity; it is testing robustness. For example, if the team decides on a hypothesis, someone might say, Let us assume this hypothesis is false; what would the world look like? Do we see signs of that world? If yes, then maybe the hypothesis is false. *Structured Thinking*: Consultants habitually structure even their thoughts. This can come across oddly in normal life (the reasons are threefold…). But it is how they parse complexity. If faced with a messy situation, an experienced consultant will quickly try to categorise issues, identify patterns, and break it into components, essentially applying MECE in their head to ensure no blind spots. This becomes a cognitive habit that speeds up problem comprehension. *Questioning and Clarifying*: They ask a lot of clarifying questions. If given an ambiguous assignment, a consultant will ask pointed questions until the task is crystal clear (to avoid wasted effort). Similarly, in meetings, they'll often paraphrase what they heard: So if I understand, you are saying X… and the reason is Y, correct? This helps ensure accurate understanding and reveals if something doesn't add up. *Second-Order Thinking*: Top consultants think beyond immediate results to second-order effects. For instance, recommending a price cut might boost volume (first-order), but what about competitors responding or margin impact (second-order)? They train to always ask: and then what? to foresee consequences. This habit yields more robust recommendations (avoid solving one problem by creating another). *Metacognition, thinking about thinking*: They reflect on their problem-solving process itself. Are we approaching this the right way, or are we biased by something? Should we frame the problem differently? This self-awareness leads to adjustments in approach before too much time is wasted.

The Obligation to Dissent and Intellectual Honesty

McKinsey pushes the obligation to dissent as a core value. The key practices are: *Speak Up if Something's Wrong*: If a junior consultant believes the data suggests a different answer than the team's direction, they are expected to voice it, even if it contradicts a senior person. It is a duty, not just a right. *No Deference in Analysis*: While hierarchy exists, in analysis and problem-solving, the best idea should win, not the highest-ranking person's idea. This requires creating an environment where disagreement is safe (psychological safety). *Leadership Encouraging Dissent*: Good managers and partners explicitly ask for critiques: What are we missing? Do you all agree with this? And they model openness. If a mistake or a better idea is shown, they acknowledge it rather than defend their ego. *Healthy Debate*: Teams often have vigorous debates about interpretations, sometimes even heated, but they know it is in service of the solution. It is typically not personal. They back opinions with evidence or logic. A phrase like "Let us disagree on the facts" might be used ironically to ensure they check facts to resolve a debate. *Avoid Groupthink*: Dissent culture is essentially anti-groupthink. Each person is thinking critically, not just prematurely aligning with the consensus. This fosters creative alternatives and catches errors. For example, if everyone is leaning towards one strategy, someone might say, Have we considered if that fails? Should we have a Plan B or a hybrid approach? That voice can save the project if that risk is real.

7. True Time Management

Time management as a structured practice has evolved significantly over the past several decades. Looking back 20 years, a number of popular systems emerged that taught professionals how to plan their days and achieve their goals. Many of these systems came in the form of comprehensive planners or binders accompanied by training programs. One influential example was Time Manager International (TMI), created in Denmark in 1975 by Claus Møller. The TMI system was built around the idea of *key areas* of life. It encouraged users to split every aspect of life into major categories and set goals for each. In the original TMI approach, there were typically nine key areas (about six related to business/professional life and three to personal life), and an additional tab dedicated to bright ideas that might occur at any time. This structure helped people ensure that all important dimensions of their lives received attention, rather than just reacting to an endless list of unrelated tasks. TMI used a metaphor called the *Christmas Tree Principle*: the trunk of the tree represented one's overall life goals, the branches were the key areas of focus, the twigs were projects, and the pine needles were individual to-do items. This visual reminded users that tasks (the needles) should be attached to larger projects and areas (the twigs and branches), preventing them from getting lost in a disorganised flurry of activities. In fact, the TMI system explicitly warned against *flapsi hapsi*, a Danish colloquialism for chaotic busyness wherein one is surrounded by hundreds of to-dos with no overview or clear priorities. By organising tasks under key areas, users could maintain clarity on priorities and make sure their daily actions were aligned with broader goals.

Around the same era, other planner-based time management systems gained popularity worldwide. In the United States, brands like Franklin Quest (later Franklin-Covey), Day-Timer, and Day Runner offered thick ring-binder organisers with preprinted pages for calendars, task lists, goal planning, and more. The Franklin system, founded by Hyrum Smith in the 1980s, taught users to identify their governing values and long-range goals and then break them down into quarterly, weekly, and daily plans. Users of the Franklin planner were encouraged to prioritise their daily tasks (often marking them as A, B, C priorities) and even assign numbers to sequence tasks (A1, A2, etc.) each day. This method introduced the idea that not all tasks are equally important and that one should tackle high-priority items first. Stephen R. Covey, a management thinker who later merged his company with Franklin, further built on these concepts in the early 1990s. Covey's approach, as described in *The 7 Habits of Highly Effective People* and *First Things First*, emphasised organising one's life around roles and principles. Covey suggested that individuals define their key roles in life (e.g. role as a professional, parent, community member, self-care, etc.) and set weekly goals for each role. This ensured a balanced focus instead of letting urgent work projects completely overshadow other responsibilities. Covey also introduced the Time Management Matrix, which categorised activities by urgency and importance. He urged people to spend more time on so-called Quadrant II tasks that are important but not urgent (such as strategic planning, relationship building, exercise, and preparation) because those activities contribute most to long-term success and prevent crises. The older planner systems often incorporated these ideas: for example, Franklin planners had weekly compass cards where users wrote big rocks (important goals) in each role and daily pages where tasks could be tied back to those roles or values.

European counterparts to these systems also flourished. In addition to TMI, another Danish system called Time/system (introduced in 1981) became widely used. Like TMI, Time/system provided a binder with calendar and planning pages, and it used the slogan *Right things at the right time*, highlighting its focus on prioritisation. In the UK, James Noon's Time for Success (A' Time) gained traction in the late 1980s. These systems all shared common themes: they provided structured paper planners divided by sections (tabs for different planning tools or life areas) and came with methodologies to help users clarify their objectives, plan ahead, and manage tasks systematically. Notably, they often combined long-range planning (annual or monthly goal setting) with medium-range (weekly) and short-term (daily) planning and taught how those levels relate. For instance, a core principle was to always plan the upcoming week before it begins, often on a Friday afternoon or Sunday evening, by reviewing one's commitments and goals. This practice helped prevent starting the week without direction. Fixed dates like deadlines, birthdays, or appointments would be written down first, and then time blocks for important tasks or goals would be allocated next. By having monthly, weekly, and daily pages visible (some TMI and Time/system binders could be unfolded to display all three views side by side), users could maintain alignment between their day-to-day actions and their long-term plans.

However, despite their effectiveness, many of these early systems were physically and administratively heavy. A trademark of the 1980s and 1990s time management was the bulky planner binder. Often, a thick leather or vinyl book the size of a large tablet or notebook, stuffed with pages, dividers, fold-out calendars, notepaper, and even plastic pouches for business cards or receipts. Professionals carried these everywhere, to meetings and even on personal outings, which could be seen as excessive. Satirically, the yuppie image of the 1980s included carrying a Filofax or Franklin planner as a status symbol. Indeed, the Filofax (a popular loose-leaf pocket organiser) was sometimes mocked for being more about looks and appearances than utility. While dedicated users did get value from these planners, there is no denying that lugging a thick calendar binder everywhere was rather inconvenient. The binders could weigh a kilo or more once filled, and their users had to routinely archive old pages and insert new ones (e.g. replacing an entire year's calendar each January), which was a laborious process. Moreover, these systems demanded a significant *time investment* (sic!) just to maintain the planning ritual itself: one had to daily rewrite unfinished tasks, file away notes in the proper tab, and meticulously uphold the organiser's structure. In some cases, people spent more time color-coding or fine-tuning their planners than actually executing the work, a form of procrastination-by-planning that earlier time management literature warned against.

Despite those drawbacks, the core ideas from these classic systems have proven enduring. The excessive forms (physical binders and ultra-detailed paperwork) have faded with time and technology, but the *principles* continue to influence modern productivity methods. Two such key ideas were: (1) structuring life by its major domains or roles, and (2) linking daily actions to long-term goals. As we saw, TMI explicitly divided life into major *Key Areas*, and many users who trained in that system report that decades later they still organise projects and even digital files according to those areas. Likewise, Covey's role-based weekly planning encourages identifying key life roles (up to about 7) and ensuring each role gets some attention in one's plan. This notion of comprehensive life management, not just managing work tasks, but also personal, family, and self-development tasks, is a valuable legacy of the old systems. It reminds us that time management is ultimately about aligning how we spend our time with what truly matters to us across all facets of life.

Another lasting idea is the value of routine and habit in freeing the mind. The old masters of time management often advised developing regular planning habits (like a weekly review every Friday, or a morning ritual to prioritise tasks). The underlying rationale is supported by psychology: making certain behaviours automatic can reduce the mental effort required for the day-to-day organisation. Over a century ago, William James observed that strict routines free our minds to advance to really interesting fields of action by rendering many daily decisions automatic. In other words, you don't have to agonise over when to do your admin tasks or whether to go for your afternoon walk. Because you've established a fixed time for them, you conserve mental energy for creative and strategic thinking. Modern cognitive science backs this up: research on limited cognitive bandwidth and willpower shows that if we waste resources deciding trivial scheduling matters, we impede our capacity to focus on meaningful work. The old paper systems, by encouraging users to follow a consistent planning regimen and daily structure, were tapping into this insight. Many highly creative figures in history intuitively adopted similar approaches, maintaining strict daily schedules or rituals. For instance, the novelist Gustave Flaubert wrote late into the night while Le Corbusier woke at dawn for exercise, but they each did what they did with iron regularity. The poet W.H. Auden advised to decide what you want or ought to do each day, then do it at *exactly* the same time every day and passion will give you no trouble. This reflects the same principle: a stable structure can act as a safety net, preventing the chaos or existential terror of no structure at all that often paralyses creative people when they have a completely free choice. So while we may no longer carry 300-page binders or summon our assistants with a bell or ceiling tap, the wisdom of planning around life categories, prioritising important over merely urgent tasks and building steady routines remains highly relevant today.

Before moving on to modern methods, it is worth noting one more transitional framework that emerged in the early 2000s: David Allen's *Getting Things Done*

(GTD). GTD became a bestseller in 2001 and is now a classic in productivity/management literature. It took a somewhat different approach from the traditional top-down planners. Allen's system focused on capturing every single task or commitment in an external system (a series of lists and folders) and defining the next action for each. By doing a thorough mental inventory, the practitioner aims to achieve a mind like water, a mental state free of stress, because all obligations are tracked somewhere reliable. GTD is often described as a bottom-up approach: it doesn't initially emphasise life roles or long-term goals, but rather getting control of the flood of daily tasks first, then later organising them by projects and desired outcomes. Many creative and entrepreneurial people were drawn to GTD's promise of stress-free productivity and its flexible nature that did not require carrying a single huge planner (one could implement GTD with simple folders or with software). However, some found that GTD could become too granular and overwhelming, with its exhaustive lists of next actions and contexts. As one commentator noted, GTD is powerful but not for the faint-hearted. It can take days of work just to set up the system and capture everything, though the payoff is supposed to be that once you have set up the system, it is worth the time and effort. GTD's rise indicated that by the 2000s, people were seeking new ways to handle the increasing complexity of work in the digital age. In fact, the 21st century brought new challenges: always-on email, back-to-back meetings, and the disappearance of clear boundaries between work and personal life. These challenges set the stage for rethinking time management for modern creative personal innovators.

The Modern Challenge for Creative Personal Innovators

Today's personal innovator operates in a vastly different environment than the professionals of a generation ago. The nature of work and life has become more digital, dynamic, and distributed. We are surrounded by powerful tools (such as smartphones, collaborative software, and endless information via the internet) but also by endless distractions and demands on our attention. Paradoxically, while classic time management taught people to shield their time from interruptions, modern technology has made interruptions more frequent than ever. Studies show that the average office worker or innovator switches tasks astonishingly often. One observational study found that the average office worker switches tasks every three minutes, and once distracted, it can take up to 20+ minutes to regain focus. In fact, researchers at UC Irvine quantified that it takes about 23 minutes and 15 seconds to return to the original task after an interruption. This means that a single 30-second glance at a phone notification or a quick check of social media can cascade into a half-hour of lost productivity. Multiply these interruptions over a day, and it is clear how a creative project can stall despite one's best intentions. All these distractions not only eat up time but also increase stress. Attention distraction can lead to higher stress, a bad mood, and lower productivity, according to research by Gloria Mark of UCI.

For a creative professional working with Design Thinking, these challenges are especially pronounced. Design Thinking as a methodology thrives on phases of open-ended exploration, brainstorming, and empathetic understanding, activities that require deep focus and often large unstructured blocks of time. Creative innovators often experience the *flow* state when working on an idea, that immersive focus where hours pass by unnoticed, once coined by Mihály Csíkszentmihályi. Indeed, creative people are often perceived as having trouble with time in the conventional sense: they might be late to meetings or miss deadlines because they get absorbed in their work. Psychologists note that the *flow phenomenon* can make one lose track of the clock. While flow is wonderful for producing creative breakthroughs, the surrounding responsibilities and appointments still need to be managed. This is the modern creative worker's dilemma: how to protect and cultivate those precious periods of creative flow and deep work, while still handling the myriad of *administrivia* and external demands that won't go away. Without a strategy, the inventive individual may either become overwhelmed by emails, meetings, and minor tasks, leaving little time for creative thinking. Or conversely, they may focus exclusively on their passion projects and let practical matters slide. Either imbalance can lead to trouble.

Another aspect of the modern context is the blend of professional and personal life. With remote work, flexible hours, and the gig economy, the boundaries between work time and personal time have been significantly blurred. While earlier generations might have had clearer work hours and then personal hours, today one might answer work emails from home at 9 PM and instead deal with a personal finance task during a 2 PM work break. This integration can be positive (more flexibility), but it also creates pressure to constantly micro-manage time slices and switch contexts. Personal innovators in 2020 typically juggle multiple projects or roles: for example, one might be designing a new product (core job), while also freelancing on a side project, learning a new skill online, maintaining a household, caring for children, and trying to squeeze in hobbies or exercise. Each of these roles generates its own stream of tasks and to-dos. It is no wonder that people often feel like there is an endless list of things to keep track of, leading to a mental burden of remembering and planning each small step. In many cases, individuals resort to *microplanning*, constantly figuring out what the next thing they should do now is throughout the day in an ad-hoc manner, which is mentally exhausting.

Statistics highlight how time is spent in suboptimal ways when there isn’t a clear system. A well-cited McKinsey study found that office workers spend about 61% of their time on busy work like emails, information searches, and communicating internally, leaving only 39% for their actual role-specific, skilled work. Similarly, the Wall Street Journal reported that 40% of an average professional’s workweek is consumed by meetings, phone calls, and email, activities that often cut into time that could be used for creative or strategic endeavours. For a personal innovator, this is a troubling allocation because it means their creative energy is being drained by reactive tasks. Thus, a modern time management approach needs to help carve out and protect significant time for deep, creative work, while efficiently handling the necessary communication and administrative tasks.

Another challenge is maintaining a healthy work-life balance and personal well-being while pursuing creative innovations. There is a growing understanding that creativity doesn’t happen in a vacuum. It is fuelled (or hindered) by factors like mental health, physical health, relationships, and downtime. A personal innovator typically pours a lot of themselves into their work. But without proper time management, they might sacrifice sleep, skip exercise, or neglect family, all in the name of a project. In the short term, this might yield a burst of output, but in the long run, it often leads to burnout or a collapse in productivity. Especially in creative fields, burnout is the enemy of innovation; a burned-out mind struggles to generate new ideas or insight. Modern thinking thus emphasises sustainable creativity, balancing intense work with rest, and achievement with renewal.

Design Thinking, which the user practices both professionally and personally, also provides some insight into managing this balance. One of the pillars of DT is being human-centred and empathetic, and that includes being empathetic to oneself. There is a concept of designing your life that has emerged in recent years, spearheaded by authors like Bill Burnett and Dave Evans from Stanford. They propose that we can apply design principles to craft a well-lived life, not just solve product problems. Notably, in Burnett and Evans' framework, life is viewed in terms of four domains: Work, Love, Play, and Health. This is quite similar to the categories we will discuss shortly. The idea is that to achieve a fulfilling, creative life, one must prototype and iterate their lifestyle to include all these elements in harmony. They even suggest measuring how full each of your four tanks, work, love (relationships), play (hobbies/leisure), and health (physical and mental), are, and adjusting your activities if one area is deficient. This kind of holistic, design-inspired approach acknowledges that being an effective personal innovator or creator is not just about managing tasks. It is about designing your time and habits in a way that nurtures your creativity and happiness.

So the modern creative personal innovator faces a double mandate: be productive and innovative in work and also maintain personal well-being and balance. There is a need for a time management method that is tailored to this reality, one that is flexible enough to handle the fluidity of modern work, structured enough to prevent chaos, and comprehensive enough to cover professional and personal priorities. The method should help reduce the *cognitive load* of constant small decisions (micro-planning every hour), allowing the mind to focus on higher-level creative thinking. It should also integrate concepts from Design Thinking, such as iteration and empathy, to allow the person to continually refine their system and ensure it suits their evolving life. In the next section, we introduce a modern framework that attempts to meet these needs by dividing time and goals across four main areas of life: Work, Spare Time, Family, and Personal. This framework draws inspiration from the old key areas concept while updating it for the way we live and create today.

Time Management for the Personal Innovator

To address the challenges outlined above, we propose a structured yet flexible method of organising time, tailored for a creative personal innovator who must juggle professional projects, personal passions, relationships, and self-care. The framework is built around four main areas of life:

1. *Work* – your professional projects, creative ventures, and career-related activities.
2. *Spare Time* – your hobbies, leisure pursuits, and recreational activities (the play component of life).
3. *Family* – your relationships and responsibilities towards family and close loved ones (which we can broadly equate to the *love* domain).
4. *Personal* – your individual well-being and personal growth activities, further divided into four subcategories: North, East, West, and South (N-E-W-S), which will be explained shortly.

This division intentionally covers both professional and personal spheres, recognising that an accomplished personal innovator needs to allocate time to each. It echoes ideas such as TMI's key areas, or Burnett and Evans's Work/Love/Play/Health while providing a novel structure that can be adapted to modern lifestyles. The strength of this approach is that it prevents any one aspect of life from completely eclipsing the others. By explicitly laying out these four buckets, we create a mental model that your time is not one monolithic resource to be given entirely to work or entirely to others. Instead, it is meant to be distributed among these key areas intentionally. Let us briefly introduce each area and how it is intended to be used:

- *Work*: In this framework, Work encompasses all the professional or creative projects you are involved in. Instead of treating work as one amorphous category, you will track each major project as a separate entry, plus an entry for general or minor work tasks (Other tasks). For example, if you are working on designing a new prototype, writing a research paper, and consulting on a side project, each of these would be listed separately under Work, along with an Other Work entry for miscellaneous duties (emails, administrative paperwork, etc.). The rationale is to ensure you dedicate planned time to each significant project and avoid neglecting one because another is consuming all attention. It also allows you to plan project-specific next actions and goals in an organised way.

- *Spare Time*: This area covers all hobbies, interests, and leisure activities you wish to pursue in your free time. Here again, you will list one entry per hobby or major pastime. If your spare time activities include, say, painting, playing guitar, reading novels, and hiking, each of these gets its own slot. By naming and recognising your hobbies in your planning system, you are less likely to let them fall by the wayside when life gets busy. It is a gentle way of holding yourself accountable to enjoy life outside of work. Spare Time isn't merely leftover time, but a category to be planned and valued.
- *Family*: This area pertains to time and tasks involving family and close relationships. The framework suggests having one entry per close family member (or significant other) and additional entries for shared family responsibilities or assets (for example, one for Household/House, one for Car if you own a car, etc., as relevant to your life). Each person who matters deeply, such as your spouse/partner, each child, or perhaps an aging parent you care for, is given a dedicated place in your planning. This way you can plan things like one-on-one time, remember important dates (birthdays, anniversaries), or tasks specific to that person (helping with homework, planning a surprise gift, having a deep conversation they've been requesting). The category entries like House or Car ensure that maintenance tasks (fixing appliances, scheduling a service, paying bills, shopping for groceries) are not ad hoc afterthoughts but are part of your system. In essence, Family covers both the emotional/social investments in loved ones and the practical chores of domestic life.
- *Personal* (NEWS): The Personal area is about *you* and your self-maintenance and growth. It is subdivided into four aspects, conveniently labelled as North, East, West, and South. Think of it as the four points of a compass guiding your personal development and well-being:

- *North*: Stands for *True North* in daily life – essentially your time planning and daily framework. It involves how you orient each day, including daily scheduling, routines, and a time-blocking checklist for recurring tasks.
- *East*: Symbolises *Sunrise* or enlightenment – referring to mindfulness, meditation, or spiritual practice. This is about dedicating time to quiet reflection, mindfulness exercises, prayer, or any practice that feeds your spiritual or mental calm.
- *West*: Symbolises *Sunset* or introspection – referring to self-reflection, journaling, or therapeutic activities that help you process emotions and experiences. It could include writing a journal, attending therapy or counselling sessions, or doing personal reflection exercises.
- *South*: Symbolises the foundational *Ground* – referring to bodily health: diet, food, exercise, and sleep (essentially your physical well-being routines).

The NEWS acronym not only helps remember the four subcategories but also implies a 360-degree, well-rounded coverage of personal well-being. We will explore each of these in detail later on. The main idea is that the Personal area ensures you take care of yourself because creative knowledge work is mentally and physically demanding. Just as a compass must be balanced in all four directions to give accurate orientation, a person should maintain all four aspects, daily structure, mindfulness, self-reflection, and physical health, to stay balanced and oriented in life.

By structuring plans around these four areas, the framework does something powerful: it externalises and organises all the arenas where your time and energy are needed, reducing the mental load of keeping track in your head. Instead of constantly asking yourself What should I be doing now? Did I forget something important? You will have a map of sorts that you can consult. Each area can be reviewed regularly (for instance, in a weekly review session) to ensure nothing critical is slipping through the cracks. This approach resonates with the old time-management wisdom of getting an overview (recall TMI’s emphasis on avoiding an un-prioritised heap of tasks) and it also addresses the modern need to juggle multiple roles.

Let us now delve into each of the four areas, Work, Spare Time, Family, and Personal, to discuss conceptual and practical aspects of how they support a creative, productive life while minimising the burden of microplanning. For each area, we will examine why it is important for a personal innovator type of individual and how, in practice, one might manage time within that category. Throughout, the emphasis will be on maintaining a balance that fosters creativity and reduces stress.

Work: Focusing on Creativity and Managing Projects

For a personal innovator or any creative professional, the Work category is central. It represents the arena where you bring your designs, ideas, and projects to life. Conceptually, the key challenge in the Work area is finding focus amid multiple projects and tasks. Creative people often have more ideas than they have time; it is common to be working on, or at least ideating on, several projects concurrently. Without structure, this can lead to jumping between tasks impulsively or giving disproportionate attention to the project that currently sparks the most interest, while others languish. The framework's solution is to explicitly list each major project as its own entry and plan time for it. This creates a project portfolio view of your work: at a glance, you see all the big efforts on your plate. This helps in various ways:

Prioritisation and Goal Setting: By listing projects separately, you can set specific goals or outcomes for each. For example, under *Project A* (say, designing a prototype), you might have a goal "Complete initial prototype by June 1" and a next step "Sketch out component X this week". Under *Project B* (writing a report), a goal could be "Finish literature review this month". If everything were lumped under generic work, these project-specific objectives could be overlooked. The separate entries encourage you to think strategically about each project in turn, which is akin to how a project manager would review each work stream. This ensures that important long-term projects (perhaps those without urgent deadlines but of high importance) get the attention they deserve. Make sure you categorise every major task in every project according to the DPS principle: driver, passenger, spectator. Either the progress of the task depends on your initiatives, i.e. you are the driver. Or you participate actively but the progress is monitored by others, i.e. you are a passenger. Finally, maybe you are not directly involved in the task but its results and outcome might influence you, i.e. you are a spectator.

Balanced Allocation of Time: During your weekly or daily planning, you can allocate blocks of time to each project entry. For instance, Monday morning 9–11 AM is blocked for Project A, Monday afternoon for Project B, and so on. By doing this intentionally, you guard against a scenario where one project (maybe the most urgent or the most exciting) eats up all your time while others fall behind. It is a bit like crop rotation for your mental fields: dedicating time slices to different projects keeps them all moving forward and keeps your mind fresh by providing variety. Moreover, by including an Other Work entry for miscellaneous duties (emails, minor tasks, administrative forms, quick favours, etc.), you confine those small tasks to their own block of time rather than letting them continuously interrupt your major work blocks. This is crucial: one of the biggest drains on creative productivity is constant context-switching. If you bounce from an email to a coding task, to a meeting, and back to writing a design brief, you never reach the deep focus needed for quality creative output. Research confirms this: workers who allow frequent interruptions take a significant hit in performance and need a long time to refocus. Therefore, batching Other tasks (like emails) into a defined period (say, 30 minutes in the late morning and 30 minutes late afternoon) can greatly reduce random interruptions during your core work periods.

Stress Reduction and Memory Offloading: Knowing that each project has its place in your schedule reduces mental stress. You are less likely to lie awake at night thinking, “Project X is behind, when will I ever get to it?” because you have a plan. Perhaps Project X time is always Wednesday and Friday mornings. This turns worry into a concrete appointment with yourself. It also means that if an intrusive thought about another project comes up while you are working on the current one, you can note it down for that project’s next session (or in a bright-ideas capture list) and continue what you were doing, confident you won’t forget it. This practice is reminiscent of GTD’s idea of capturing tasks to *get them off your mind*, and it is very relevant for a creative mind that’s always generating ideas. For example, if you are focusing on Project A and suddenly have a great idea for Project B, instead of impulsively switching, jot it in Project B’s notes or task list. Because you know you have Project B time scheduled, you can trust that the idea will be processed later.

This habit prevents the fragmented schedule that so many personal innovators fall victim to, leading to lots of half-finished projects.

From a conceptual standpoint, the Work category structured this way supports creativity by providing *clarity and focus*. Instead of feeling overwhelmed by the abstract notion of Work I have to do, you have a concrete map: *these* are the projects I am working on, *these* are the steps for each, and *here* is when I will work on them. This clarity can liberate creativity. Many creative individuals actually thrive under some constraints. Having a fixed time or deadline for a task can spur inventive solutions (think of hackathons or creative competitions as examples where a time limit boosts creativity). When you time-block your project work, you are giving yourself a healthy constraint: in the next two hours, my goal is to draft the outline of Chapter 1 of my book. This focused objective can spark intense creative effort, in contrast to an open-ended I'll work on my novel whenever I feel inspired which may or may not happen consistently.

It is important, however, that the structure remains *flexible and humane*, especially since Design Thinking work can be nonlinear. If a flash of insight strikes or one project truly needs extra time in a given week, the system can accommodate moving blocks around. The aim is not rigidity for its own sake, but rather intentionality. As Peter Drucker noted, one of the keys to knowledge work productivity is working on the *right task* at the *right time*. By reflecting on your projects and planning your week, you increase the chance that at any given moment, you are working on what is most important (or at least, not forgetting something important).

In practice, using the Work category might look like this: each week, you review all your active projects. For each, you ask: What is the next milestone or deliverable? What progress do I need to make this week? Then you schedule time accordingly. You might discover you have too many projects to make meaningful progress on all. This insight is valuable and may lead you to defer or drop some, or seek help. The system shows you your capacity. On the other hand, if you find one project has light needs this week, you can double up time on another that's a priority. During each day, when it is time for Project A block, you focus just on Project A tasks; you resist doing Project B or random things in that time. If something unrelated comes up (say a colleague calls), you either defer it to the appropriate block (Sure, I'll handle that in my Other tasks time at 4 PM) or, if it is truly urgent, at least you know which project's time you are sacrificing and can reschedule accordingly. This mindful trade-off thinking is far better than an unconscious reaction to every interruption.

Finally, consider how this approach reduces microplanning. Microplanning is when you constantly shuffle tasks and make decisions hour by hour about what to do next. Under this framework, those decisions are mostly made in your weekly planning session and each morning when you review the day's plan. During the day, you follow the plan, trusting the process you set up. Of course, unexpected events happen, and you adjust if needed, but you are not starting from scratch every hour. The mental relief from this is significant. As William James and later researchers observed, having routine and predefined plans conserves mental energy. You don't wake up each day with a hundred options and decisions; you wake up knowing today you will focus on Project A and B (for example) as planned, and you can direct your creative energy into those projects instead of into deciding what to do. This structure creates a positive discipline, which paradoxically *frees* the creative mind. In the words of W.H. Auden, once you've resolved what you are to do at a given time, it will be far more likely to happen and you won't be distracted by inner debates.

So the Work area of the framework, with one entry per major project and an additional Other Tasks entry, serves to focus the inventive mind, ensure balanced progress, and eliminate a great deal of the mental noise that comes from unorganised multitasking. It provides a stable platform upon which creativity can play. With Work time well-managed, you gain efficiency that actually leaves more room for innovation because your energy isn't sapped by task-hopping or by crises that arise from neglecting parts of your work. As a result, when you do engage in a Design Thinking session (brainstorming ideas, prototyping, etc.), you can be fully present, knowing that other projects and duties have their own time allotted.

Spare Time: Investing in Hobbies and Restful Play

The Spare Time category is often undervalued in traditional productivity circles, but it is absolutely vital for a creative individual. This category encompasses your hobbies, recreational activities, and any pursuit done for enjoyment or personal interest rather than for pay or obligation. Conceptually, Spare Time is about play, exploration, and renewal. It is the time when you step away from work per se, yet often these are the moments when your subconscious mind makes new connections, or when inspiration strikes from an unexpected direction. There is truth in the saying that *creativity flourishes when the mind is at play*. Many great innovators and artists have gotten ideas during leisure activities. Taking a walk, playing a musical instrument, cooking, or even daydreaming.

By including Spare Time as one of your four main areas, the framework is making a statement: leisure is not a luxury; it is a key area of life to be managed and cherished, much like work. It combats the modern tendency to either ignore leisure (feeling guilty for taking time off) or to spend it unthinkingly (scrolling through social media for hours, which may not actually be fulfilling leisure). Instead, you proactively identify the hobbies or pastimes that truly rejuvenate you or spark joy and give each a space. For example, if you love painting, you list Painting as a subentry under Spare Time; if you enjoy biking, add Cycling; if you play video games to relax, include that too. These become like mini-projects, but without the pressure, think of them as *joy projects*. The conceptual benefits include dedicating time to spare-time activities supports creativity and productivity in multiple ways:

- First, hobbies often serve as creative cross-training. They engage different parts of your brain and can provide fresh perspectives. A software designer who also plays guitar might find that playing music in the evening clears his mind or even gives a rhythmic insight into a coding problem. A writer who enjoys hiking might find story ideas forming during a long walk in nature. There is evidence that breaks and seemingly unrelated activities can incubate creative solutions. The classic *Aha!* moment in the shower is a testament to how stepping away from direct work can yield breakthroughs. Doing almost anything other than sitting at a desk can be the best route to novel insights. Tchaikovsky insisted on a two-hour daily walk, believing returning even a few minutes early would harm his inspiration. That might be a superstition, but it underscores how valuable many personal innovators have found leisure time to their work.

- Second, leisure activities function as restorative niches for your mental energy. Cognitive psychologists describe how intense focus (especially in knowledge work) depletes certain mental resources, and that downtime is needed to replenish them. Engaging in a fun activity you love can reduce stress, improve your mood, and thereby set the stage for more productive work later. Hobbies often induce a state of relaxed focus or even mild *flow* (e.g., a gardener completely absorbed in tending plants). This can be deeply refreshing compared to the often goal-driven focus of work tasks. By planning regular spare time, you are effectively scheduling in recovery and preventing burnout. You can think of it as analogous to the way physical training requires rest days; creative work requires mental rest periods.
- Third, having a rich personal life with hobbies makes you a well-rounded individual, which in turn can feed your creativity with more raw material. A personal innovator who only ever studies his narrow field may run out of fresh ideas, whereas one who also reads history, plays sports, and travels will have a broader palette of experiences to draw from. Spare Time is when you nurture those other dimensions of yourself. Even something as simple as watching quality films or reading fiction can expand your imaginative capacity and empathic understanding, which are useful in Design Thinking for understanding users and scenarios.

Now, from a practical standpoint, how do we manage Spare Time in this framework? Much like with Work, by listing each hobby or interest, you can set intentions and allocate time for them. Suppose your Spare Time sub-entries are: *Guitar, Photography, Reading, and Gardening*. In your weekly overview, you might set a simple goal for each: e.g., practice guitar 3 times this week, go on a photo walk on Saturday, finish reading novel X by Sunday, or spend 2 hours gardening. These are not meant to turn leisure into a chore, but to ensure that these activities actually happen and don’t get indefinitely postponed. Many people have the experience of saying “I love doing X” but realising weeks have passed without doing it, because they never specifically made time. By treating Spare Time seriously in your planning, you avoid that pitfall. You might block an evening for a hobby: e.g., Tuesday 7–9 PM for painting in your studio, or keep Sunday afternoon open for family hikes (if hiking is shared between Spare Time and Family). Paradoxically, scheduling leisure can make it more likely that you will get leisure. It is far too common to come to the end of a week exhausted and realise you had no fun at all; a bit of foresight can prevent that.

Another practical tip is to guard some portion of your week as untouchable free time. This might be a few hours designated as do-whatever-you-feel-like time. This ensures spontaneity doesn't get lost. You can spontaneously choose from your listed hobbies or even do something entirely new during that slot. The key is that Spare Time is an area you actively manage, not just leftover scraps of time after work. In terms of reducing microplanning, when you have pre-decided that, say, Saturday 10 AM–1 PM is for Spare Time: Woodworking, you are not waking up Saturday thinking "Hmm, what should I do today?" and then possibly defaulting to whatever is easiest (such as mindless web-browsing). You have a fun plan you decided on when you were in a rational, big-picture mindset. Then you can immerse in it guilt-free.

It is worth addressing a potential concern: does structuring spare time take the spontaneity or relaxation out of it? If done too rigidly, it could, but the idea here is not to micromanage every minute of fun, but to ensure that fun happens and that it is the kind of enriching fun you truly want. There is still plenty of room for spontaneity. Think of it this way: you might block time for Reading but you can choose whatever book you are in the mood for at that moment. Or you plan Saturday as a hiking day but you can pick the trail spontaneously. The structure is just enough to protect that time from being overtaken by work or chores.

The Spare Time category also has an indirect benefit on the others: it provides breathing room in life. When you know you have leisure activities to look forward to, it can actually improve your focus during work time. "I'll push through this task now, and this evening I get to go play basketball with friends". It is a psychological carrot that keeps motivation high and prevents feeling trapped. Conversely, if one skips all leisure, work can start to feel like a grind and procrastination may increase since the mind rebels when denied any play.

In Design Thinking, there is this concept of divergent and convergent thinking. Divergent thinking (generating many ideas, exploring) requires a relaxed, open mindset, whereas convergent (choosing, implementing) thinking is more focused.

Spare Time often encourages a divergent mental mode. You might stumble on an idea for a work project precisely when you are tinkering in a different medium with no pressure. Many famous inventors have cited *hobbies or unrelated interests as sources of innovation*. For example, and connecting directly to DT, the inventor of the Post-it note, Art Fry, got the idea of a lightly adhesive bookmark through a mix of work on adhesives and his experience singing in a church choir where he needed bookmarks that wouldn't fall out. That kind of cross-pollination is more likely if you cultivate varied interests.

In practical terms, during your weekly planning, after scheduling your Work blocks, you schedule your Spare Time activities. Treat them as important appointments with yourself. If someone tries to intrude on that time (say, a non-urgent work request for a Saturday when you planned a family hike), you can often say no or negotiate, because you recognise that time has value, even if it is not money-generating. Of course, flexibility is there if something truly important arises, but the default is that your leisure plans are real plans. This mindset can be initially foreign to those who are used to sacrificing personal time whenever work calls, but over the long run, it leads to a healthier balance and actually higher productivity during work hours.

By managing Spare Time in this way, the mental burden of constant microplanning on weekends or evenings is reduced. You are not constantly saying, "I really should do something fun… but I also have chores… maybe I'll just do nothing." Instead, you have an outline (e.g., Saturday morning: hobby, afternoon: family event, evening: free; Sunday: maybe a long bike ride planned, etc.). Having a plan for rest is as important as having a work plan. It means you can truly relax during your leisure because you've set aside that time deliberately and you know you are allowed to enjoy it without guilt or anxiety about work. This last point is much more important than it seems at first sight, since many burnouts come from never allowing play or rest without the accompanying guilt.

In conclusion, the Spare Time area supports a creative and productive life by ensuring regular doses of play, relaxation, and personal enrichment. This not only makes life enjoyable (which is a worthy goal by itself) but also fuels your creative mind with new energy and ideas. By planning and protecting spare time, you reduce decision fatigue (Should I work or relax now?) and prevent burnout, thereby indirectly making your productive time more effective. As a personal innovator, consider your hobbies as part of your creative ecosystem. They keep the ecosystem diverse and resilient. The framework's inclusion of Spare Time as a core area is a reminder that great ideas often blossom in the garden of leisure.

Family: Nurturing Relationships and Fulfilling Responsibilities

The Family category addresses the domain of relationships, home, and loved ones. For a holistic time management system, this area is just as crucial as work, because our connections to others and our home life form the foundation that supports our professional and creative endeavours. A personal innovator might be deeply focused on projects, but if they neglect family commitments or relationship needs, stress and conflict can quickly spill over and undermine their creativity and productivity. Conversely, a strong, positive family life can provide emotional support, inspiration, and a sense of purpose that fuels creative work. As research and long-running studies have shown, healthy relationships are a key component of happiness and life satisfaction, even much more so than fame or wealth, and a happy personal innovator is likely to be a more creative and effective one.

Conceptually, dividing Family into sub-entries of individual people (and key household categories) achieves a couple of things. It underscores that each important person in your life deserves mindful attention. In practice, life's urgent tasks often crowd out time with family unless it is deliberately planned. By having, say, Spouse/Partner, Child A, Child B, Mom, etc., each as an entry, you remind yourself to consider what you want to do for or with each of those people on a regular basis. This echoes Covey's role-based planning, where family-member roles would be explicitly planned for. The framework brings it into a concrete form. Not just a role in the abstract, but a pipeline of plans and tasks per person. For example, under Alice (daughter), you might note: attend her school play on Thursday, plan a father-daughter outing this month, and help her with the science project by Tuesday. Under Bob (brother), perhaps: call him on his birthday, check in about how he is settling in his new job. Under Spouse/Partner, you might plan a weekly date night or note something thoughtful, such as picking up his/her favourite dessert for Friday dinner. These aren't tasks in a cold sense, they are acts of relationship maintenance and love.

Having them written down doesn't make them impersonal; it makes it more likely that you will actually do them, especially when busy. If it still feels that way, just don't mention where the reminder came from. In terms of supporting a creative, productive life, maintaining your family and home life in good order has multiple benefits:

- Emotional Stability and Support: A contented family life provides emotional security. When your partner and kids feel cared for and your home is running relatively smoothly, you as the innovator have the peace of mind to fully engage in creative work without guilt or worry. It is hard to concentrate on an innovation if you just had an argument at home or if you are anxious about neglecting your child's needs. By proactively scheduling family time and attending to issues, you prevent many such negative situations. Plus, loved ones often cheer us on. Their encouragement can boost confidence and resilience. Knowing you have dinner with your family in the evening, for example, can act as a stress buffer during a tough day: you have something warm to look forward to.
- Perspective and Inspiration: Family interactions can also inspire creativity. Talking with a child can spark playful ideas; explaining your project to a non-expert spouse can clarify your own thinking; helping a family member can strengthen empathy, which is crucial in Design Thinking (understanding users often parallels understanding people close to you). Also, family life has its share of problems to solve. Many innovators have gotten ideas by trying to solve a personal or household problem (a classic example: an innovator creates a gadget to help their aging parent with daily tasks, then realises it has broader market potential). By being engaged in family life, you stay connected to everyday human experiences, which can ground and inform your creative work to make it more human-centred.
- Preventing Crises Through Planning: On the practical side, listing entries like House and Car under Family covers the maintenance of your living environment and essential assets. This is important to reduce *firefighting*. If you schedule Car: annual inspection in March or House: fix roof leak before

> rainy season as tasks in your Family category, you are less likely to be blindsided by a breakdown or emergency later. When your personal life is well-managed, it means fewer sudden interruptions to your work due to neglected chores turning into urgent problems (e.g., a car breakdown on a day you have an important meeting, or a missed bill leading to service cut-off). Thus, a bit of time management in household affairs directly saves time and stress down the line. This reflects the old time-management advice to do Quadrant II activities (important but not urgent) in personal life as well, like maintenance, so they don't become urgent later.

From a practical standpoint, how can one manage the Family category effectively without making the family feel like a business project? The key is to use the planner as a servant to your values. You are not mechanising care; you are ensuring that the *expression* of care isn't lost in the shuffle of life. Practically, you might have a section in your planner for Family. Under each name or subcategory, jot down things as they come to mind: maybe your spouse mentioned wanting to go see an exhibit. Put that under his or her name as an idea. Maybe you notice the lawn is getting high, note that under House. Then, during weekly planning, review these. Decide what you will do this week: maybe schedule Saturday morning for lawn mowing (House task), and Saturday evening to take your wife to the exhibit. Also, see if any birthdays or anniversaries are within the next month so you can plan gifts or calls in advance (this is fixed-date planning similar to what classic systems recommended for personal life events).

Time-blocking family commitments is as important as blocking work tasks. If every evening from 6–9 PM you dedicate to family time (dinner, kids' homework, talking, relaxing together), mark that in your mental or actual calendar as unavailable for work. That boundary will help you be present with family and also help you work more efficiently before and after, knowing those hours are non-work. Many successful people attribute their ability to maintain a work-life balance to explicitly reserving family time. For example, you might implement a rule like no work after dinner, or Sundays are family days and treat those as sacred. By planning ahead (perhaps you and your partner coordinate schedules each week), you reduce last-minute conflicts. This also reduces microplanning: you are not every day negotiating

"Should I work late or go home?" because you've set the norm that, unless exceptional, you go home.

The Family category also encourages sharing the load of planning with family members. For instance, you might maintain a shared family calendar for events or a shared task list for groceries and chores (we will touch on digital tools for this later). This way, family time management becomes a collaborative effort, not solely on your shoulders. But your personal system will remind you to check in on those shared obligations.

One might ask, what about friends or community? The Family category can be stretched to include very close friends (perhaps under a Relationships subheading) or you could treat social life under Spare Time if it is more about outings with friends. The framework is flexible and open-ended and the principle is inclusive: nurturing important relationships, whether family or close friends, should be part of your time design. By actively managing the Family area, you greatly reduce the mental burden of "I really should be doing X for my family" which can nag at you while working, or vice versa, the worry about work while you are with family. When you plan both, you can be fully present in each sphere at the right time. It is a truism that no one on their deathbed wishes they spent more time at the office. But at the same time, one has professional ambitions and creative dreams. Good time management allows you to pursue those dreams *and* be there for your family. Structurally, this means putting family events on your calendar first sometimes. Covey advised writing in the big rocks first in your schedule, and family commitments are often big rocks.

This category, like Spare Time, is a guard against the system becoming lopsided. It injects humanity and heart into your planning. The outcome of giving family its due time is a happier life and often a more motivated and inspired creator. The inventor Dr. Yoshiro Nakamatsu, in an interview about creativity, said: It is crucial to be able to find the time and the freedom to develop your best ideas. Knowing that your family is taken care of and that you are not neglecting loved ones is one way of granting yourself that freedom, the freedom from guilt and relational strain, to truly dive into your creative work when the time comes. Finally, in terms of reducing microplanning, family life can be a source of constant small decisions (Who's picking up the kids today? What do we cook for dinner? When will we visit the grandparents?). By having a family plan or rhythm set up (perhaps a weekly family sync-up or using the framework to note all these tasks), you minimise day-to-day scrambling. You might, for instance, decide each Sunday evening what the week's dinners will be and who's cooking, and then it is settled, no daily negotiations. That's one less set of decisions each day, freeing mental space. Many families find that a bit of planning (meal plans, chore rotations, and scheduled activities) actually brings more peace and more time for fun.

So the Family area of this framework emphasises that relationships and personal responsibilities are a top priority that can be managed proactively, just like projects at work. Doing so creates a stable, supportive environment in which a personal innovator can thrive. It aligns your time with your values. If you say family is important, your calendar and task list should reflect that. By planning for family, you build a life where both your creative passions and your loved ones prosper, without perpetually stealing time from one to give to the other. This balanced approach reduces stress and constant planning on the fly, thereby contributing to an overall *mental clarity and focus* that benefits all areas of life.

Personal: The NEWS Compass for Self-Care and Growth

The Personal category is all about *you*. Your individual well-being, growth, and daily functioning. It is divided into four subcategories: North, East, West, and South (NEWS), each representing a vital aspect of self-care or self-management. The idea is that, much like a compass, tending to all four directions keeps you oriented and balanced. For a creative personal innovator, the self is the instrument of creation; just as a musician must tune their violin, a personal innovator must keep their mind, body, and spirit in tune. The NEWS framework ensures that you allocate time to sharpen the saw (to use Covey's term) in multiple dimensions: time management itself (North), mental/spiritual wellness (East), emotional/reflection (West), and physical health (South). By systematically addressing these, you reduce the mental burden of coping with day-to-day chaos and enhance your capacity to perform and create. This is a highly individual section of the system that must be tailored to each and every person's personal needs. Hence, we will not go through the contents in detail, except for North whose contents are more generic.

North signifies *True North*. Your North Star, guiding direction each day. In practical terms, this subcategory is about time planning, daily routines, and the practice of time-blocking recurring tasks. It covers the meta-level of time management: how you organise each day's schedule and ensure recurring duties are handled smoothly. Conceptually, North is crucial because it creates the *scaffolding* for everything else. It is where you implement the decisions of your weekly planning on a daily basis. Key components of North include:

- *A Dayframe* (daily framework/routine): This could mean having set times for certain activities (morning routine, lunch break, and evening routine). For example, you might establish that you wake up at 7 AM, spend 7–8 AM on personal morning rituals (like East and South activities such as meditation and exercise), start work at 9 AM, etc. The specifics vary per person, but the idea is to have a general template for weekdays (and maybe a different one for weekends) so that not every day is a blank slate. As discussed earlier, having regular habits can free mental energy. And if you want to start a new habit, include it into the dayframe at least until it becomes automatic. Modern science concurs that routines reduce decision fatigue, making it easier to focus on important decisions when they arise. Many creators have cherished routines: Haruki Murakami, for instance, described his regimen of rising early to write and running or swimming every afternoon, noting that the repetition itself helped him reach a deeper creative state.
- *Daily Planning Sessions*: Under North, you may include a short daily planning ritual. Perhaps each evening you preview the next day, or each morning you set priorities for the day. This might only take 10 minutes, but it is a time to orient to your *true north* (your key planning goals) and make sure the day's plan aligns with them. This practice can drastically cut down on time wasted in the day due to indecision or forgetfulness. By checking your to-do list and calendar, you mentally prepare and you can catch any conflicts or unrealistic expectations (e.g., noticing you planned 10 hours of tasks for a day where you only have 6 hours of actual working time available). Adjusting in this planning moment prevents stress later.

- *Time-Blocking and Recurring Task Checklist*: This refers to scheduling specific blocks for activities and systematically handling repetitive tasks. Recurring tasks are things like daily email processing, weekly reports, daily tidy-ups, taking medication, etc. By creating a *checklist* of such tasks or automatically scheduling them, you ensure they aren't overlooked and you minimise context switching. Essentially, North deals with the maintenance of your personal productivity system. Some people use techniques like the two-do (two slots per day for deep work) or a concept of a daily highlight (pick one top priority each day) to anchor their day. Such micro-strategies fit under North.

East and West are very individual, and can contain anything that caters to your taste. However, just remind yourself that devoting too much time to endlessly reiterating past event or regretting decisions seldom contribute to either happiness or success. Since we leave East, West, and South to be structured and populated as required by each individual personal innovator, this hereby ends the chapter and we move on to career paths before ending with decision-making, another cornerstone in the personal innovator's toolbox. But before we leave the topic of time management altogether, here is a pro tip that often surprises people. It relates to the overall efficiency of a personal innovator's time use, especially when handling mid-sized tasks that demand some creativity and have firm deadlines. Examples include preparing a one-hour presentation or lecture, writing a medium-length report that involves some research, or planning a larger event. What these tasks have in common is that they require creativity and a new arrangement of information. In such cases, the tip is controlled procrastination. This might sound counterintuitive at first, but the keyword is *controlled*. It involves two steps. First, you do the foundational research and broad planning well in advance. Then you let the material settle in your subconscious mind for a few days, perhaps a week, while doing other things. The second phase happens closer to the deadline, when the approaching pressure pushes you to complete the task with high focus. But this is not stressful, because the groundwork has already been laid. And as if almost by magic, the talk or report flows from your mind to the screen (or paper) with clarity, structure, and even a few unexpected gold nuggets. This requires some training and must of course be practised with care.

8. Career Paths

In the modern knowledge economy, creativity and productivity flourish when individuals are working in roles well-suited to their personality. Extensive research on person-job fit shows that aligning one's job with personal traits and preferences leads to significantly higher job satisfaction, better well-being, and superior performance. Conversely, pursuing a career that is fundamentally misaligned with one's personality can result in chronic stress, low engagement, and wasted potential. In other words, *one should never pursue a career misaligned with one's personality*, as the mismatch inevitably breeds unhappiness and inefficiency. A highly creative but free-spirited individual, for example, may feel stifled and unproductive in a rigid bureaucratic role; similarly, a methodical, detail-oriented person will struggle in a chaotic, unpredictable work environment. Over time, such misalignments not only hamper professional performance but also erode personal well-being. Even a great organisation cannot compensate for a poor person-job fit, as day-to-day work will remain a strain when the role conflicts with one's core disposition.

This chapter explores how personal innovators can select or shape their jobs to fit their personality traits, thereby maximising both creativity and efficiency. We begin with an overview of the Myers-Briggs Type Indicator (MBTI), an early and still popular personality typing system rooted in Carl Jung's theories. The MBTI gained widespread use in career counselling and organisational settings throughout the 20th century, but we will see that it lacks scientific validity despite its enduring popularity. We then turn to the traits in the Big Five model, the framework that personality psychologists overwhelmingly consider to be scientifically sound. However, rather than focusing on all five of the classical dimensions, we will deliberately omit *Neuroticism* (the tendency toward emotional instability) from our discussion. While Neuroticism is an important dimension in psychology, it pertains more to emotional health and stress reactivity than to day-to-day work style. Thus, for the purpose of aligning careers with personal work tendencies, it has little relevance and we set it aside. In its place, we introduce a new fifth dimension, the X factor (or aXess), which we define as the speed of memory access or information retrieval in the brain. This *aXess* factor captures how quickly an individual can recall relevant ideas or facts on the spot, as opposed to only being able to retrieve them after reflection. This novel trait has profound implications for creative knowledge work: some people think and respond in real-time with lightning speed, while others generate their best insights only in solitude or given sufficient time. By adding aXess to the four core traditional traits, *Openness*, *Conscientiousness*, *Extroversion*, and *Agreeableness*, we arrive at an updated OCEAX model for understanding how personality drives creativity and efficiency on the job.

In the sections that follow, we will examine each of these five dimensions in turn. For each trait, we will discuss its definition and psychological significance, outline how different levels of the trait (high vs. low) influence an individual's creative style and work performance, and illustrate what kinds of jobs or work environments are most suitable for those trait levels. The goal is to provide an academic yet practical analysis of why *aligning career choices with personality traits* is important for personal innovators. Throughout, we emphasise that no trait is inherently good or bad. Each has its strengths and potential downsides, and the key is finding the right niche where one's natural disposition can be an asset rather than a liability. By understanding the OCEAX model in depth, readers will be better equipped to reflect on their own personalities and make informed career choices that unlock their highest creativity, productivity, and professional fulfilment.

The Myers-Briggs Type Indicator

Any discussion of personality and career selection must acknowledge the outsized influence of the Myers-Briggs Type Indicator (MBTI), one of the most famous personality classification systems in popular culture. The MBTI was inspired by the work of Swiss psychiatrist Carl Gustav Jung, whose 1921 book *Psychologische Typen* introduced the idea of fundamental psychological preferences (such as Introversion vs. Extroversion and Thinking vs. Feeling). In the 1920s, American writer Katharine Cook Briggs became fascinated by Jung's theories. In 1923, she read the English translation of Jung's *Psychological Types* and felt that she had found a definitive framework for understanding personality differences. Together with her daughter, Isabel Briggs Myers, Katharine Briggs began developing a practical instrument to implement Jung's ideas. This mother-daughter team extrapolated Jung's abstract typology into a set of questions and a scoring system that could categorise individuals into 16 distinct personality types. Their efforts eventually resulted in the first version of the Myers-Briggs Type Indicator during the early 1940s. By 1962, the MBTI was adopted for broader use by the Educational Testing Service (which also administers TOEFL and GRE), and over subsequent decades it achieved remarkable commercial success and popularity. It is estimated that tens of millions of people have taken the MBTI over the years, and it remains a common fixture in career counselling, corporate team-building workshops, and online personality quizzes.

From a scientific standpoint, however, MBTI has serious shortcomings. The instrument sorts people into dichotomous categories (e.g. Introvert (I) or Extrovert (E), Thinking (T) or Feeling (F)) and yields a four-letter type such as INFP or ESTJ. While these types are memorable and appealing, the test's scientific validity is highly questionable. Psychometric evaluations have found significant deficiencies in the MBTI, including poor validity (it often fails to measure what it purports to measure) and poor reliability (people's type results often change on re-testing). Essentially, many psychologists consider the MBTI a pseudoscientific instrument. It gives users flattering, horoscope-like feedback that feels insightful due to the Barnum effect and confirmation bias, but it does not reliably predict real-world outcomes. This can actually cause a lot of long-term harm. For example, one major criticism is the MBTI's lack of predictive validity in workplace settings: an individual's four-letter MBTI type does not consistently correlate with job performance, job satisfaction, or other key outcomes. Studies have shown that MBTI classifications fail to predict how well someone will do in a given career or whether they will be happy in that role. Yet, it is even to this day used by some employers to fit job applicants into roles. Another bug problem is that the MBTI assumes bimodal distributions. It forces people into either/or categories, whereas in reality traits like introversion and extroversion are continuous dimensions. This oversimplification means the MBTI loses important nuance by labelling someone an extrovert even if they are only mildly so or ignoring differences between two people of the same type. Furthermore, the theoretical foundation of the MBTI is not derived from rigorous data but from Jung's early 20th-century ideas and the personal observations of

Briggs and Myers. Jung's typology itself was based on clinical impressions rather than quantitative research, and it was never intended as a strict test for the general population. The MBTI's creators did admirable work in trying to make Jung's ideas accessible, but they lacked formal training in psychometrics, and the resulting tool reflects that limitation.

Despite these scientific critiques, MBTI's influence persists. Part of its appeal lies in its positive, non-judgmental tone. All 16 types are described in generally flattering terms, which makes it easy for individuals to accept and even celebrate their reported type. It also offers simple answers (you are type XYZT) in a realm where people naturally seek self-understanding. However, given the weak empirical support for the MBTI, psychologists and career advisors are increasingly cautious about using it as a basis for important decisions. The MBTI can be fun and may serve as a starting point for self-reflection, but relying on it to choose a career path is problematic. Over-emphasising MBTI types might lead someone to pursue a career that *seems* matched to their type profile but is not actually a good fit at an individual level. For instance, an MBTI profile might suggest one is suited to be a librarian versus a lawyer, but these recommendations may overlook critical nuances like personal interests, skills, and the continuous nature of traits. Ultimately, while the MBTI popularised the notion that personality matters for career choice (an important insight indeed), it should be viewed with scepticism in terms of scientific accuracy. Modern personality research offers more robust frameworks for understanding individual differences, chief among them the Big Five model which we turn to next.

The Big Five

In contemporary psychology, the prevailing model for understanding personality is the five-factor model, commonly known as the Big Five. Unlike the MBTI, which sorts people into predefined types, the Big Five describes personality in terms of five broad dimensions that each person possesses to a greater or lesser degree. These five traits are often remembered by the acronym OCEAN: Openness (to Experience), Conscientiousness, Extroversion, Agreeableness, and Neuroticism. The Big Five model emerged from decades of empirical research, notably lexical analyses of trait-descriptive words in language and large-scale factor analyses of survey data, rather than from the speculations of a single mind. By the 1980s, multiple research teams converged on the finding that five major dimensions are needed to capture the most important factors in human personality. Since then, the Big Five framework has been extensively validated across different populations and cultures. In fact, numerous studies have demonstrated its reliability and validity worldwide, to the point that it is considered the *gold standard* in personality psychology. The five dimensions consistently emerge in factor analyses of personality data, show fairly high heritability, and remain relatively stable in adulthood (with some gradual changes and aside from the effects of major life trauma). Moreover, measuring people on these five scales provides meaningful predictions of various life outcomes.

One huge advantage of the Big Five is that traits are measured on continuous scales rather than binary categories. This acknowledges that, for example, there is a wide spectrum from extreme introversion to extreme extroversion, with many people falling in between. As a result, the Big Five can provide a more nuanced and accurate personality profile than a categorical type system. A person might score, say, *65/100 on Extroversion* (somewhat extroverted) and *20/100 on Neuroticism* (low in neurotic tendencies), etc., giving a multifaceted picture. Another strength is the Big Five's strong empirical backing in predicting outcomes: scores on these traits have been linked to job performance, academic achievement, interpersonal behaviour, and more. For instance, psychologists have found that highly Conscientious individuals tend to perform better in many jobs (especially those requiring organisation and dependability), while high Openness is associated with creativity and adaptability in the workplace. Indeed, meta-analyses have confirmed that some Big Five traits (especially Conscientiousness and low Neuroticism) correlate with overall job performance across occupations, supporting the use of Big Five measures in personnel selection and career guidance. A classical meta-analysis by Murray Barrick and Michael Mount in 1991 showed that among the Big Five, Conscientiousness is the most consistent predictor of job performance across different occupations, and traits like Extroversion and Agreeableness can predict success in specific roles (such as sales or customer service). Unlike the MBTI, which has scant evidence of predicting work outcomes, the Big Five model's predictive validity has been documented in numerous peer-reviewed studies.

Given these strengths, the Big Five provides a far more solid foundation for thinking about *personality–career alignment*. Rather than classifying people into a limited number of types, it allows us to consider *how much* of each relevant trait a person has, and how that pattern might suit certain work environments. For example, instead of asking Are you an introvert or an extrovert? (which the MBTI would do), the Big Five approach asks How extroverted are you, and in what ways does that level of extroversion manifest in your work preferences? This quantitative and individualised approach is better suited to fine-tuning career choices.

For the purposes of this chapter, we will focus on a modified version of the Big Five: O, C, E, A, and X (OCEAX). As noted, we are excluding the fifth standard trait, Neuroticism (sometimes relabelled *Emotional Stability* when inverted), from detailed consideration. Neuroticism, the tendency to experience negative emotions like anxiety, moodiness, and frustration, certainly can affect work life (for instance, higher Neuroticism often predicts lower job satisfaction and greater stress). However, it is somewhat distinct from the other traits in that it reflects emotional reactivity and stability, which border on mental health. Our focus here is on aligning jobs with one's work-relevant personality *style* and strengths. Therefore, we will not delve into Neuroticism, and instead propose an alternative trait to consider: what we call the X factor or aXess.

The introduction of factor X (aXess) as a fifth dimension is an attempt to capture a facet of individual difference, especially pertinent to knowledge work but not explicitly covered by the traditional Big Five. The term aXess is derived from *mental access*, specifically, the speed and efficiency with which a person's brain can access information, memories, and ideas. In cognitive psychology, there is the concept of *processing speed*, defined as how quickly the brain can take in and respond to information. aXess is related, but here we tailor it to the context of idea generation and recall in the flow of work and communication. In essence, aXess measures how quickly you can think on your feet. High-X individuals can rapidly retrieve facts, recall past experiences, or improvise ideas in real time, even under the pressure of an ongoing discussion or a tight deadline. Low-X individuals, by contrast, have a slower retrieval speed. Their thoughts may need more incubation, and while they can be just as insightful, their best contributions often emerge after deliberation rather than in the heat of the moment.

Why include aXess in a discussion of career alignment? Consider that different jobs and work environments place very different demands on *response speed* and on-the-spot cognition. In some roles, for example live television broadcasting, emergency management, high-pressure sales, or politics, success hinges on the ability to react swiftly and cogently. A news presenter or political debater with high aXess will much more likely excel because they can articulate a coherent, witty, or convincing response with little prep time. They effectively think out loud and thrive when improvising. On the other hand, jobs in research, writing, strategic planning, or innovation often allow (and indeed benefit from) longer periods of reflection. An academic researcher or novelist who is lower on aXess might be tongue-tied in an impromptu discussion, but given a few hours or days, they can produce brilliant analyses and creative outputs. Their strengths emerge in solitude or low-pressure environments, where they can methodically refine ideas. Many of us have experienced this difference first-hand: for instance, a low-X person in a brainstorming meeting might not speak up immediately and may only later (in the shower or during a quiet afternoon) think of the perfect idea or a solution that eluded them during the fast-paced group discussion. Or at a dinner party, the conversation has already moved on to the next subject when the low-X person comes up with a brilliant comment, but too late. This common phenomenon, coming up with the perfect retort or idea after the fact, exemplifies a low-aXess style. It is not a flaw, simply a different cognitive pace.

It is worth noting that while aXess can correlate with known traits (quite often, quick-thinking high-X people are extroverted and slow, deep-thinkers are introverted), this is not a strong correlation. Some introverts are very quick thinkers in their domain of expertise, and some extroverts might still prefer a lot of rehearsal. aXess is best seen as an independent continuum of cognitive style. Some scientific findings lend credence to the idea that people differ in social information processing speed. For example, studies have indicated that extroverts tend to perform better on certain memory and recall tasks in social contexts, possibly because they pay more attention to external cues and are more practiced in rapid social cognition. Introverts, in contrast, may not encode or retrieve social information as readily during an interaction, which could contribute to feeling blank when put on the spot. These differences suggest a physiological or attentional basis for variance in recall speed. Regardless of the underlying causes, the aXess concept captures a salient real-world observation: in challenging work situations, some people excel at improvisational thinking while others excel at deliberative thinking. Both styles can produce creative and effective results but in different settings.

Having defined our updated model (OCEAX) we will now delve into each of its components: Openness, Conscientiousness, Extroversion, Agreeableness, and aXess. For each trait, we will describe what it entails, discuss how varying levels can influence one's creative output and efficiency, and identify types of careers or work environments that align well or poorly with those levels. The running theme is that when individuals find roles that leverage their natural strengths on these trait dimensions and mitigate their weaknesses, they are far more likely to innovate, excel, and stay motivated. By contrast, a severe mismatch on any of these dimensions, for example a highly introverted, reflective person forced into a rapid-fire sales job (mismatch on Extroversion and aXess), or a highly open-minded creative working under extremely rigid routines (mismatch on Openness), can lead to frustration and underperformance. Understanding these traits thus provides a roadmap for both individuals and managers to create better alignment in the workplace.

Openness to Experience

Openness to Experience is the Big Five dimension that reflects curiosity, imagination, and a preference for novelty and variety. People high in Openness are intellectually curious, appreciative of art and beauty, sensitive to their feelings, and drawn to new ideas and unconventional perspectives. In essence, Openness captures the degree to which an individual is receptive to new experiences and willing to engage with complexity. A highly open person is often described as creative, innovative, or broad-minded, whereas a person low in Openness might be described as practical, traditional, or preferring routine. High-Openness individuals tend to enjoy learning for learning's sake, exploring unfamiliar topics, and brainstorming possibilities.

Those low in Openness are more comfortable with what is tried-and-true. They prefer familiarity and concrete facts and may be resistant to change or abstract ideas.

It is no surprise that Openness to Experience is linked to creativity. In fact, among the Big Five, Openness is the trait most consistently associated with creative achievement and creative thinking. Studies show that individuals who score high on Openness tend to exhibit greater divergent thinking (the ability to generate many novel ideas), are more likely to participate in artistic or intellectual pursuits, and often find imaginative solutions to problems. Indeed, an openness to diverse perspectives and new experiences can help people discover novel solutions to daily challenges. Empirical findings partly back this up: Openness correlates positively with performance on creativity tests and with supervisors' ratings of innovative behaviour at work. Moreover, openness has been linked not only to artistic creativity but also to scientific and entrepreneurial innovation. An open-minded scientist, for example, is more likely to question assumptions and explore unorthodox hypotheses, leading to breakthroughs, while an open entrepreneur might envision a product that breaks the mould of what exists.

In the context of knowledge work, which often demands creativity and adaptability, Openness is a particularly beneficial trait. Personal innovators high in Openness excel in roles that require brainstorming, strategic thinking, design, research, or continuous learning. They tend to be intellectually agile, enjoying complex problems that require thinking outside the box. For instance, a software developer high in Openness might relish adopting new programming languages or paradigms, an architect might boldly incorporate unconventional design elements, and a marketing professional might devise an imaginative campaign that challenges industry norms. These individuals feed on variety. If their job becomes too routine or static, they are likely to feel bored or stifled. Indeed, for a highly open person, a job that lacks creative scope can be deeply unsatisfying. Consider a free-thinking graphic designer stuck in a role that only permits following strict brand templates, or an academic researcher forced to adhere to a narrow, prescriptive methodology without room for exploration. Such constraints clash with the open individual's core drive to explore and create, leading to frustration and a sense of wasted potential. As a rule, high-Openness people should seek careers or projects that allow for exploration, novelty, and intellectual challenge, environments where creativity is rewarded rather than suppressed.

By contrast, individuals low in Openness thrive in more structured, stable, and detail-oriented roles. They tend to excel at implementing and following procedures, maintaining quality and consistency, and managing concrete tasks. Rather than re-inventing the wheel, they often prefer to optimise or execute established processes. For example, a person lower in Openness might be very effective in roles like quality control, project administration, or accounting, fields where adherence to standard practices and focus on factual details are paramount. They bring reliability and an eye for practical solutions drawn from experience. In team settings, less open individuals can serve as a balancing force to highly open colleagues by questioning overly fanciful ideas and ensuring that proposals are realistic and detail-checked. However, if a very conventional, practical person is pushed into a highly uncertain, rapidly changing creative environment, they may feel anxious or out of their depth. Imagine an individual who prefers clear instructions and proven methods being hired by a start-up where their role is to disrupt and continuously pivot; the lack of clear direction and the premium on constant innovation could be stressful for them. Such a person might long for guidelines and find the creative chaos inefficient or even frightening.

Yet the story does not end there. At Openlab Stockholm, where we run a semester-long Master’s course in Design Thinking and societal innovation, we have had the opportunity to observe this dynamic up close, semester after semester. In this course, each student’s cognitive style is assessed at the beginning along the dimensions of divergent and convergent thinking, proxies for the level of the broader Openness trait. Students get to know their own profile but not that of their teammates, and they work intensively in multidisciplinary teams of six to nine members for 20 weeks. What unfolds over the semester-long course is a living laboratory of innovation in action, where personalities, thinking styles, and real-world challenges intersect. The findings are both revealing and reassuring. While divergent thinkers, who often score higher on Openness, do indeed shine in some stages of the design process, especially during Empathise and Ideation, convergent thinkers are no less vital. They come into their own in the more convergent phases: Define, Prototype and Test, i.e. when defining/reframing the point of view, shaping interactive prototypes, and critically testing ideas. These stages demand focus, precision, and the ability to make decisions under constraint while participating in a wildly creative and sometimes unstructured process, areas where convergent thinkers naturally excel. In other words, both ends of the Openness spectrum contribute meaningfully to innovation, albeit in somewhat different ways.

This real-world observation challenges the common assumption that creative success is the exclusive domain of the highly open, wildly imaginative individual. It turns out that innovation often is teamwork, even for the personal innovator. What matters is not that every individual is generative and exploratory, but that the team collectively possesses a balanced capacity to diverge and converge. Those who prefer structure, clarity, and detail are not impediments to creativity; they are the essential stewards who turn abstract ideas into actionable plans, who ground their visions in feasibility, and who see possibilities through to implementation. Our key observation is not about Openness as a trait, but rather about the *willingness and ability to participate* in a creative process with the abilities one possesses. In short, there is no compelling reason for not being able to become a personal innovator regardless of Openness score.

The key insight is that innovation processes, particularly those rooted in Design Thinking, are multi-phased and cognitively diverse by nature. There is room, and indeed, need, for all thinking styles. Those who score lower on Openness are not disqualified from innovation. On the contrary, they are often the ones who bring a concept to life, refine it, stress-test it, and ensure it works in the messy, constraint-filled world. So while the headlines might go to the blue-sky thinkers, the breakthroughs often depend just as much on those who do the scaffolding, challenge ungrounded assumptions, and stay with the problem until it is solved.

In short, innovation is not a personality type. It is a choreography of minds with different rhythms. And when the full range of cognitive styles is respected and deliberately integrated, the dance becomes not only more balanced but far more powerful. It is also important to note that *Openness exists on a continuum*. Moderate levels of Openness can also be advantageous, depending on the job. Not every role requires sky-high creativity. In many cases, moderate Openness (willingness to consider new ideas, but also an appreciation for existing knowledge) is ideal. It allows one to adapt when needed, but also to work within known systems. For creativity-centric careers, however, very high Openness is often a distinguishing feature of top performers. Creative professionals (artists, writers, innovators) and visionary leaders tend to be higher in Openness than the general population. The psychologist Gregory Feist's meta-analytic research on creative personalities found that one of the largest differences between highly creative people and others was indeed Openness: creative individuals are significantly more open to new experiences. Feist also found that such individuals tend to be less conventional. This hints at a potential downside: extremely open people may become easily bored with routine work, may struggle with structure, or may resist standardisation even when it is beneficial. They might also appear unconventional or non-conforming, which can occasionally lead to friction in very traditional organisations.

For a personal innovator assessing their own Openness, key questions include: *Do I thrive on new ideas, art, or conceptual discussions? Do I enjoy learning about topics outside my expertise? Do I feel constrained by routines?* An affirmative answer suggests high Openness, and thus a career with ample room for creativity (such as research science, product design, writing, consulting, or entrepreneurship) could be a good fit. On the other hand: *Do I prefer tried-and-tested methods? Do I value expertise in a narrow area rather than being a generalist? Am I uncomfortable when procedures aren't clear?* Those leanings suggest lower Openness, indicating that roles with clear workflows (such as operations management, technical support, or finance) might be more comfortable. Importantly, neither end of the spectrum is better, they are simply different. Innovative industries absolutely need the imaginative spark of high-Openness minds, but they also need the grounded execution and risk awareness of more conventional minds. A balance often yields the best results. The key for individuals is not to force themselves into a mould that doesn't fit. A highly open, imaginative person stuck in a monotonous administrative job is likely squandering creative talent and will feel psychologically unfulfilled; a cautious, detail-oriented person thrust into a tumultuous creative role may constantly feel overwhelmed and inadequate.

So Openness (to Experience) is a trait that underpins some aspects of creative thinking and adaptability. Its presence in the OCEAX model highlights the importance of matching one's creativity needs with one's occupation. For maximum creative output and efficiency, those high in Openness should seek environments rich in novelty and flexibility, while those lower in Openness should ensure their roles provide structure and utilise their strength in applying practical knowledge. Misalignment on this trait (a creative spirit in a constraining job or a traditionalist in an unpredictable job) is a classic recipe for dissatisfaction and underperformance. Aligning Openness with the right career path unlocks passion and innovation, fuelling both personal and organisational success.

Conscientiousness

Conscientiousness is the Big Five trait associated with orderliness, self-discipline, responsibility, and diligence. It reflects how goal-directed, reliable, and hardworking a person is. High-Conscientiousness individuals are often organised, detail-oriented, and good at delaying gratification. They will persist in a task until it is finished, follow rules and deadlines, and tend to plan ahead. In a work context, someone high in Conscientiousness is the employee who keeps meticulous to-do lists, arrives on time (or early), double-checks their work for errors, and can be trusted to meet their commitments. Those low in Conscientiousness are more spontaneous, less structured, and often more comfortable with improvisation and multi-tasking (or sometimes with procrastination). They might struggle with organising their time or maintaining consistent work habits, but on the flip side, they can be quite adaptable and comfortable with changing priorities or loosely structured tasks.

When it comes to job performance, Conscientiousness stands out as the single strongest personality predictor of overall performance across a wide range of occupations. Decades of research including large meta-analyses have affirmed that, on average, highly conscientious employees tend to receive higher performance evaluations, earn higher grades in academic settings, and exhibit greater career success in terms of income and advancement. This makes intuitive sense: Conscientiousness encompasses traits like reliability, efficiency, and grit, qualities that are valuable in virtually any job. A conscientious engineer will thoroughly test their code; a conscientious nurse will meticulously follow procedures; a conscientious manager will be diligent in executing plans and monitoring progress. These habits generally lead to higher-quality work and fewer critical mistakes. In roles where attention to detail and consistency are paramount (e.g. accounting, law enforcement, surgery, and data analysis), high Conscientiousness is almost a prerequisite for success. It contributes to what we might call personal efficiency, the ability to effectively organise one's workload and steadily work through tasks without needing external prodding.

However, Conscientiousness isn't only about raw efficiency; it also has implications for creativity and flexibility. Interestingly, while a moderate to high level of Conscientiousness supports the implementation of creative ideas (by providing discipline to carry them out), extremely high Conscientiousness can sometimes clash with creativity. Creative endeavours often require a willingness to break rules, tolerate disorder during the incubation of ideas, or pursue a sudden inspiration even if it disrupts the schedule. Highly conscientious people, with their strong preference for order and planned action, may find it psychologically difficult to take the kinds of risks or divergent leaps that breakthrough creativity sometimes demands. In fact, research on creative personalities has found that eminently creative individuals (especially in artistic fields) tend to be less conscientious than average. They can be more impulsive and nonconforming. This is not to say that all creative people are disorganised. Certainly, many successful scientists and writers are both creative and conscientious, but rather often there is a trade-off. As one meta-analysis summarised, a bit stereotyped, creative people are less conventional and less conscientious than others, likely because extreme orderliness might inhibit the free flow of ideas. We can think of Conscientiousness as providing *structure*, whereas creativity sometimes thrives in *unstructured, chaotic* environments where rules can be bent.

For personal innovators, the optimal level of Conscientiousness may depend on the nature of their work. In highly creative or innovative roles, a very high-Conscientiousness individual might need to consciously loosen up at times to allow creative sparks to fly. For example, a product designer with strict work routines might benefit from occasionally embracing a bit of messiness or spontaneity (e.g. a hackathon or a free-form brainstorming session) to spur creativity. Conversely, an extremely unconscientious creative person (brimming with ideas but poor at execution) might struggle to turn any of those ideas into reality. They may constantly start projects but never finish. Such individuals might need external support or systems to help with follow-through, or they should consider teaming up with conscientious collaborators who can implement plans. Complementary partnerships are common in creative industries: the visionary idea-generator paired with the detail-oriented organiser. This underscores that neither trait extreme is sufficient alone; bringing an idea to fruition often requires the imaginative looseness of lower Conscientiousness *and* the disciplined structure of high Conscientiousness at different stages.

In more routine or execution-focused knowledge work (like project management, quality assurance, or administration), high Conscientiousness is almost unequivocally a strength. A conscientious project manager will keep the team on schedule, document progress, and adhere to standards, thereby increasing efficiency. A low-Conscientiousness person in that role might let details slip through the cracks, leading to costly errors or delays. It is telling that employers often prioritise Conscientiousness when hiring; traits like dependability and work ethic are highly valued because they reliably contribute to performance. For the individual, being well-organised and disciplined typically translates to personal productivity and success in meeting goals.

The potential downsides of extremely high Conscientiousness in the workplace include inflexibility and perfectionism. A very conscientious worker may struggle to adapt when plans change suddenly; they might also spend too much time perfecting minor aspects of a task (diminishing returns) or be overly critical of others who are less fastidious. For instance, in a fast-moving tech start-up, a perfectionist who insists on flawless procedures might slow down the team's ability to iterate quickly. In contrast, the potential downsides of very low Conscientiousness are missed deadlines, disorganisation, and unreliability, clearly problematic in most professional settings. Thus, while in general one cannot be too conscientious from an employer's perspective, there is a balance to be struck between discipline and flexibility.

From a career alignment standpoint, an individual should consider their natural level of Conscientiousness and how that meshes with the demands of a job. High-Conscientiousness individuals (those who answered yes to questions like Do I work ahead of deadlines? Do I keep my workspace and plans well-organised? Do I feel discomfort when things are out of order?) will do well in roles that reward meticulousness and self-discipline. These could range from scientist or researcher (where careful methodical work is needed to validate results) to operations manager (keeping complex processes running smoothly) to any role that involves significant planning and responsibility (such as event coordinator, executive assistant, or military officer). They may also prefer work environments with clear expectations, well-defined workflows, and a culture of reliability. If they find themselves in a very lax or chaotic workplace, they might become frustrated that others are not meeting the same standards or that projects lack structure.

Low-Conscientiousness individuals (who might relate to statements like “I often procrastinate or switch between tasks impulsively” or “I dislike rigid schedules and prefer to keep things casual”) should be careful about taking roles that require heavy organisation and consistency, as these may play to their weaknesses and cause stress. However, they might excel in jobs that are dynamic, spontaneous, or entrepreneurial in spirit. For example, roles that require quick pivoting between tasks, handling unpredictable situations, or creative improvisation. A journalist covering breaking news, for instance, might not need extreme conscientious planning since the situations evolve rapidly; adaptability and quick thinking (more tied to aXess and Extroversion) are key. Similarly, a creative director who jumps from brainstorming to client meetings to design tweaks might thrive on a bit of disorder and would chafe at having every hour scheduled. If a person knows they are not naturally organised, they may thrive in such fluid environments or in roles where they can rely on support staff to handle detail management. Alternatively, they can develop coping strategies to mitigate their weakness (using external tools or strict routines to compensate for a lack of internal inclination).

For creativity, an optimal mix might be a mid-high Conscientiousness to ensure execution, combined with high Openness for idea generation. An extremely conscientious person can be creative too, especially in fields like architecture or engineering, where precision and creativity must go hand in hand. But if a conscientious individual feels their creativity is stifled, they might consider deliberately incorporating more flexibility in their work or collaborating with more spontaneous colleagues. On the other hand, an extremely spontaneous person with many ideas might partner with or hire a detail-focused manager to help structure their efforts.

Conscientiousness is a trait of diligence and reliability, critical for efficiency in any job, but with nuanced effects on creativity. High Conscientiousness generally predicts better job performance and is a boon in roles requiring organisation and consistency. However, too much rigidity can impede innovative processes, and too little conscientiousness can lead to disorganisation and inefficiency. For career alignment, one should honestly assess their own place on this spectrum. A naturally methodical person will shine in a role that values planning and thoroughness (and should be cautious of roles where messiness reigns), whereas a person who is more impulsive and flexible may seek out roles that capitalise on quick responsiveness and variety (and should be cautious of jobs that demand intense routine and precision). Aligning Conscientiousness with job demands helps ensure that one's work style enhances rather than hinders productivity and creative output.

Extroversion

Extroversion is the Big Five dimension that describes where people draw their energy and how they engage with social environments. It encompasses traits such as sociability, assertiveness, talkativeness, and enthusiasm. Highly extroverted individuals are gregarious, enjoy being around others, and often thrive in high-stimulation settings. They are energised by conversations, team activities, and public events. They tend to speak up in meetings, network with ease and may be viewed as natural leaders or spokespeople due to their outgoing nature. In contrast, those low in Extroversion (i.e., introverts) are more reserved, often preferring solitary activities or interactions with a small circle of close colleagues/friends. Introverts are energised by quiet reflection and may find excessive social interaction draining, even if they have good social skills. It is important to note that Introversion is not shyness or antisocial behaviour. Rather, it is a preference for lower-stimulation environments. Introverts typically excel at listening and focused, independent work, whereas extroverts excel at quickly connecting with others and multi-tasking in busy environments.

In the workplace, Extroversion can significantly influence the kinds of roles in which people feel comfortable and excel. Highly extroverted individuals often gravitate towards (and succeed in) jobs involving extensive interpersonal contact. Classic examples include sales, marketing, public relations, politics, teaching, consulting, and management. In such roles, being outgoing and assertive is a direct asset: an extroverted sales manager will enthusiastically network with clients and motivate their team with high energy; an extroverted teacher keeps a lively classroom and isn't drained by the constant social engagement it entails. Indeed, research has found that Extroversion is a positive predictor of performance in roles like sales, where engaging with customers and persuasive communication are key. Extroverts also tend to *emerge as leaders* in group settings; their assertiveness and talkativeness often put them in the centre, and others may gravitate toward their confident demeanour. A well-known meta-analysis on leadership found that Extroversion is one of the strongest personality correlates of leadership emergence and effectiveness (though not the only important trait). Extroverted managers are often comfortable making quick decisions on the fly and handling multiple interactions (with team members, stakeholders, etc.) throughout the day.

From a creativity and efficiency perspective, extroverts and introverts bring different strengths. Extroverts, with their affinity for stimulation, may excel in collaborative creativity. They enjoy brainstorming sessions, bouncing ideas off others, and might be skilled at thinking aloud. Because they process externally, they can generate and refine ideas through discussion. In group problem-solving, extroverts often keep the momentum going, voicing suggestions and building on others' inputs. They also tend to handle interruptions or multitasking better, since switching gears to talk to someone may even invigorate them. This can be efficient in fast-paced work environments where one must respond to constant communication. Extroverts often have a positivity and enthusiasm that can be contagious, lifting team morale and thus indirectly boosting creative confidence in a group. Their higher baseline of positive affect is a known correlate of extroversion, which can broaden people's thought-action repertoires according to the broaden-and-build theory.

Introverts, on the other hand, shine in individual creative work and deep focus. They are more likely to prefer working in a quiet setting where they can concentrate without constant social stimulation. This can lead to high-quality creative outputs, particularly for tasks requiring complex solitary thinking, such as writing, programming, design, or research analysis. An introverted writer might spend hours absorbed in crafting a novel or report, a task that would exhaust an extrovert who hasn't had social interaction in that time. Moreover, introverts typically excel at listening and observation, which are undervalued components of creativity and problem-solving. They may catch subtle details in discussions or research that talkative extroverts miss. When introverts do speak, it is often after thoughtful reflection, and thus they may bring well-considered ideas to the table. In leadership, while extroverts often take charge, introverts can be effective leaders too by leveraging their listening skills and thoughtfulness. Studies have actually shown that introverted leaders can outperform extroverted leaders when managing proactive teams; introverts are more likely to listen to and implement employees' ideas rather than dominate the exchange. This means in settings where team members are initiative-taking and creative themselves, an introverted leader can foster an environment for others to shine, whereas a strong extrovert leader might inadvertently overshadow or direct the contributions.

In terms of career alignment, Extroverts should seek roles that provide ample social interaction, variety, and opportunities to influence or entertain others. They tend to be unhappy in isolated roles with minimal human contact. An extremely extroverted person, for example, would likely find a solitary data entry job or a research lab position (working alone all day with a computer) to be draining and monotonous, regardless of intellectual fit. They would miss the stimulation of teamwork or client engagement. Such a person would be better off in a role where interaction is frequent. For example, a project coordinator who constantly communicates with different departments or a customer success manager engaging with clients. Extroverts also often prefer a quicker pace and may grow impatient in environments where decisions or actions move slowly. They enjoy being in the mix of action.

Introverts, conversely, should pay attention to the social demands of a job. An introvert can absolutely function in social jobs (many are excellent professors, negotiators, or even actors), but they need to ensure they have sufficient downtime or control over the social intensity. If a job mandates continuous networking, large meetings all day, or rapid-fire interpersonal tasks, a strongly introverted person may experience burnout or chronic exhaustion. They might excel for a while by pushing out of their comfort zone, but in the long run, it could degrade their performance and well-being. Careers that allow significant independent work or one-on-one interactions (as opposed to large group settings) are often more suitable for introverts. Examples might include research scientist, writer, software developer, or any role where one can work behind the scenes for portions of the day. Introverts typically prefer depth over breadth in relationships, so they might enjoy roles where they form a few deep client relationships rather than many superficial ones.

Both extroverts and introverts can be highly creative, but the nature of their creativity might differ. Extroverted creativity often happens in real time and in interaction. Think of a team workshop where ideas are flying, or an improvisational brainstorming meeting. One Psychology Today writer described it as extroverts' creativity leans toward being with other people, doing things out in the world, fuelled by their social energy. Introverted creativity, in contrast, often flourishes in quietude, the proverbial lone innovator in the garage or the novelist at her desk. As Susan Cain argued in *Quiet: The Power of Introverts*, solitude can be a catalyst for creativity for introverts, as it provides the freedom to explore thoughts deeply without distraction. Both kinds of creativity are valuable. Many creative industries now recognise that brainstorming sessions might not tap everyone's best ideas; some people generate better ideas independently. Thus, workplaces could benefit from hybrid approaches (allowing private ideation time followed by group discussion) to accommodate both extroverted and introverted styles.

One interesting intersection is with the aXess (X) factor introduced earlier. Often, extroverts are quick to respond in social situations. They might be high on aXess, firing off ideas rapidly in a meeting. Introverts sometimes take longer to respond because they process internally, suggesting lower aXess in group settings. As one commentator humorously noted, extroverts think quickly on their feet and speak their thoughts out loud, whereas introverts process things so thoroughly internally that by the time they're ready to speak, the conversation may have moved on. This can make extroverts seem more immediately creative or contributory in group forums, even if introverts have equally valuable ideas that simply emerge later. Both individuals and managers need to recognise this dynamic so that introverts aren't unfairly labelled as uncreative or disengaged just because their creative process is less visible or immediate. Allowing different modalities of contribution (verbal, written, asynchronous) can help ensure the introverts' ideas surface. Meanwhile, extroverts might practice patience to give quieter colleagues space to contribute and also train their own listening skills. Extroverts can sometimes dominate discussions, which might limit their ability to fully absorb others' input or reflect deeply.

In conclusion, Extroversion vs. Introversion is a critical factor in career satisfaction and effectiveness. Extroverts generally flourish in interactive, high-energy roles and can leverage their social confidence for leadership and collaborative creativity. Introverts excel in roles that require concentration, thoughtfulness, and deep expertise, and they bring a calm and deliberative presence that can lead to well-considered decisions. A mismatch (say, an extrovert confined to solitary work, or an introvert in a relentlessly social job) can lead to underperformance and strain. Aligning this trait means finding the right balance of social interaction and quiet work in one's job. Many careers can be suitable for either extroverts or introverts if flexibly structured (for instance, programming can be solitary, but an extroverted programmer might thrive in a team programming environment or as a tech lead engaging with multiple stakeholders). The key is self-awareness: understanding how one's social energy operates and ensuring one's work life has the appropriate rhythm of engagement and solitude. In the end, teams benefit from both types: the extroverts often propel action and maintain external connections, while the introverts ensure depth and reflection. When each person is in a role that suits their position on this spectrum, the whole organisation's creativity and efficiency benefit.

Agreeableness

Agreeableness is the Big Five trait that captures differences in interpersonal orientation, essentially how sympathetic, cooperative, and considerate one is versus how tough-minded and competitive. High-Agreeableness individuals are typically described as *warm, friendly, empathetic, trusting,* and *generous*. They are attuned to others' needs and are inclined to help, avoid conflict, and maintain social harmony. In a team setting, an agreeable person is one who fosters collaboration, listens to colleagues' concerns, and is willing to compromise to get along. Conversely, low Agreeableness (sometimes bluntly termed *disagreeableness*) is associated with being more critical, sceptical, uncooperative, and at times antagonistic. Less agreeable people are willing to engage in conflict or competition to assert their point of view; they can be seen as tough or even abrasive, but they are often straight-shooters who don't shy away from pointing out problems.

In the workplace, the level of Agreeableness can influence one's role and effectiveness in various ways. Highly agreeable individuals tend to excel in roles that require teamwork, customer service, or nurturing and developing others. For example, in human resources or counselling, being able to build trust, show empathy, and handle interpersonal issues gently is important, strengths of an agreeable personality. A manager high in Agreeableness might be particularly good at mentoring employees, resolving conflicts smoothly, and creating a supportive team climate. Agreeableness also correlates with service orientation: studies have found that in customer service roles, more agreeable employees often receive higher performance ratings, likely due to their patience and positive interpersonal skills. Professions such as nursing, teaching, social work, or non-profit advocacy similarly benefit from a high dose of compassion and cooperative spirit; agreeable people find fulfilment in these helping-oriented careers and are often very effective in them.

Low Agreeableness individuals bring a different set of strengths. They are often more analytical or sceptical, which can be useful in roles that require tough decision-making, objective critique, or negotiation. For instance, a very agreeable person might find it psychologically difficult to fire an underperforming employee or to drive a hard bargain with a client, whereas a less agreeable (more detached) person can do so more readily when it is necessary for the business. In fields like law, auditing, or research critique, a healthy level of disagreeableness can be an asset. It allows one to question assumptions, challenge ideas openly, and not be overly swayed by a desire to please. Creative fields also sometimes reward a bit of disagreeableness: great innovators or artists are occasionally noted to be unconcerned with others' approval and willing to defy norms (a stance linked to lower Agreeableness). In fact, personality research on creativity indicates that creative achievers, especially in artistic domains, tend to be lower in Agreeableness. They are more likely to be non-conformists, even at the risk of social friction. This makes sense: producing radically novel work or ideas often involves challenging consensus and not worrying too much about offending sensibilities. A disagreeable scientist might doggedly argue for an unpopular theory (and eventually be proven right), whereas a too-agreeable one might give up in the face of colleagues' scepticism.

However, while low Agreeableness can correlate with creative boldness or critical thinking, it carries interpersonal risks. Being very low on Agreeableness might lead to difficulty maintaining collaborative relationships. Such individuals can be seen as abrasive, uncooperative, or overly competitive, which can create conflict or resentment in a team. In extreme cases, a highly disagreeable person could disrupt team cohesion or alienate clients. Thus, certain high-contact careers (like hospitality or diplomacy) would be ill-suited to them. But in jobs where independent work is key and blunt honesty is valued (say, a forensic accountant who must call out fraud, or a journalist pressing politicians with tough questions), low Agreeableness might enhance effectiveness.

For personal innovators selecting careers, consider how much of your success and satisfaction depends on smooth interpersonal relations versus independent, critical work. If you are highly agreeable, you find yourself instinctively trusting others, avoiding arguments, and striving to be helpful, you will likely thrive in cooperative environments. Roles on cross-functional teams, client relations, or any position requiring lots of interpersonal coordination will play to your strengths. You might also prefer an organisational culture that is collegial rather than cutthroat. A highly political, competitive corporate environment, for instance, might stress out an extremely agreeable person who dislikes confrontation and office politics. They had be happier in a culture known for teamwork and mutual support. Agreeable individuals also have to be mindful of not becoming pushovers; in negotiations or in advocating for resources, they might yield too quickly. If an agreeable person finds themselves in a role that demands constant hard negotiation or critical feedback (like a contract negotiator or a quality inspector who must call out errors), they may need to develop a more assertive communication style to cope or reconsider if that environment is right for them long-term.

If you are low in Agreeableness (you don't mind debate, you are comfortable with conflict, and you prioritise truth or results over people's feelings), you might excel in competitive environments or in roles where firmness is required. You probably won't shy away from making unpopular decisions. For example, entrepreneurs with low Agreeableness might make bolder moves because they worry less about everyone's approval; they can handle the stress of others disagreeing with them. Similarly, a project leader low in Agreeableness might more readily call out a team member's poor performance and enforce accountability, which can improve results (though it might hurt some feelings). You might find you have a talent for critical analysis, able to dissect ideas without sugar-coating. Research careers (where peer review can be blunt), consulting (telling clients hard truths), or management in high-performance cultures might feel suitable. However, a word of caution: if your role requires significant empathy (like healthcare or customer care), low Agreeableness could be a hindrance unless you consciously adjust your style. Additionally, networking and collaboration are important even in competitive fields. A reputation for being too disagreeable can limit career growth. Many low-agreeable individuals benefit from partnering with a more agreeable colleague who can handle relationship management, allowing them to focus on the tasks at hand. For instance, in a business partnership, one partner may handle most client relations with warmth while the other deals with tough negotiations.

In terms of creativity and knowledge generation, agreeableness can influence group creativity dynamics. Teams composed of highly agreeable members tend to have a pleasant, non-confrontational brainstorming process. The upside is that everyone feels comfortable sharing ideas. The downside could be groupthink. If people are too agreeable, they may hesitate to critique or debate ideas critically, possibly settling on a mediocre consensus to avoid hurting feelings. Introducing a more disagreeable personality into the mix can actually spur debate and lead to a more thorough evaluation of ideas, improving creative outcomes (though it may cause tension). Research supports this nuanced view: one study on team innovation found that a moderate level of task conflict (often driven by less agreeable members) can enhance creativity, provided there is enough baseline trust. The key is balance, ensuring that critical viewpoints are heard (often from less agreeable individuals) without letting interpersonal animosity take over. An agreeable team leader can play referee, encouraging constructive criticism while smoothing over personal frictions. For individual creativity, particularly in solitary endeavours like writing, agreeableness might determine style more than output. A highly agreeable writer might shy away from controversial topics, producing universally palatable work, whereas a low agreeable writer might be provocative and challenge readers. Both can be creative; their audiences and impacts differ.

To align this trait with career and work style, one should reflect on questions like: *Do I derive satisfaction from helping and cooperating, or from winning and being right? Do I avoid confrontation, or do I not mind it?* If you lean toward the former in each case, a cooperative and people-centred career track is likely to be fulfilling. If the latter, you might prefer roles where performance is measured more objectively and directly (sales quotas, research publications, etc.) rather than by peer appreciation. Many careers have room for both types: for example, in academia, highly agreeable scholars might gravitate to collaborative, interdisciplinary projects or teaching, whereas less agreeable ones might focus on solitary research and vigorously debate theories at conferences.

So Agreeableness influences how we work with others, whether we thrive on cooperation or on competition. High Agreeableness generally facilitates teamwork, customer relations, and any context where empathy and trust are vital, thus contributing to a harmonious and often efficient work environment. Low Agreeableness can be advantageous for objective analysis, innovation, and leadership in challenging situations, ensuring that difficult decisions are made and that norms can be questioned when needed. Both ends have pitfalls: too much agreeableness may lead to being overlooked or failing to advocate for necessary change, while too little may result in conflict and poor team cohesion. The optimal scenario is to find a role that matches your interpersonal style, and if necessary, to develop strategies to manage the downsides of your trait level. For example, an agreeable person in a leadership role might train in assertiveness to handle tough calls, and a disagreeable person might practice active listening and empathy to avoid alienating their team. By aligning the agreeableness trait with the right work context, personal innovators can ensure that their natural way of relating becomes an asset that drives collective creativity and efficiency, rather than a source of friction or personal stress.

The X factor

The X factor (aXess) in our OCEAX model represents a dimension of cognitive style not captured by the traditional Big Five, namely the speed and manner in which an individual accesses information and ideas from their mind. We define aXess as cognitive retrieval speed, which influences whether a person can respond to mental demands almost instantaneously or requires additional time and reflection to formulate responses. In essence, this trait differentiates the *fast thinkers* from the *slow, deep thinkers*. It is somewhat analogous to the concept of *processing speed* in cognitive psychology, but applied specifically to scenarios like problem-solving on the fly, participating in discussions, or coming up with ideas under time pressure.

High-aXess individuals are those who seem to have a mental lightning bolt at their disposal. In conversation or during a meeting, they can almost *immediately* retrieve relevant facts, recall names, data, or past incidents, and produce a coherent answer or idea. They often excel in environments that demand quick reaction or improvisation. Think of a skilled politician facing tough questions in a live debate. The high-X politician will deftly recall statistics or an apt anecdote and respond without pause. Similarly, a broadcast journalist reporting breaking news or a customer support representative handling rapid-fire inquiries would benefit from a fast retrieval brain that can juggle information in real time. These individuals thrive on immediacy; the act of thinking and responding swiftly is comfortable and even enjoyable for them. Many high-X people describe feeling in the zone when under pressure. Their cognitive engines rev up, and they achieve clarity in the moment, much like an athlete in a high-speed sport.

Low-aXess individuals, by contrast, have a slower, more reflective cognitive tempo. They might not have a ready answer to a complex question posed in a meeting; instead, they often say, Let me think about that and get back to you. It is not that they are less intelligent or knowledgeable. Frequently, they have just as much (or more) information stored in memory, but their retrieval is more deliberate. Their thoughts may come in *layers*: initial silence or uncertainty, then perhaps a tentative idea, and later on, after some incubation, a more fully formed and insightful response. Many low-X people relate to the scenario of the *staircase wit*, known in French as *l'esprit de l'escalier*. The phenomenon of thinking of the perfect reply or solution after the moment has passed. For example, an introverted team member might contribute little during a fast-paced brainstorming session but then email a brilliant suggestion that occurred to them later in the evening, once they had time to process. Such individuals often shine in contexts that allow for asynchronous communication or delayed reflection. Writing reports instead of verbal debriefs, or solving problems in solitude rather than in front of a whiteboard with an audience. To mitigate the X-factor influence, we have at Openlab Stockholm modified the Scrum project management method to include a mandatory backchannel point on every daily Scrum morning agenda, where anyone can bring up anything that has come up since the team met last, without being disrespected or scorned at.

This aXess trait is critical for career alignment because different jobs and cultures put very different emphases on immediate responsiveness. In some corporate cultures, the loudest and quickest voice in the room gets rewarded; in others, thoughtful analysis and written follow-ups are valued over instant answers. A person high in aXess will flourish in roles where being *sharp and reactive* is at a premium. Examples include: emergency medicine (ER doctors making split-second decisions), live broadcasting or podcast hosting (thinking of engaging responses on air), trading and finance (reacting in real-time to market changes), and certain types of consulting or client-facing roles (fielding unexpected questions or crises in real time). These roles can be highly stressful for someone who needs extra time to formulate thoughts, but for high-X individuals, the rapid pace is stimulating rather than stressful. They may also gravitate toward leadership positions in dynamic fields, as they are comfortable making calls on the spot. It is worth noting that many high-X individuals are extroverted, as the extrovert's tendency to think by talking aligns with quick retrieval, but this is not an absolute rule. One could be a fast-thinking introvert who processes internally at high speed, or an extrovert who speaks a lot but perhaps without much content if their retrieval is shallow. Thus, aXess adds a complementary lens to Extroversion.

Conversely, a low-X person should be mindful to find roles where *quality of thought* is prioritised over *speed of response*. In research and development, for instance, what matters is the depth and accuracy of insights, not that they were conceived in five minutes. A scientist or scholar with a reflective style might produce ground-breaking theories after months of rumination. Their comparative slowness in daily meetings is inconsequential compared to the end result of their careful thinking. Similarly, writers, strategists, or system architects often benefit from being slow cogitators; they connect the dots and synthesise information in ways that a rapid thinker might skip over. These individuals might feel most comfortable in careers that allow independent work time or scheduled meetings with prepared agendas (so they know the questions in advance and can prep). They might excel in asynchronous collaboration, for example contributing via detailed written proposals or code rather than in impromptu discussions.

It is important to emphasise that *slow does not mean inferior*. A race car and a bulldozer move at different speeds but are suited to different tasks. High-X brains are like race cars: great for quick manoeuvres, less ideal for heavy intellectual lifting that requires sustained torque. Low-X brains are like bulldozers or perhaps long-distance trains: they might start slowly, but they can carry complex loads of thought and reach profound destinations given time. There are also intermediate, or moderate aXess individuals, who can handle moderate pressure but not extreme, or who might do fine in most meetings but still appreciate some time to refine ideas.

The real-world implications of aXess mismatches are evident in many workplaces. Consider an employee who is excellent in written analysis but falters in every spontaneous meeting. If the company culture heavily favours those who speak up on the fly, this employee's contributions might be undervalued. Over time, they could become demoralised and not be seen as go-getters, when in fact they have a trove of ideas that simply aren't being heard in the format the organisation demands. The opposite scenario is also possible: a quick-witted worker who thrives in crises might feel underutilised in a slow-moving, bureaucratic job where there are no opportunities to show their rapid-fire problem-solving; they may get bored waiting for something to react to, and their tendency to jump into action could be viewed as impulsiveness in a setting that requires patience. For maximum creativity and efficiency, one should align their role with their aXess level. High-X individuals will contribute best when they can leverage their quick recall, e.g., leading discussions, handling live projects, or serving as an on-call expert who fields urgent issues. Low-X individuals will contribute best when allowed to work in their thoughtful manner, e.g., being given complex problems to solve with research, or writing the definitive analysis after gathering inputs.

To illustrate, imagine two personal innovators in a tech company: Alice and Bob. Alice has high aXess; in meetings, she is quick to suggest solutions and can debug code on the fly while people talk. Bob has lower aXess; in the same meetings, he is quiet and sometimes seems uncertain, but later he often presents a well-thought-out improvement that Alice's quick solution overlooked. An enlightened manager would see that both have value: Alice's quick thinking keeps projects moving and handles emergencies, while Bob's careful thinking optimises the system and prevents future issues. If the manager only rewards Alice (for her visible quick contributions) and ignores Bob, Bob might become disengaged or leave, and the team would lose his deeper insights. The lesson is that teams do well to appreciate both fast and slow cognitive styles. Many high-performing organisations explicitly encourage a mix: brainstorming (to harness quick ideas) coupled with follow-up reflection (to allow further ideas to emerge). Individuals, too, can adapt once they recognise their style. A slow thinker might request agenda questions in advance or follow up in writing to ensure their ideas are heard, and a fast thinker might learn to occasionally pause and double-check their rapid responses (which sometimes can be prone to error due to insufficient reflection).

Does aXess correlate with overall creativity? Not necessarily. It is more about the timing of creativity. High-X individuals might produce more *immediate* creative outputs. For example, a jazz musician improvising a brilliant solo on the spot. Low-X individuals might produce more *considered* creative outputs. Another musician might compose a beautiful piece in solitude but not be as stellar at improvisation.

Some evidence from creativity research aligns with this: historically, there have been conceptual innovators (like Mozart, who composed quickly and prodigiously early in life) versus experimental innovators (like Beethoven, who laboured over his compositions with many revisions). Economist David Galenson's study of artists classified them similarly: some innovate in quick bursts, others through slow iteration. This maps well onto the X factor. Those quick-burst innovators have minds that can marshal ideas rapidly (high-X), whereas slow iterators have the patience to let ideas mature (low-X). Each style can yield great creativity but on different timelines. When choosing a career or even a specific project, understanding which style you gravitate toward can ensure you set realistic expectations. A high-X person may excel in hackathons or sprint projects, while a low-X person may prefer long-term research or writing a comprehensive book.

In conclusion, the X factor is about cognitive tempo, and aligning this with one's work can dramatically impact both performance and comfort. High-X personal innovators should seek or shape roles that take advantage of their quick reflexes: situations requiring real-time analysis, spontaneous creativity, or rapid decision-making. Low-X personal innovators should gravitate to roles where thoughtful analysis, thoroughness, and after-the-fact insight are valued over instantaneous response. One is not better than the other; they are different modes of operation. By recognising where one falls on this spectrum, personal innovators can avoid beating themselves up for not thinking as fast as someone else, or conversely, not having the patience others might expect. Matching these preferences with the demands and culture of a job will lead to greater efficiency (because the individual can work in their optimal mode) and greater creativity (because the mind, whether fast or slow, is allowed to operate at its full capacity without undue stress).

Aligning Personality and Career

Having examined the OCEAX model (Openness, Conscientiousness, Extroversion, Agreeableness, and aXess) in detail, we can now appreciate how each trait contributes to an individual's unique pattern of strengths and predispositions in the workplace. The core insight that emerges is straightforward yet profound: when people choose careers or work environments that align with their personality traits, they unlock higher levels of creativity, productivity, and personal satisfaction, whereas misalignment causes friction, inefficiency, and unhappiness. This alignment is not about limiting oneself or putting people into rigid boxes; rather, it is about recognising the natural tendencies one has and finding or shaping roles where those tendencies can be assets.

A useful way to think about it is in terms of energy and strain. If your job is well-matched to your personality, you will likely feel energised by many of its core tasks. They will play to your strengths and feel right. Challenges at work will still exist, but they will be surmountable and often invigorating, contributing to a sense of *flow* in your activities. In contrast, if your job is a poor fit for your personality, even relatively simple tasks can feel unnecessarily arduous. You will be expending energy just to counteract your natural inclinations (for instance, an introvert forcing themselves to engage in back-to-back meetings, or a low-Conscientiousness person fighting against their habits to meet constant deadlines). Over time, this creates stress and burnout, and as studies on person-job fit have shown, it can lead to lower well-being, poor performance, and a desire to quit.

The personal innovator of today often has more flexibility and options than in past generations. Career paths are not as fixed, and many jobs even allow tailoring of roles to some extent. This offers an opportunity to consciously seek out environments that suit one's OCEAX profile. For example, a highly open and introverted person might flourish in a research-oriented tech company that values innovation and gives employees quiet time to develop projects, but they might feel stifled in a conservative firm with strict hierarchies and incessant group meetings. Or consider a low-X but highly conscientious individual: they might avoid fast-paced consulting firms (where quick turnaround is prized) and instead look for roles in project management or academia, where thoroughness and careful planning are assets even if one isn't the fastest thinker on the spot. It is also important to note that alignment does not mean total homogeneity. One can seek growth and development in a job by leveraging strengths while also gradually improving on weaker areas. However, the foundation should be solid. If a job calls almost exclusively on a person's weakest traits, that person will likely never feel competent or fulfilled in it. A classic case is the socially introverted but intellectually brilliant engineer who is forced into a sales role. A wiser approach for that person would be to remain on a technical track or find a compromise role (like sales engineer, where technical expertise is primary and social interaction is more structured).

Ultimately, the advice to never pursue a career misaligned with your personality is about honouring your authentic self in your professional life. This does not mean avoiding all challenges or refusing to learn new skills. Growth often happens just outside our comfort zones, but it does not mean forcing yourself long-term into a mould that doesn't fit. Human potential is wasted when a person's energies are consumed by compensating for a misfit, rather than building on their unique gifts. In contrast, when your role aligns with who you are, work feels more like an extension of your natural activities. You are likely to achieve a state of internal harmony and confidence that propels creative thought and effective action. Not only do you benefit personally, but your organisation and broader society benefit as well because you are contributing at your full capacity.

In a practical sense, aligning a career with personality involves continuous self-assessment and sometimes courageous choices. It may require turning down a prestigious offer that doesn't feel like you, requesting a role change, or carving out a niche in your current job that better suits your style. The effort is well worth it. A large body of organisational psychology research supports that better person-job fit leads to better performance and greater happiness on the job. By using the OCEAX model as a guide, personal innovators can make informed decisions, turning the abstract ideal of do-what-you-love into the concrete practice of do-what-aligns-with-your-personality. That alignment is a cornerstone of sustained creativity, efficiency, and career fulfilment.

Before we move on to teamwork, we conclude with the *no-quack rule*. This principle is closely linked to the earlier advice: never build a career that conflicts with your personal traits. The no-quack rule has two parts: i) do not engage in matters you do not know well enough, and ii) do not engage in matters that lack real impact. Failing either one is to *quack*. If the first condition is not met, learn before you engage. If the second is not met, move on. Following this rule helps ensure that your time and energy as a personal innovator are spent effectively and with purpose.

9. Team Dynamics

The personal innovator will interact with organisations, either daily or in other, perhaps project-based forms. In all knowledge-driven organisations, teams are the engines of innovation. While individual insight plays a critical role, complex problems, especially those requiring creativity and implementation, demand coordinated effort. Whether the task is designing a new product, rethinking a service, or solving an organisational challenge, collaboration brings together diverse minds to generate, refine, and execute ideas. However, putting individuals together does not automatically create a team. It creates a group, and the transformation of that group into a functioning team is neither instant nor always smooth. Understanding the phases of team development is therefore essential for leaders and participants alike. This is particularly true in time-bound, high-intensity contexts like Design Thinking projects, where expectations for creativity are high, and interpersonal tensions can easily derail progress if not properly managed.

Team development is not a linear or uniform process. It involves psychological, social, and procedural adjustments as members learn to work with one another. The most widely accepted framework for understanding this progression originates from psychologist Bruce Tuckman, who in 1965 proposed a model of team development in four phases: Forming, Storming, Norming, and Performing. Later, organisational psychologist Susan Wheelan expanded and validated this model in empirical studies, reinforcing its applicability across various team types. These four phases describe not only what happens in teams, but also what must happen for a team to mature. Ignoring these stages or assuming a shortcut to teamwork usually leads to disillusionment or dysfunction.

This chapter explores the four-phase model in detail, with particular attention to Design Thinking teams, which often experience more pronounced turbulence due to the nature of their creative processes. We will see that the Storming phase, the period of conflict and adjustment, is inevitable but far from infinite. The key insight is that all members must expect it, prepare for it, and understand that it will pass. In fact, the ability to weather the storm is what enables a team to reach its true collaborative potential. This awareness is especially critical in Design Thinking, where creative tension is both a feature and a risk. By exploring how each phase manifests and how Design Thinking projects move through (or fail to complete) these stages, we provide practical insights for personal innovators engaged in collaborative innovation.

Phase 1: Forming – Enthusiasm and Ambiguity

The Forming stage is where a group first comes together. In this phase, individuals are polite, cautious, and focused on understanding their place within the team. Communication is typically superficial but courteous. There is little conflict because members are not yet comfortable enough to challenge one another, and priorities are centred on orientation: who are we, what is our goal, and how do we work together? In Design Thinking teams, forming often happens rapidly, sometimes within the space of a kick-off meeting or workshop session. Members may be drawn from different functions or organisations, and roles are not always predefined. Because Design Thinking encourages interdisciplinary input, the team may be unusually diverse in terms of professional backgrounds and cognitive styles. This can be a strength, but it also means that assumptions, jargon, and work norms may differ sharply. At this stage, ambiguity is high, and most energy is spent building basic rapport and clarifying expectations. Teams in the Forming phase often exhibit overconfidence in their potential. With few tensions yet surfacing, there is a sense that collaboration will be easy and enjoyable. In reality, the difficulties lie ahead. While it is important to establish trust and shared purpose during Forming, it is equally vital to anticipate that the group will soon be tested, which brings us to the next phase.

Phase 2: Storming – Tension, Conflict, and Self-Assertion

Storming is the most volatile and important phase of team development. It is characterised by conflict, competition, and the surfacing of differences in personalities, working styles, priorities, and communication preferences. Members begin to assert their views more forcefully. Questions of leadership, authority, responsibility, and influence often come to the fore. In some teams, Storming is overt and dramatic; in others, it may be more subtle, expressed through passive resistance or disengagement. Design Thinking teams are particularly prone to intense storming, for two reasons.

First, the method deliberately invites divergent thinking. It asks participants to go wild during ideation, to defer judgement, and to push beyond safe or obvious ideas. This unleashes creativity but also creates space for deeper personality traits to emerge. A highly open, spontaneous individual may propose unconventional concepts, while a more structured, risk-averse colleague may feel threatened or annoyed by the perceived lack of discipline. If one team member dominates the discussion, others may feel excluded or resentful. If a facilitator pushes the group to generate ideas rapidly, introverted or reflective members may struggle to contribute in time. The very qualities that make Design Thinking powerful (openness, speed, ambiguity) also make it more emotionally demanding than standard working groups.

Second, the short duration of many Design Thinking projects means that storming happens quickly, often within the first day or two. In a traditional team with a longer timeline, storming may take weeks to emerge and be resolved gradually. In Design Thinking, the compressed format forces issues to the surface early, often while the group is still grappling with the problem definition. This can create a sense of chaos or even failure. Participants may become disillusioned, interpreting tension as a sign that the team is not working. Facilitators and participants alike may misread the storming phase as a breakdown, rather than recognising it as a necessary developmental process.

Here lies the central insight: storming is not a failure. It is a rite of passage. The discomfort is a sign that the team is becoming real, that people care enough to disagree, and that the group is moving beyond politeness to authenticity. However, for this process to be productive rather than destructive, teams need to know that storming will happen, that it is normal, and that it does not last forever. Misunderstanding this can lead to premature withdrawal, interpersonal damage, or abandonment of the process. Strategies for navigating storming include:

- Normalising conflict as part of the process
- Ensuring everyone understands the purpose of the Design Thinking methodology
- Encouraging metacognition (talking about how the team is working, not just the task)
- Actively managing participation to prevent dominance
- Giving space for both fast and slow thinkers to contribute

With good facilitation and psychological safety, storming becomes a turning point to grow from rather than a dead end.

Phase 3: Norming – Building the Social Contract

Once storming has played out, either fully or partially, the team begins to establish norms. In this Norming phase, the group negotiates unwritten rules about how to work together. There is less tension, more acceptance of differences, and emerging clarity about roles, decision-making, and expectations. Members begin to listen to one another more generously. Trust deepens. Performance improves not necessarily because of increased ability, but because of increased coordination and reduced friction.

In a Design Thinking context, Norming often coincides with the transition from ideation to prototyping. The divergent phase has ended; now the group must converge on ideas to test and refine. This requires greater alignment, cooperation, and discipline. If storming was navigated well, the team may now enjoy a sense of cohesion and mutual respect. If not, unresolved issues may fester beneath the surface, leading to uneven participation or low morale.

Importantly, some short-term Design Thinking teams may reach only a partial norming stage before the project ends. With tight deadlines, the group may be forced to agree on a prototype even if social dynamics remain fragile. In such cases, Norming is incomplete but still visible. For example, a tacit agreement that a certain member leads the testing phase, or a shared decision to frame feedback in positive terms. Even minimal norming improves the chances of a coherent output. Facilitators can support norming by:

- Clarifying roles explicitly
- Reinforcing collaborative behaviours
- Encouraging reflection on group functioning
- Helping the team document their decisions and agreements

Norming is the point where the team starts to become more than the sum of its parts. Even if time is limited, some level of norming is possible and valuable.

Phase 4: Performing – Flow, Confidence, and Shared Mastery

The Performing stage represents the culmination of team development. Here, the team operates with a high level of autonomy, trust, and effectiveness. Members know their roles, anticipate one another's needs, and coordinate fluidly. Communication is open, efficient, and constructive. Conflict still occurs, but it is handled productively. The team is focused on achieving its goal, not managing its internal dynamics.

In most Design Thinking projects, especially shorter ones, teams do not reach the Performing stage. The limited timeline may end the project while the group is still in late Storming or early Norming. This is not a failure. Rather, it is a structural reality of time-bound collaboration. However, knowing what Performing looks like can still inform practice. Experienced teams or those with strong facilitation may reach a light version of Performing: shared purpose, efficient prototyping, and mutual accountability. In longer-term design projects, or organisations where Design Thinking is routine, some teams do attain full Performing. These are often cross-functional innovation teams or design labs, where members work together across projects and have time to develop deep collaboration.

In any case, the goal is not to demand perfection, but to recognise the path. The four-phase model helps manage expectations: teams do not become high-functioning overnight. They go through emotional and procedural turbulence. Awareness of this model allows participants to contextualise their experience and remain engaged even during the storm.

Risks and Opportunities in Design Thinking Teams

Design Thinking offers tremendous value: it encourages boldness, empathy, user focus, and rapid experimentation. But it also comes with psychological risks. Because the process demands vulnerability (sharing untested ideas, exposing assumptions, and embracing failure) it amplifies personality traits and tensions. High-Openness individuals may seem erratic to Conscientious teammates. Introverts may feel drowned out by high-X extroverts. Low-Agreeableness thinkers may challenge others harshly in storming, while high-Agreeableness members may withdraw to avoid conflict. These dynamics are intensified in the creative push of DT. Unlike in routine working groups, where tasks are predictable and behaviours constrained, DT invites people to stretch, and stretching reveals fault lines.

Thus, Design Thinking teams must be especially deliberate about managing team dynamics. Investing time in establishing psychological safety, acknowledging differences, and reflecting on group functioning is not a luxury; it is a prerequisite for creative success. Facilitators and team leads should not only guide the method but also shepherd the team through its emotional journey. One of the most helpful tools is simply sharing the four-phase model upfront. When people know that Storming is normal and temporary, they are more likely to stay engaged, less likely to personalise conflict and be better prepared to collaborate.

The key takeaway is this: *Storming will occur*. All members must expect it, understand it, and endure it. Doing so creates the conditions for true collaboration. In high-stakes creative work, managing the team journey is just as important as managing the design process. Knowing the path does not eliminate the struggle, but it gives the team the strength to persist, and perhaps, to perform. This was a short chapter to prepare you for the next, which is considerably longer and deals with team management. It will give you, as a personal innovator, a priceless tool to either manage a creative DT team or participate in a team that is using DT and might or might not use the methods in the next chapter. The important thing is that you have a good understanding of how creative DT teams could be managed. Use can use the knowledge either to influence the team's processes or, if that is not possible, perform your part as well as possible, still adhering to the principles on a personal level as much as possible. And all of that will be much easier if you understand the team dynamics that is the topic of this chapter. Having said that, on to the next chapter.

10. Team Management

Personal innovation often begins as a solo journey, but it does not always stay that way. Even the most self-sufficient personal innovators find themselves sometimes collaborating in teams, whether it is a start-up duo, a cross-functional project group, or a team assembled for a specific challenge. Design Thinking (DT) has given these innovators a rich toolbox of mindsets and methods to drive creativity. Yet one question remains: how should a team organise its work day-to-day? Design Thinking excels at guiding *what* to do (empathise with users, generate ideas, test prototypes, and so on), but it says little about *how to coordinate* those activities within a team. This is where Scrum comes in. In this chapter, we introduce Scrum as a lightweight process framework that can complement Design Thinking for team-based work. Importantly, we are not suggesting a replacement of Design Thinking's exploratory spirit, but an add-on framework to give a creative team rhythm, clarity and accountability as they innovate together.

Scrum is an agile project management approach originally formalised in the 1990s for software development, but its simplicity and power have since made it popular far beyond IT. The term Scrum itself was borrowed from rugby to symbolise tight-knit teamwork moving towards a common goal (yes, students in the courses this book builds on always ask that). At its core, Scrum is a framework for structuring teamwork in complex projects. It emphasises short, time-boxed work cycles and continuous feedback. Scrum does *not* tell you how to design a product or solve a problem. That remains the domain of your good old Design Thinking process. Instead, Scrum provides a minimal set of roles, events, and artefacts that help a team plan work, track progress, and adjust course as needed. It is often described as *lightweight* because it imposes only a small number of simple rules. In practice, Scrum can wrap around existing practices. This means you can run a DT project *within* a Scrum framework: all the empathy research, brainstorming sessions, prototyping, and testing still happen, but Scrum offers a scaffolding to hold those activities together. By doing so, it brings much-needed structure to the creativity of DT without dampening it.

Why is Scrum the perfect complement to Design Thinking? Consider a team including a personal innovator excitedly embarking on a new project using DT. They have tools from DT to understand users and generate ideas. They might even have strong team dynamics awareness (as we explored in the previous chapter). However, without a shared process for managing tasks and time, the project can still falter. Ideas stall because it is unclear who will prototype them and when. User research plans remain talk instead of action. Team members, each busy with their own responsibilities, struggle to maintain momentum. DT gives *creative direction* but not *operational discipline*. Scrum adds that discipline in a friendly, low-bureaucracy way. It creates a cadence for teamwork: a repeating cycle of planning, acting, checking, and refining. This cadence helps a team maintain forward momentum and clear communication, which are important in collaborative innovation. It is important to stress that in our context, Scrum is subservient to the needs of Design Thinking. We are not adopting Scrum as a dominating methodology for its own sake. Instead, we cherry-pick Scrum's framework to support and amplify DT efforts. All the content, the insights about users, the creative solution ideas, and the prototypes come from DT. Scrum does not generate any of that. What Scrum contributes is a workflow management structure: who should be doing what this week, what the team's short-term goal is, when to regroup and review progress, and how to continuously improve the way the team works together. In essence, Design Thinking is our engine of innovation, and Scrum is the lightweight chassis that helps steer that engine in a coordinated direction. For personal innovators sometimes moving into team settings, this combination will be empowering. It allows them to retain the freedom and human-centred focus of Design Thinking while gaining the clarity and efficiency of an agile project framework.

In the sections that follow, we will first give a brief overview of Scrum's fundamental components, its roles, events, and artefacts, and explain how they map onto a Design Thinking-driven project. We will see how Scrum roles like Product Owner, Scrum Master and team members can be interpreted in informal, small-team environments common to personal innovation projects. Then we will walk through Scrum events, from Sprint Planning to Daily Scrums to Sprint Reviews and Retrospectives, illustrating how each ritual supports a DT team's needs for coordination and learning. We will also describe the key artefacts that Scrum uses to keep work transparent and organised. Throughout, our emphasis will be on practical adaptation: how to keep Scrum *light* and tailored for a creative team, excluding those agile elements that are not necessary (for example, we will not employ user stories or manage epics, they are completely unnecessary since DT already provides focus on what needs to be done). Finally, we will offer advice and an example of how a small team, say a mix of freelancers or colleagues, but also including a personal innovator, might actually run a DT project using Scrum. By the end of this chapter, you should have a clear idea of how to blend the imaginative freedom of DT with the simple discipline of Scrum to achieve much better results more easily as a team.

DT ♥ Scrum

Scrum's popularity comes from its elegant simplicity. In a nutshell, a team using Scrum breaks their work into short, fixed-length cycles called Sprints (often around two weeks, though anywhere from one to four weeks is common). At the start of each Sprint, the team plans exactly what they aim to accomplish in that time frame. They then collaborate on those tasks, checking in with each other through brief daily meetings. At the end of the Sprint, they review what was achieved and gather feedback, then hold a retrospective to discuss improvements to their process. This cycle then repeats, Sprint after Sprint, until the project's goals are met. That is Scrum at a high level: a repeating loop of plan → do → review → improve. It is essentially an implementation of a scientific method (see Chapter 5) or a design cycle (plan, act, observe, adjust) applied to teamwork.

Let us align this with a Design Thinking work scenario. Imagine you and a small team are working on a challenge, for example designing a better on-boarding experience for new employees at a company (to pick a context outside pure product design). Using DT, you know you need to start by understanding the users (new employees and managers), perhaps through interviews and observations; then define the problem more clearly, ideate possible solutions, prototype a couple of promising ideas, and test them out. Rather than tackling all of that in one long stretch with no checkpoints, you decide to use Scrum to structure the work. You set a Sprint length, let us say two weeks, which is a comfortably smaller planning horizon. In Sprint 1, your goal might be to complete an initial round of user research (Empathise) and problem definition (Define). In Sprint 2, the goal could be to generate (Ideate) and prototype a set of solution ideas (Prototype). In Sprint 3, perhaps you focus on testing those prototypes with users and gathering feedback (Test). Each of these sprints is like a mini-project within the project, with its own objectives that add up to the overall mission. Importantly, at the end of each sprint, the team will pause to reflect: *What did we learn? What should we do next?* This is perfectly in tune with Design Thinking's iterative nature. After all, DT encourages cycling back and forth between understanding, ideation and experimentation as new insights emerge. Scrum simply formalises that cycling in a team-friendly way. It ensures everyone knows the short-term plan, sees progress frequently, and has regular moments to course-correct.

Scrum is defined by a small set of roles, events, and artefacts that work together to support this cycle. By design, there is not much more to it. Scrum is purposefully incomplete, meaning it does not prescribe techniques or specific tools for everything. You are meant to fill in the blanks with whatever methods suit your context (in our case, those methods are DT tools). This makes Scrum almost by definition very compatible: it will readily accommodate the interviews, brainstorming sessions, journey maps, or prototyping workshops that you decide to run during a Sprint. The only thing Scrum asks is that you fit those activities into the structure of the Sprint and uphold a few basic rules (like doing the Sprint meetings). In return, Scrum provides transparency and adaptability. Team members and stakeholders get a clear view of what is happening and what has been accomplished, and the team can adapt its plan based on real observations. For a DT team, this means insights and prototypes do not fall into a black hole. They are continually being inspected and acted upon.

Before diving deeper, let us outline the main components of Scrum. There are three key roles in Scrum: Product Owner, Scrum Master, and the Team (often called Developers in Scrum terminology, from its software origin, meaning the people doing the work). There are four primary events (meetings) in Scrum: Sprint Planning, Daily Scrum (sometimes called the daily stand-up), Sprint Review, and Sprint Retrospective. And there are three main artefacts (documents or tools): the Product Backlog, the Sprint Backlog, and the Increment (the tangible outcome or deliverable achieved in a sprint). We will explain each of these in turn, and importantly, describe how they apply in a Design Thinking project context with some modifications. Remember, we will keep the focus on how Scrum supports your innovation work. Many traditional Scrum practices like user stories, story point estimations, burndown charts, etc., are not used at all for our purposes. They are not required for Scrum to fulfil its basic role of providing structure, and more importantly, other DT tools do those things better for us. The goal is not to burden a creative team with overhead, but rather to give just enough framework to coordinate efforts and manage complexity as lightly as possible.

Scrum Roles

Scrum defines three roles that together form the Scrum Team. In a formal setting, these roles have distinct responsibilities, but in a small, informal team (like a DT team), the lines can be allowed to blur a bit and one person might wear multiple hats. That is okay as long as the *functions* of each role are fulfilled. Let us look at each role and how it can be adapted to a Design Thinking team.

In Scrum, the Product Owner (PO) is the person responsible for defining what the team should achieve and in what order. They represent the voice of the customer or the end goal, ensuring the team is always working on the most valuable thing at any given time. In a software product team, this would be the person who manages the product backlog, prioritises features, and accepts the work done. For a Design Thinking team, the Product Owner role translates to the person who best represents the project's purpose and stakeholder needs. At Openlab Stockholm, where we work with societal innovations, the PO is instead called the Challenge Giver (CG).

Who might this be in your context? It could be the team member who initiated the project or who has the deepest understanding of the client's or end-user's needs. It might even be an actual client representative if you are working with a client. For instance, if you are a social innovation team developing a solution for a hospital, a manager from the hospital will be the PO/CG and convey what outcomes are most important, and give feedback on deliverables. In a start-up context, the PO might be the founder who has the vision of the product and knows what success looks like. In any case, the PO's main job is to guide the team on what problems to tackle and which ideas or features to prioritise. They help answer questions like Which user pain point should we focus on first? or Among all these prototype ideas, which one should we develop further now?

For personal innovators, taking on a Product Owner role and mindset means constantly aligning the team’s work with the ultimate goals. Design Thinking projects can produce a flurry of insights and ideas. The PO helps the team navigate this creative chaos by saying which direction needs to be explored next. They maintain the backlog of work (more on that artefact soon), which is essentially the to-do list of research activities, concepts to prototype, problems to solve, etc. If something changes, say new user research reveals a more urgent need, the PO can adjust the plan and reprioritise tasks accordingly. One could say they are the custodian of value: they ensure the team’s efforts yield the highest value outcomes for the stakeholders involved, be it the end users, the client, or the team itself.

Decide early in a project whether the PO is an external person/role, in which case the project becomes an externally driven one, or if the PO is one of the team members, making the project internally driven. That choice, which might not be the team’s to make, sets the tone of the teamwork and also influences the team dynamics (see the previous chapter). In practice, for a small team, you might not formally call someone Product Owner, but it is wise to decide who will perform that function. If you are a three-person team, ask yourselves: Who is in the best position to judge and prioritise what we should do next? It might be a shared responsibility, but having a single voice to finalise priorities can avoid confusion. Whomever it is should be empowered to make decisions about scope and priority. For example, deciding to spend another week iterating on a prototype because users found promise in it, or the team has learned enough about problem X, move to ideation now. The rest of the team should trust this person’s guidance, just as in Scrum, the Product Owner’s decisions about the backlog are respected by the team. This prevents endless debate and ensures the team stays focused on the most meaningful work.

The second Scrum role is the Scrum Master. This is perhaps an unfortunate title in some ways, because the Scrum Master role is not a master of the team at all. Rather, it is a servant of the team by being master of the process. Importantly, it is not a project manager or anyone in a managerial decision-making capacity. Rather, think of him or her as a secretary when it comes to the work process. In traditional Scrum, the Scrum Master is responsible for making sure everyone understands and follows Scrum practices and that the team stays true to the Scrum values (such as focus, openness, and respect). They are often described as a servant leader: their leadership is in service to the team's effectiveness. They do not boss people around; instead, they facilitate meetings, help remove any obstacles that are impeding the team's progress, and constantly look for ways to improve the team's collaboration.

In a design-oriented project team, a Scrum Master is essentially the team's facilitator. If you have experienced how a good DT workshop facilitator works, keeping the group on track with the agenda, encouraging quieter members to contribute, smoothing over conflicts, and making sure the environment is right for creativity, then you know what a Scrum Master does across the life of a project. They ensure the Scrum events (planning, daily scrums, etc.) happen and are effective. They remind everyone of the agreed process (for example, we all agreed to have our standup at 9 am, let us stick to that). They also play a role in fostering a culture of continuous improvement by guiding the team through retrospectives (more on those soon) and encouraging reflection.

For a personal innovator in a small team, the Scrum Master role can be taken by one of the team members, a personal innovator or not, or the responsibility can rotate. You will probably not have someone whose full-time job is process coach (as is sometimes the case in larger organisations), but it is valuable to have at least one person who *cares about how the team works*. Think of this role as the guardian of teamwork principles. For instance, if the team falls silent and stops communicating, the Scrum Master gently prompts a discussion or schedules a quick sync. If meetings tend to derail into unrelated chatter, the Scrum Master brings focus. If a team member is facing a roadblock, maybe they cannot get access to a user for an interview or they're stuck waiting for a tool, the Scrum Master tries to help resolve it, perhaps by connecting them with the right person or suggesting an alternative approach.

In addition, the Scrum Master educates and reminds the team of Scrum practices. If your team is new to Scrum, this role might say, "Okay, our sprint is ending tomorrow. Let us not forget we need to do a review and a retrospective. I'll arrange those." Or "I notice our task board is not updated; does anyone need help with it?"

They keep the wheels of the process turning so that the others can focus on the content of the work. In a way, a good Scrum Master in a design team is like a good stage manager in theatre: mostly behind the scenes, making sure everything goes smoothly, stepping in to fix problems quietly, and ensuring the actors (i.e., the rest of the team) can deliver their best performance.

It is worth noting that Scrum Masters can also play a role in protecting the team. Creative design work can be sensitive to interruptions or external pressure. If, say, a manager outside the team keeps trying to redirect the team mid-sprint or pile on extra tasks, the Scrum Master can act as a buffer, politely educating that stakeholder on the agreed process. This keeps the team from thrashing and allows them to maintain a state of flow. In summary, the Scrum Master ensures *process integrity* and shields the team from unnecessary disruptions, fostering an environment where innovation can flourish.

The third role in Scrum is simply the Team itself, sometimes called the Developers in the Scrum Guide (we use developer in a broad sense: the people who develop or create the product or outcome). In a Scrum context, the team members are self-organising professionals who actually do the hands-on work of the project. This includes everyone contributing to the solution: designers, researchers, engineers, writers, marketers, and anyone whose work is needed for the team to achieve its goal. Scrum intentionally uses the term *team* as a unified entity to emphasise that everyone is in it together, without sub-teams or hierarchy inside the core team. They share accountability for delivering results.

In a Design Thinking team with one or more personal innovators, the Team likely includes all the main contributors: perhaps a UX designer, a business analyst, a subject matter expert, maybe a developer if a digital prototype is needed, or a service designer, etc. You yourself, as a personal innovator, are likely a team member contributing your expertise (be it creative facilitation, research, or whatever your strengths are). The key shift that Scrum brings is encouraging the team to be cross-functional and collaborative. Rather than each person only sticking to their speciality in isolation, the Scrum mindset is that the team works together as a unit on the sprint goals. This does not mean people abandon their skills; it means they coordinate closely and pitch in to help each other where possible so that the team succeeds as a whole. For example, if one task is to conduct user interviews (usually a researcher's job) and another task is to sketch solution ideas (usually a designer's job), the team does not treat these as silos. Perhaps the designer joins a couple of interviews to gain empathy firsthand, while the researcher sits in on ideation sessions to provide input. This cross-pollination ensures that insights flow freely and no part of the project becomes a disconnected silo.

Scrum teams are meant to be self-managing, which is very empowering for a creative group. Self-management means the team collectively decides who will do what and how, rather than being directed by a boss. In a Scrum sprint, once the goal and backlog items for that sprint are agreed (with input from the Product Owner on priorities), the team is trusted to figure out the best way to achieve it. They might distribute tasks according to expertise, or pair up on challenging problems, or swarm together on one big task, whatever works. This autonomy fits well with the Design Thinking culture, which values initiative and collaborative creativity. Team members can reorganise themselves as needed. For instance, if halfway through a sprint they realise a prototype is more complex than thought, they can decide among themselves to have two people focus on it while another takes on some extra research tasks to lighten the load, ensuring the sprint goal is still met. There is no rigid hierarchy telling them that Person A must do Task X. The team figures it out. This fluid way of working taps into each member's intrinsic motivation and usually leads to a happier, more engaged team.

In practice, if you are setting up Scrum in a small team, make sure everyone understands this sense of shared ownership. The success of each sprint is the responsibility of all team members, not just the person assigned to a particular task. If one person falls behind or hits a wall, the rest should care and adapt, rather than thinking it is not their problem. Regular communication (as we will discuss in Daily Scrums) helps the team stay coordinated on this. When done right, a Scrum team becomes a tight-knit group where people leverage each other's strengths and compensate for weaknesses, very much what you want in any collaborative innovation effort.

One more note: in some settings, you might formally designate someone as Product Owner and Scrum Master, who are also part of the team, but in many smaller teams, these roles are taken by people who also contribute to the work itself. For example, you as a personal innovator might act as both the Product Owner (because you are leading the project vision) and a team member (doing research or design), and maybe a colleague acts as Scrum Master while also doing development work. This is fine. Scrum does not forbid people from wearing multiple hats, especially in small groups; it only clarifies the responsibilities that need to be covered. So long as the team is clear on who is guiding priorities (PO) and who is facilitating the process (SM), and everyone is collaborating on execution (team members), you have the roles effectively filled.

The Scrum Process

With the roles established (Product Owner guiding *what* to work on, Scrum Master guiding *how* the team works, and the Team doing the work), let us explore the heartbeat of Scrum: its events. Scrum prescribes a set of events (or meetings) that occur in a regular cadence each Sprint. These events create a repeating pattern that structures the team's collaboration. They ensure that planning, communication, feedback, and improvement happen consistently. For a Design Thinking team, these events can be a game-changer. They introduce a disciplined rhythm to what can otherwise be an ad-hoc process. Yet they are straightforward and typically short, respecting everyone's time and need to actually do work.

The main events in Scrum are: Sprint Planning (at the start of a sprint), Daily Scrum (each day of the sprint), Sprint Review (at the end of the sprint, to present outcomes and get feedback), and Team (or Sprint) Retrospective (also at the end of the sprint, to reflect on the team's process). And for learning between projects, there is a Project Retrospective at the end of the final sprint instead of a Team Retrospective. Additionally, the Sprint itself is considered a container event, basically the fixed time period in which work happens. Let us break down each event and see how it can be used in a Design Thinking project context.

A Sprint is the fundamental time-box of Scrum. Think of it as a micro-project, typically lasting two weeks for many teams if working full-time (though some teams use one week, and some use up to four weeks, depending on how fast they want feedback versus how much time they need to produce something meaningful). The key is that a Sprint has a fixed duration and a clear goal. At the beginning of each sprint, the team commits to what they aim to accomplish in that sprint. This is often captured as a Sprint Goal. In Design Thinking terms, a sprint might align with a phase or a particular set of activities, but it does not have to map one-to-one to the classic DT stages. You might use one sprint to mostly Empathise and Define, another to Ideate and Prototype, and a third to Test, or you might mix modes within a single sprint. The guiding idea is: give the team a short-term focus.

The Sprint starts with Sprint Planning, a meeting where the whole team (Product Owner, Scrum Master, and team members) decides what to do in the upcoming sprint and how they will do it. This is where the Product Owner's priorities meet the team's capabilities. In a typical agile software project, Sprint Planning involves selecting user stories from the backlog and estimating tasks. For our design-centric team, Sprint Planning can be a bit more fluid but should still cover two main things: what is our Sprint Goal? and what tasks will we complete to reach that goal?

Suppose our project is improving the new employee on-boarding experience (from our earlier example). If we are about to plan Sprint 1, and say our aim is to understand the problem deeply, the Product Owner (or team lead) might propose: Our goal for this sprint is to research and pinpoint the biggest pain points in the current on-boarding process, and frame a clear problem statement. That is a Sprint Goal. It gives the team a concrete outcome to strive for: by the end of this sprint, we will have a defined problem backed by research insights. Next, the team discusses what specific tasks are needed to achieve that. They might break it down into items such as: a) Interview 5 new hires and 5 managers, b) Observe one orientation session, c) Compile research findings and do an affinity map, and d) Draft a point-of-view statement (problem redefinition). These items form the Sprint Plan (a term we will revisit under artefacts), essentially the to-do list for the sprint. During Sprint Planning, the team ensures these tasks are realistic to do in the time given and that collectively they will accomplish the Sprint Goal. A few tips for Sprint Planning in a design thinking project:

1. Keep the sprint scope focused. It is tempting to pack in too much (e.g., we will interview 20 people, brainstorm 50 ideas, and build three prototypes all in one sprint). Given the creative nature of the work, it is better to aim slightly lower and actually finish with a solid result than to overcommit and scramble. You can

always iterate next sprint. So, perhaps decide on one or two primary activities (like research in sprint 1, ideation in sprint 2, etc., or a balanced combination if the sprint is longer).

2. Ensure the Sprint Goal is *tangible*. For example, understanding user needs is a bit vague, whereas identifying the top three user pain points and articulating a problem statement is more concrete. Or in a later sprint, instead of “brainstorm solutions”, a goal could be to develop and test at least one low-fidelity prototype to see how it addresses the top problem. This tangibility helps at the Sprint Review to clearly say if you achieved it.
3. Encourage team input. Sprint Planning should not be the Product Owner dictating tasks. It is a collaboration: the PO brings the priorities and the team brings knowledge of feasibility. If someone has a concern, e.g. a researcher might say that five interviews might not be enough to validate or a designer might say that she needs a day to mock up a prototype for the test, incorporate those perspectives. This ensures the plan is realistic and team-owned.
4. Time-box the planning meeting itself. For a two-week sprint, planning might be max two hours. It should be focused and not eat into too much working time. The Scrum Master can facilitate keeping it efficient: address the goal, list tasks, confirm understanding, and then move on.

Once Sprint Planning is done, everyone should leave that meeting clear on what the sprint’s objective is, what each person will be roughly working on (at least initially), and how success will be measured at the end of the sprint. It is a moment of alignment that sets the stage for effective collaboration.

Once the sprint is underway, each day the team performs their tasks: conducting interviews, analysing data, sketching ideas, coding prototypes, etc. In independent personal work, it is easy to just dive into these tasks and reconvene only much later. However, a team needs to stay coordinated continuously, especially when the work is interconnected. This is where the Daily Scrum comes in. The Daily Scrum (or daily stand-up) is a brief meeting, ideally about 15 minutes, held every day, typically at the same time and place. In this meeting, team members quickly update each other on what they did since the last meeting, what they plan to do today, and if they have any impediments or need help with something.

The power of the Daily Scrum lies in its regularity and brevity. It is not a long discussion or a status meeting for a manager; it is a rapid alignment exercise for the team, by the team. In a creative project, things change fast. You might discover a new insight that changes an approach, or run into a snag recruiting a user for testing. By sharing these things daily, you prevent minor issues from festering into major delays and you enable quick knowledge transfer among team members.

For example, let us say yesterday Alice interviewed two new hires and learned something surprising about their first-day experience, while Bob was working on mapping the on-boarding journey, and Carol, the Scrum Master, was arranging a visit to observe an orientation session. In the daily scrum this morning, Alice might say that she completed two interviews yesterday, and a key insight was that new hires feel overwhelmed with paperwork on Day 1. Today, she plans to interview one more person and then start compiling notes. She might need help later synthesising the findings. Bob might say that he finished a draft journey map of the current process, and will today incorporate Alice's interview insights into it. Carol might report that she confirmed the team can observe the orientation next Thursday. In about 10–15 minutes, now the whole team knows: interviews are yielding particular insights (and maybe they all now know to look for issues around paperwork), the journey map is underway and will integrate these findings, and an observation is scheduled (which they might all attend or at least prepare for). This daily touch point keeps everyone on the same page and allows micro-adjustments. Perhaps from hearing Alice, Bob decides to add a specific question in the next interview, or Carol realises the observation should specifically note where paperwork is involved. These subtle course corrections can dramatically improve the team's coherence and the quality of the outcome. A few guidelines make daily scrums effective for a Design Thinking team:

- *Keep it short and focused*. It is not a brainstorming session or problem-solving meeting. If a big issue comes up, note it and perhaps have the relevant people chat right after the stand-up to solve it. The whole team does not need to dive into the details during the 15 minutes; they just need to know it is an issue and that a plan to fix it will be made. This discipline ensures that the daily scrum does not become a time sink.

- *Same time, same place* (or channel). Routine helps. Maybe every morning at 9:30 in the team space or via a quick video call if remote. Consistency makes it a habit and avoids coordination overhead. In some creative cultures, people worry a daily meeting disrupts flow, but at 15 minutes, it is a short pit-stop, not a disruption. Many teams do it standing up to emphasise brevity and energy.
- *Everyone speaks, everyone listens*. Each core team member should share their update. Even if someone's work seems solitary (say, one person was solely doing desk research all day), it is important to share progress so others know where things stand. And everyone listens because sometimes two pieces of information from different members connect in a useful way. The Scrum Master ensures the vibe remains supportive and not like an interrogation. It is the team checking in with each other, not reporting to a boss.
- Highlight impediments. If you are stuck or something's blocking you, daily scrum is the time to raise your hand. In a supportive Scrum culture, admitting a blocker is not seen as failure, rather it is expected. The worst case is staying silent and the team discovering a blockage too late. Daily scrum prevents that through transparency.

In a design project environment, the daily scrum might also be a place to share quick inspirations or user anecdotes that could spark creativity. But caution not to let it turn into a prolonged discussion. If someone learned a really striking user quote, they can drop it in. A comment like that in daily scrum keeps the user's voice present. If it leads to an idea or needs unpacking, the team can follow up after the stand-up or in a later session. Thus, daily scrums can also serve to maintain a human-centred focus every day, by continually surfacing what we are learning about the users or problem as we go.

At the end of the sprint, after, say, two weeks of work, the Scrum framework calls for a Sprint Review. This is essentially a show-and-tell session where the team presents what was accomplished during the sprint to key stakeholders and discusses it. In software teams, this often means demoing working features to product owners, managers, or customers to get their feedback. For a Design Thinking team, the Sprint Review is a chance to show the outputs of your design work and get feedback from outside the core team. This could include the client, users, or other stakeholders who have an interest in the project.

The nature of what you show will depend on the sprint goal. If the sprint was a research sprint, you might present the insights you gathered, e.g. a summary of user pain points, maybe with some quotes or persona sketches. If the sprint was about ideation and prototyping, you could demonstrate the prototypes built (physically show the mock-up or walk through the storyboard, etc.) and perhaps even share highlights of user reactions if you managed to test quickly. The aim is twofold: inspect the outcome (see what was achieved and how well it addresses the problem) and gather feedback or new ideas from people outside the immediate team.

Let us imagine in Sprint 2 of our project, the team prototyped a new on-boarding tool (say an interactive welcome guide app for new hires) and tried it with a few current employees for initial feedback. In the Sprint Review, they might invite the HR manager (who is a key stakeholder), maybe one of the new hires who gave feedback, and a senior executive sponsor. The team would walk these stakeholders through what they made: “Here’s our prototype welcome app, it has these three sections… we tested it with 3 recent hires, and here’s what they said... one loved the quick links to resources, another was confused about the timeline feature.” They might even do a live demo of the prototype. Then they invite discussion: does the HR manager see this as addressing the issues they hoped it would? Does the new hire find it an improvement over the current situation? What concerns or suggestions do they have?

This meeting should be conversational. It is not a formal presentation where the team defends their work. The Product Owner typically facilitates the Sprint Review, making sure the right people are there and the right topics are covered. The stakeholders give input. Or a stakeholder might raise a constraint. All this feedback is gold for the team because it helps refine the direction. Scrum is built on the idea of inspect and adapt; the Sprint Review is a formal point of inspection of the *product* (or solution) and adaptation of the backlog (the plan of what to do next).

For a design team, you might also use the Sprint Review to incorporate actual user feedback into the project trajectory. If your stakeholders include end users (which, ideally in human-centred design, they often should), hearing their reaction at this stage is invaluable. If, for example, the prototype did not resonate with users, it is far better to learn that now, after a two-week sprint, than at the end of a six-month project. You can then pivot or tweak accordingly. If users loved it, that validation energises the team to keep going in that direction.

It is worth noting that in some Design Thinking workflows, teams wait until a final presentation to show their work. This is not recommended for a personal innovator. Scrum encourages showing iterative progress instead. This can be a cultural adjustment. Stakeholders might not be used to seeing half-finished prototypes or preliminary findings. But by educating them through the Sprint Review practice, you build a culture of iterative development and continuous engagement. They start to expect ongoing involvement rather than big reveals. Many stakeholders appreciate this because they feel more involved and can influence the outcome early on. For personal innovators working with clients, this is a good way to manage expectations and ensure the client is on board at each step. There are fewer unpleasant surprises at the end because the client's feedback has shaped the solution throughout. When conducting a Sprint Review:

- Prepare what to show beforehand, but keep it minimal. You might have a few slides or a demo environment. Avoid spending a ton of time polishing a presentation. The focus is on the work itself, not flashy slides. This is a working session, not a sales pitch.
- Be honest about what is done and what is not. If you also hoped to test a second prototype but did not get to it, you can mention it. Stakeholders value transparency. It also sets up what might be tackled next.
- Encourage questions and discussion. After showing your outputs, ask stakeholders what their thoughts are. Does this address the issue from your perspective? Any concerns? This should feel like a collaboration, not a one-way briefing.
- Mind the time. Sprint Reviews can vary in length based on how much there is to show and how many people are involved, but for a small project, maybe an hour

is sufficient. If remote, sometimes recording a quick walkthrough of the prototype to share can be effective too, but a live meeting is better for interactive Q&A.

- Document key feedback. Someone (often the Product Owner or Scrum Master) should jot down important points. These will inform adjustments to the backlog. For instance, if stakeholders suggest a new idea or a requirement, that likely becomes a new item to consider in the next sprint planning.

By the end of the Sprint Review, ideally the team gets a pat on the back for what they accomplished (celebrate the progress!), and a richer understanding of what to do next. This naturally leads to updating the backlog and thinking ahead to the next sprint. But before that next sprint starts, there is one more important event focused not on the *product* you are building, but on the *team* itself: the Retrospective.

The Sprint (or Team) Retrospective is the final event in a sprint, typically held after the Sprint Review and before the next Sprint Planning. If the Sprint Review was about *what* we made, the Retrospective is about *how* we worked. The team comes together to reflect on the sprint that just ended and discuss what went well, what did not go well, and what improvements they can make for the next sprint. It is essentially a structured reflection session for the team's process and collaboration. For a personal innovator, this practice is incredibly valuable. It creates a regular space to fine-tune teamwork and constructively address any friction or issues. Remember the previous chapter's discussion on team dynamics: teams benefit greatly from pausing to reflect on their ways of working and resolving tensions. The retrospective is a built-in mechanism to do exactly that.

A typical retrospective involves the Scrum Master (or whoever is facilitating) asking the team: What has gone well (I like), what can be done better (I wish), and other thoughts (what if). There are many formats (at Openlab, teams use sticky notes to collect thoughts under What went well / What can be improved / New ideas), but the goal is the same. Everyone has a chance to voice observations: It is a safe space to bring up concerns because the spirit is one of learning, not blaming. The Scrum Master ensures psychological safety in the retrospective, encouraging openness and keeping the discussion respectful and focused on actions. In a Design Thinking project scenario, retrospectives might surface both procedural and interpersonal insights. For example:

1. *Process insight*: The team might realise that they spent too much time perfecting interview questions and, as a result, rushed through analysis. In the retrospective, they note this and decide next time to time-box the planning of research to allow more time for synthesising results.
2. *Tool/Technique insight*: Perhaps they tried a new online whiteboard tool for remote collaboration, and it was clunky. The team could agree to try a different tool or approach next time. Or conversely, maybe using a digital Kanban board to track tasks was helpful. They commit to keep using it and maybe customise it further.
3. *Team dynamics*: Maybe one team member felt they were overwhelmed with tasks, while others could have helped if they had known. This can be discussed in a sensblie way. Such admissions lead to a collective agreement to watch out for each other's workload more actively.
4. *Scrum ada*ptation: Since the team is new to Scrum, they might reflect on the framework itself. The team can tweak timing or format as long as they keep to the intent of Scrum. For instance, some teams might find a quick text-based check-in works on certain days if not everyone can meet. They might decide if that is acceptable or if a meeting is always preferred.

What is essential is that the retrospective results in a small set of actionable improvement items. Scrum Masters often ensure the meeting does not end until the team has identified at least one or two concrete changes to implement in the next sprint. These could be experiments like trying a 20-minute time cap for brainstorming sessions next sprint to keep them focused or creating a shared glossary so that the terminology is consistent among the team and stakeholders. It could be something simple such as starting Daily Scrum 15 minutes earlier to have more overlap with colleagues in a different time zone. The team agrees to these changes, and then in the next Sprint Planning or during the next sprint, the Scrum Master will remind and encourage the team to follow through on them. Over time, these continuous small improvements add up. The team becomes more efficient and more attuned to each other. For personal innovators, this means you are not only improving the solution you are designing but also improving your design process and collaboration skills continuously. It is a bit of a meta perspective, designing the teamwork itself. But this meta-design of process is vital; great innovation teams are not born overnight, they are built through iterative refinement of how they work together. The retrospective is the formal opportunity to do that.

In sum, the Sprint Retrospective closes the feedback loop of Scrum: we had a hypothesis about how to work, we carried it out, now we analyse the result and adjust. It embodies the growth mindset that is at the heart of both Scrum and Design Thinking: always learning, always adapting. And by making it a regular practice, it avoids issues festering. Teams that skip retrospectives often let minor annoyances become major dysfunctions. By contrast, teams that embrace Retrospectives find that even conflicts or failures become productive lessons. For example, if something went wrong (maybe a user test was poorly run), rather than that causing blame or demotivation, the retrospective allows the team to frame it as learning and move forward stronger.

With the roles, events, and artefacts (the remaining piece) covered, you will have the complete picture of Scrum and how it operates in tandem with Design Thinking. Let us now turn to those artefacts, the simple tools Scrum uses to keep work transparent, and see how they fit into our project framework.

Artefacts

Scrum's artefacts are essentially the information radiators of the process. They make the work and progress visible to everyone involved. There are three primary artefacts in traditional Scrum: the Product Backlog, the Sprint Backlog, and the Increment (or Product Increment). In plain terms, these correspond to: the list of everything that could be done (product backlog), the list of what the team *is* doing *this sprint* (sprint backlog), and the outcome or deliverable produced (increment). The names chosen for these simple artefacts are surprisingly bad: who wants to have a *backlog*? It sounds like the job has not been done properly. A feeling of lagging behind through a large backlog is not the least inspiring. For a design thinking project, we can adapt (and rename) these artefacts to suit our needs, ensuring that all the tasks and outputs of our creative process are captured and transparent. Let us examine each in turn.

The Product Backlog is an ordered list of everything that might be needed in the project. In software, it is usually a list of feature ideas or user stories, things we do not use in Design Thinking. In a DT project, your “backlog” will look a bit different, but the concept is the same: it is a single, prioritised list of work items that the team might tackle in order to achieve the project’s objectives. These items could include research activities, problem areas to explore, ideas to prototype, tasks to implement a solution, and more. A reasonable name for this artefact in a DT process is the Project Plan. Basically, if it is something the team has identified that needs to be done or a potential solution component, it goes in the Project Plan.

For example, at the very start of a project (perhaps even before the first sprint), the team, led by the Product Owner, can brainstorm what needs to be done. Suppose our goal is to improve the on-boarding experience. The initial Project Plan might include items like Interview new hires, Interview managers, Map current on-boarding journey, Identify pain points, Research on-boarding best practices, Brainstorm solution ideas, Prototype an on-boarding checklist too”, Prototype a mentorship program concept, Test prototypes with new hires, Develop final on-boarding toolkit, and so on. These vary from broad tasks to specific ideas. That is okay; the Project Plan is a living artefact. It will be refined over time.

One principle of Scrum is that the backlog is ordered by priority, with the most important (highest value or most urgent) items at the top. In our design context, what is a priority might equate to what is most critical to learn or build next. Early on, research tasks will be high priority (since you need to understand the problem first). Later, certain solution concepts might take priority if they seem promising. The Product Owner typically owns this ordering, taking input from the team and stakeholders. If a stakeholder says that they absolutely need to address the paperwork issue, the Product Owner ensures backlog items related to that are near the top.

It is important to note that the backlog is not set in stone. It is expected to evolve as you gain new insights. Design Thinking is inherently exploratory, so as you empathise and define, you might discover the problem is different from what you thought. Thus, new items are added or existing ones re-prioritised. For instance, mid-project, you might add Interview IT department about their role in on-boarding if you realise they are an important part of the process. Or you might drop an item like Prototype a mentorship program if research shows mentorship is not a pressing need. Scrum embraces this agility: the backlog is continuously refined (a process often called backlog grooming or refinement, usually done informally during or between sprints). This ensures the team always works on the most relevant things as the project evolves.

Managing the backlog for a small team can be simple. You do not need special software (though tools like Trello, Jira, or even a spreadsheet can help, at Openlab we recommend Trello). A shared document or a physical whiteboard with sticky notes can serve as a backlog, as long as it is accessible and visible to the team. Each item on the backlog should be brief but clear enough that the team understands what it means. For example, an item could be phrased as an action or outcome: “Conduct five user interviews with new hires, or Prototype: Interactive Day-1 Checklist (web app). We do not need the formal user story syntax (As a new hire, I want an interactive checklist so that…) unless the team finds it helpful. Given that we are not focusing on Scrum’s user story format, clarity in plain language is fine.

An important part of backlog management is scope and content come from Design Thinking artefacts. The empathy maps, problem statements, and ideation sketches you create in DT can directly inform backlog items. Suppose your team generates 10 different solution ideas in an ideation workshop. Not all will be pursued, but you might put a few of the promising ones as separate backlog entries, like Prototype idea: Buddy System for new hires or Explore idea: gamified learning module. The Product Owner, with team input, might prioritise one or two to actually prototype next sprint and leave others lower on the list for later consideration. In this way, the backlog acts as a bridge between the free-form creativity of workshops and a more concrete project plan.

Throughout sprints, the backlog will be referenced during Sprint Planning (to pull items into the sprint) and updated after Sprint Reviews (perhaps marking items as done, adding new ones based on feedback). By the project’s end, the backlog will tell the story of what was considered, done, or dropped. For transparency, it can also be shared with stakeholders if appropriate, so they see the roadmap of activities and features. However, in a design project, it might contain a lot of technical jargon or shorthand, so often the Product Owner communicates the high-level picture rather than sharing the raw list.

To summarise, the Product Plan (aka backlog) in our context is the master list of work that feeds the team's sprints. It keeps the team oriented towards the goal while remaining flexible to change. Keeping a well-groomed backlog means the team is never at a loss. There is always a next item to tackle that has been deemed important. This reduces downtime and confusion. It also prevents the project from meandering aimlessly, because the existence of a prioritised list imposes a gentle discipline: you cannot just chase random ideas without considering their relative importance.

If the Product Backlog is your entire playbook, the Sprint Backlog (Sprint Plan) is the play you are executing right now. At Sprint Planning, as described, the team selects a set of items from the Product Plan to work on in the upcoming sprint. Once selected, those items (and only those) form the Sprint Plan. The Sprint Plan is essentially the subset of the Product Plan that the team commits to accomplishing in the current sprint, plus a plan for how to accomplish them. And again, you need to know and be aware of the term Sprint Backlog, but try to avoid it if you can.

In practical terms, if you are using a board or list, you might have a separate section or a new board for the current Sprint. Such boards are often called Scrum Boards. Many teams use a simple task board to manage the sprint backlog, often with columns like To Do / Doing / Done. Initially, during sprint planning, all selected tasks start in To Do. As the sprint progresses, team members move tasks into Doing when they start work on them, and into Done when completed. This visualises progress daily. For a physical team, this could be sticky notes on a whiteboard; for a remote team, tools like Trello, Notion, or Jira can do this. In fact, at Openlab, we encourage the use of Trello even though we have physical campus teams. This is for a couple of reasons. If someone is ill or away for one day, they can follow the progress and update themselves seamlessly. And if worse comes to worst and something happens to the whiteboard, the team is really at a loss.

As an example, building on our Sprint 1 scenario (research and define phase): the Sprint Plan might include these items: Schedule and conduct five interviews, Observe orientation session on day Z, Compile interview notes and identify themes, Map current process and pain points, or Draft problem statement. Each of those might further be broken into smaller subtasks if needed. Scrum does not require you to, but teams often break items into actionable tasks. Perhaps Schedule interviews and Conduct interviews are separate subtasks, so that progress can be tracked. The team divides and conquers these, pulling tasks from To-Do to Doing to Done. Very easy in Trello using three columns of cards. This gives a clear visual picture of how the team is doing in the current sprint, and this type of visuals are recommended in DT, so again Scrum fits very well. By mid-sprint, some tasks will be done (e.g., all interviews completed), some in progress (analysis underway), some yet to start but planned.

The Sprint Plan is owned by the team. Unlike the Product Plan, where the Product Owner (if external) has a strong say in priorities, the Sprint Plan is more of an execution list. Once the sprint has started, the team should feel free to adjust within that scope. If they discover a task was underestimated (say, mapping the process is bigger than first thought), they can decide to break it into two or adjust as needed, possibly deferring something less critical if absolutely necessary. Ideally, you want to avoid mid-sprint scope changes, but if reality hits, the team can renegotiate among themselves (and with the Product Owner if it affects the sprint goal) to stay on track. The Scrum Master monitors that any changes still keep focus on the Sprint Goal. Importantly, no one outside the team should be changing the Sprint Plan during the sprint. The Product Owner or others should not slip in one more task once the sprint is underway, unless the team agrees and it is minor. Otherwise, it undermines the whole point of committing to a set of work for that time box. This calls for some skilful role separation if the PO is internal, and thus both a team member and a goal setter. If something urgent comes up mid-sprint, a healthy Scrum practice is usually to either swap it with something of equal size (so workload stays constant) or wait and put it in the next sprint’s plan, unless it is an absolute emergency.

In a DT context, sometimes unexpected opportunities arise, such as maybe mid-sprint, you get a surprise chance to talk to a key user who was previously unavailable. Scrum is not meant to be so rigid as to make you ignore that. The team can adapt: possibly take that interview (even if not explicitly planned) and adjust elsewhere. Scrum’s flexibility allows responding to new information, as long as the team communicates and adapts intelligently. The Sprint Plan can be updated to reflect such changes, it should be a helpful tool, not an impediment.

By the end of the sprint, ideally, everything in the Sprint Plan has moved to Done.

This means the team accomplished what they set out to. If something is not done, it is usually carried back to the Product Plan (possibly re-prioritised or re-estimated) and considered in the next planning. There is no use keeping it in the Sprint Plan once the sprint is over. Incomplete items return to the main backlog to be replanned. We try to keep that to a minimum by planning realistically, but in innovation projects, uncertainty is high, so occasionally a task will not get done. Scrum treats that as a learning: perhaps it indicates we overcommitted or a task was larger than thought. The retrospective can reflect on that, and the next sprint plan can adjust accordingly.

One more element in Scrum is the Sprint Goal, a short statement of what the sprint is meant to achieve. While the Sprint Plan lists the *what*, the Sprint Goal states the *why*. For example, the Sprint Goal might be Understand and clearly articulate the main user needs and problems in on-boarding. The Sprint Backlog items all serve that goal. If mid-sprint, something comes up that is not helping achieve that goal, ideally it waits. The Sprint Goal can be noted somewhere visible (top of your board or in the sprint plan document) to remind everyone. It helps ensure that if trade-offs are needed, you keep the core objective in focus. In creative work, it is possible to get sidetracked with an intriguing tangent; a well-crafted Sprint Goal can anchor the team’s priorities.

In practice, maintaining a Sprint Backlog is not burdensome: if using a physical board, it might take 5 minutes a day to move sticky notes; if digital, similarly quick updates. The reward is clarity: at any given moment, anyone (including stakeholders if they walk in) can see what the team is working on right now and how far along it is. That transparency is part of Scrum’s emphasis on making progress visible. It not only helps manage the work, but psychologically it gives a sense of progress, which can be motivating and is an aspect of the DT philosophy. There is a little dopamine hit when you move a task to Done! For a team that might not have a formal boss or PM checking in, the Sprint Backlog serves as your self-management tool. It keeps you honest about your commitments and helps the Scrum Master spot if things are stuck (e.g., a task lingering in Doing column for many days might prompt them to ask in Daily Scrum if help is needed or if something’s wrong).

The Increment is what you get when you finish a sprint. It is the sum of all the work completed during that sprint, integrated and in a usable state. In software, an increment typically means a working piece of software (potentially shippable to users). In a Design Thinking project, the increment might not always be a “shippable product” in the traditional sense, but it should be a concrete advancement that moves the project forward in a meaningful way. It could be a report of research findings, a set of personas, a problem statement, a prototype, test results, or a refined concept. Whatever the sprint goal was oriented around.

Think of the Increment as the answer to What do we have now that we did not have two weeks ago? After Sprint 1, our increment might be a validated understanding of user needs, documented as key insights and a clear problem statement. That is a deliverable, albeit a knowledge one. After Sprint 2, the increment could be a low-fidelity interactive prototype of an on-boarding app, tested with five users, with feedback collected. After Sprint 3, the increment might be an improved prototype (in the second double double diamond in Openlab’s D3 process) that addresses initial feedback, plus (in Sprint 4) a roadmap for implementation. If the project is long enough to go beyond prototypes into actual implementation, then later increments could be functioning pieces of the solution delivered gradually.

The key attribute of an Increment is that it is complete and usable in some way. Scrum emphasises the Definition of Done, a checklist or criterion that an increment must meet to be considered done. For a DT team, we can define what “done” means for various tasks or deliverables to ensure quality. For example, an interview is “done” when notes are written up and key quotes extracted, not just after the conversation. A prototype is “done” when it is built to a level that someone outside the team can interact with it and understand the concept. Having this clarity prevents misunderstandings (one person might think a task was done, but others expected more detail). It also aligns with professional rigour. Even though we are keeping things light, being clear on done-ness ensures we truly finish things properly and can move on. However, Increments are often implicit in a DT process. By its innovation-focused nature, a DT process is harder to plan in terms of describing what the actual achievement should be at the end of every sprint. This is elegantly solved by the Sprint Review meeting. What is shown there is, by definition, the Increment. This way, there is one less artefact for a DT team to keep track of, and more time and energy can be devoted to innovation.

One of the benefits of Scrum is that by delivering increments regularly, the project produces value continuously rather than all at the end. In design projects, this translates to having intermediate artefacts that are valuable in their own right. For example, research insights from Sprint 1 could be shared with the client or stakeholders and might be immediately useful (maybe the client learns something new about their employees they did not know before, influencing other decisions). Or an early prototype, even if it is not near the final solution, might spark new ideas or catch issues early. Each increment builds on the last, so by the end of the project, you have not just done a lot of activities; you've accumulated a suite of tested, refined elements that together make the final solution stronger. Another advantage is risk mitigation. If one sprint's increment reveals that an approach is not working (say, the prototype was not at all liked by users), the risk is exposed early and the team can adapt. Contrast that with a non-iterative approach where you might invest all effort into one big final output and only then find out it misses the mark. Frequent increments mean frequent validation opportunities. All those benefits are also achieved by DT teams using implicit increments through Sprint Reviews.

It is good practice after each sprint (usually during the Sprint or Team Review) to treat the increment as potentially shippable or presentable, even if you might not ship it to end users yet. For example, you might not roll out an incomplete solution organisation-wide, but you treat the prototype as if it were a real product to get feedback. This mindset ensures that quality is maintained. The team strives to make each increment as solid as possible given its level of maturity. If it is a research report, have it be coherent and evidence-based; if a prototype, have it be functional enough to test key assumptions; if a piece of code, ensure it is not full of bugs but works for the intended demo.

In summary, the increment is the result of the team's effort each sprint. It embodies the learning or value delivered in that time-box. Over the course of multiple sprints, these increments often come together (many Scrum teams integrate each increment into an evolving product). In a design project, think of each increment as a chapter of the final case study: by the end, you will have the story of problem definition, then prototype v1, then prototype v2 with improvements, etc., culminating perhaps in a final proposed solution or a pilot implementation. Even if the final outcome is a recommendation or a design proposal, the increments ensure that the recommendation is grounded in iterative testing and exploration.

With the roles, events, and artefacts of Scrum now translated into our Design Thinking team context, we have a clear framework for *how to work together*. Next, let us consider how to practically implement this in a small team setting and run through an example to illustrate it in action. We will tie everything together, showing how a personal innovator's team can adopt Scrum step by step, and what benefits they reap from doing so.

Benefits of Scrum

By now, many benefits of integrating Scrum into a design project have become evident. Let us summarise the key advantages a personal innovator and their team gain by using this lightweight version of the framework:

Clarity and Focus: Scrum forces the team to define short-term goals (each Sprint Goal) and the tasks to reach them. This clarity means that at any given week, everyone knows what the focus is. In creative projects, it is easy to feel overwhelmed by possibilities; Scrum narrows the field just enough to keep the team moving forward purposefully, without losing sight of the bigger picture (since the Product Owner maintains the overall backlog).

Steady Progress and Momentum: The regular cadence of sprints ensures that progress does not stall. There is a natural human tendency after an exciting workshop or ideation session to have a lull. Scrum replaces that with a sense of *urgency* in a positive way. With a two-week sprint ticking down, the team is motivated to push to complete the sprint goals. It is like having mini-deadlines that keep the energy up. As one sprint ends, the next begins, creating a continuous flow. This momentum is important for maintaining team morale and keeping stakeholders engaged (they see results each sprint).

Improved Team Communication: Daily Scrums and other Scrum rituals significantly enhance communication. Instead of waiting for weekly check-ins or suffering from silence, the team is in sync every day. Issues are raised quickly, knowledge is shared continuously, and no one goes off in the wrong direction for too long because course corrections happen in real time. This is particularly beneficial for teams that might not have worked together before or are cross-functional. It builds trust since you learn your teammates' working styles and constraints piece by piece and can support each other better.

Early Course Correction: Design Thinking inherently involves dealing with uncertainty and new insights. Scrum complements this by providing frequent checkpoints (end of each sprint and daily micro-checkpoints). If user research reveals a surprise, the next sprint can pivot accordingly. If an idea fails, it fails early in a safe-to-fail environment (a sprint review) rather than at final delivery. The backlog can be continually re-prioritised as new information emerges. This means the project outcome is far more likely to hit the mark, since it is been adapted along the way to reality, rather than clinging to an initial plan that might become outdated.

Accountability and Ownership: When the team commits to a set of tasks in a sprint, there is a gentle accountability that each member feels, not imposed top-down, but as a mutual agreement. Each person's contributions are visible and their impact on the sprint's success is clear. This often fosters a stronger sense of ownership: people feel responsible for delivering because they had a hand in planning it and their teammates rely on them. It also often leads to less procrastination; knowing you will be updating your team daily tends to encourage tackling tasks proactively.

Stakeholder Engagement and Transparency: Through Sprint Reviews, stakeholders (clients, end users, and managers) are kept in the loop and can influence the direction. This not only increases their buy-in (because they see the project evolving and can contribute feedback) but also educates them about the design process in a palatable way. They see the value of iterative experimentation first-hand. The transparency Scrum provides (via visible backlogs, regular demos, etc.) builds trust since stakeholders feel confident the team is not hiding anything or going off-track secretly. Problems are shared openly, successes are celebrated together. For clients in particular, this can differentiate a personal innovator's approach as very responsive and agile compared to traditional consulting.

Higher Quality Outcomes: Each Scrum event contributes to quality. Sprint Planning ensures work is thought through and scoped; Daily Scrums surface and remove impediments quickly; Sprint Reviews bring in feedback to improve the solution; Retrospectives improve the team's process for next time. Meanwhile, the Definition of Done and continuous integration of increments make sure that what is delivered each sprint is in a usable state. The end result is typically a more refined, user-validated solution. Essentially, Scrum builds quality assurance into the very rhythm of the project.

Stress Reduction and Morale: At first, teams new to Scrum worry it might be stressful to have tight loops and frequent check-ins. But many find the opposite: it reduces last-minute panics and the burden of large unknowns looming ahead. By slicing work into manageable sprints, the team can focus and achieve a sense of accomplishment regularly. That regular accomplishment ("we finished something!" every couple of weeks) is a big morale booster. It contrasts with projects where you work for months with no tangible output, which can be demoralising. Moreover, issues are addressed before they become crises, so the team is less likely to find themselves in fire-fighting mode late in the project. Knowing there is a built-in forum to talk about problems (retrospective) also improves psychological safety since team members feel heard and can influence how the team operates.

Lightweight Process (not overhead bureaucracy): Because Scrum is so minimal in rules, it does not create a heavy documentation or reporting load. The focus is on conversations and simple tools. For a small team, this means you are not spending all your time updating Gantt charts or writing lengthy status reports. Instead, you are doing quick stand-ups and updating a task board. The process overhead is low relative to the gains in coordination. Of course, it is possible to overcomplicate Scrum, but as we've framed it (omitting unnecessary agile jargon and artefacts not needed for our purpose), it remains lean. It is more of a scaffold than a cage.

Scalability and Consistency: If a personal innovator finds itself working with different teams or scaling up an initiative, Scrum provides a common language and structure. Team dynamics and individuals vary, but Scrum events and roles give a consistent backbone so one can slot into new collaborations more easily. It is widely understood since many professionals have at least passing familiarity with agile methods nowadays, so introducing it to a new group is not as foreign as some bespoke process would be. For global or remote teams, it especially helps set expectations on how to work together across distances and time zones.

In essence, Scrum empowers design thinking teams to execute better. It does not diminish the creativity since the team still does empathic research, wild ideation, rapid prototyping, etc., just as they would normally. But Scrum surrounds those activities with a supportive structure that ensures the team is aligned, the stakeholders are engaged, and the project is systematically moving towards a solution. It handles the group choreography so that the dancers (the team members) can shine in their performance.

Guidelines for Adopting Scrum

While all this sounds great, it is important to address how to get started with Scrum when your team is small, possibly inexperienced with agile methods, or loosely organised (think: you as a personal innovator and a few colleagues coming together for a one-off project). Here are some practical tips to ease adoption and make it work for you without feeling overly formal.

Start with the Basics: You do not need certification or detailed knowledge of agile frameworks to start. Simply begin by outlining your backlog, setting a sprint length, and scheduling the core Scrum meetings. Explain to the team why you are doing this. Focus on the benefits. A lightweight understanding is enough to pilot it.

Keep Roles Flexible, But Responsibilities Clear: In a tiny team, you might all share responsibilities, but still designate who will take the lead as Product Owner and Scrum Master. For instance, one person might agree to be the point for backlog questions (PO role) and one to facilitate meetings (SM role). They can still do regular work too. If one person ends up effectively filling both roles (maybe you are a team of two), that is fine. Just be conscious when you are switching hats (e.g., right now I’m speaking as the PO and this is what I think is priority).

Use Simple Tools: For managing plans/backlogs and a sprint board, use whatever is simplest: sticky notes on a wall or a digital tool such as Trello. Do not get lost in software. The goal is visibility, not perfection. A shared Google document listing plan/backlog items and a shared Task Board can do the job initially. The tool is less important than everyone accessing and updating it.

Time-Box the Meetings and Make Them Regular: It might feel awkward at first to have daily stand-ups or retrospectives if you've never done it. But stick to it for at least a couple of sprints to feel the value. Keep the daily Scrum to 15 minutes or less; if you truly have nothing new to say (maybe you were heads-down on one task all yesterday), you still state that and finish early. The consistency is key. Similarly, hold the retrospective even if the team feels they know what went wrong. Give it time, you will be surprised what insights come out when you create that space.

Adapt Scrum to Your Context: Remember, Scrum is a framework, not a rigid law. If your team is three people, you are not going to formally stand up and speak to the pigs and chickens (an old Scrum joke). You will just have a quick chat. If you are remote across time zones, daily scrums might be tricky. Maybe you do a written stand-up in Slack each day, and two times a week live, and that is sufficient. It is better to do an approximate version than nothing. Just retain the spirit: regular planning, regular communication, regular review, and regular improvement. After all, DT is about being creative, so let some of that into the process as well.

Educate the Stakeholders Lightly: If your client or other stakeholders are not used to this way of working, give them a simple heads-up. Tell them you work in iterative cycles. Tell them you will show your progress every two weeks instead of waiting until the end. Tell them that this way, they can give feedback and ensure you are on the right track. Most will appreciate that. You need not even call it a Sprint Review to them, just call it a check-in or demo. The terminology is less important than the practice.

Do Not Overdo Documentation: Scrum itself does not demand heavy documents. You might produce documentation as part of the design process (like research summaries or design specs), that is fine. But do not think you need an “Agile report” or something. The info radiators (boards, etc.) and meetings carry most of the info needed. Maybe keep brief minutes of Sprint Reviews or Retrospectives, but only if useful to you. Keep the process lean.

Watch Out for Pitfalls: A few common pitfalls in Scrum adoption are the following. i) *Turning daily scrum into a status report to one person*. It should be team-centric, not reporting to the boss. Make sure each team member addresses the team, not just the highest-ranking person. ii) *Skipping retrospectives*. Under time pressure, teams often drop the retrospective. This deprives you of improvement insights. Even a 30-minute retro is worth it, even if you just identify one thing to do better. iii) *Rigidity*. Sometimes people become zealots, believing that the plan is fixed no matter what. If a major event happens, Scrum is not meant to ignore reality. You can cancel a sprint or adjust if absolutely needed, but treat it as an exception. The framework serves you, not the other way around. iv) *Neglecting human elements*. The tone of interactions matters. The Scrum Master especially should ensure that daily scrums do not become blame sessions if tasks slip, and that quieter voices are heard. Scrum will not automatically make a healthy team culture; you still have to bring emotional intelligence to it.

Scale Up as Needed: If your team grows or you collaborate with another team, Scrum can extend. You might then need to coordinate between teams (there are techniques like Scrum-of-Scrums, but on a small scale, just having a liaison attending each other's reviews might suffice). The fact that you are using Scrum makes it easier to plug into multi-team projects because you can align sprint boundaries or share practices.

Use the Language of Value: Especially when dealing with management or clients, talk about what Scrum enables (faster feedback, more transparency, risk reduction) rather than the jargon. You do not even have to mention the word Scrum if you fear it is misunderstood; you can say that you will use an agile approach. What matters is that they experience the benefits.

By following these guidelines, you as a personal innovator can smoothly integrate Scrum into a team's way of working without feeling it is overly bureaucratic. In fact, after a couple of sprints, many teams report they could not imagine not working this way. It starts to feel natural because it addresses so many common pain points in team projects.

In the journey of a personal innovator, developing one's own creativity and efficiency is paramount. But when that journey intersects with teamwork, having a clear yet flexible framework to guide collective efforts becomes just as important. Scrum, as a lightweight project framework, offers design thinking teams a practical means to organise their collaboration without stifling their innovative spirit. It fills the gap between *what* needs to be done (as illuminated by DT methods) and *how* to do it together efficiently.

Throughout this chapter, we saw that Scrum and Design Thinking are not competitors but complements. DT fuels the project with user insights, creative ideas, and iterative experimentation. Scrum provides the cadence, accountability, and transparency to carry those ideas forward, involve the right people at the right times, and deliver solutions sustainably. By positioning Scrum as an add-on, we ensure that the human-centred, exploratory core of the work remains intact. We do not let process reign over content. Instead, process serves the content. The personal innovator and their team remain improvisational and empathetic, but now with a drumbeat that keeps everyone in sync. For personal innovators stepping into team leadership or teamwork roles, adopting Scrum can be a transformative step. It helps avoid common pitfalls of group work, such as unclear responsibilities, communication breakdowns, and endless planning with no action. Instead, teams embrace a learning-by-doing philosophy, very much in line with DT's bias towards action, structured in repeatable cycles.

One of the subtle gains of this approach is building a culture of continuous improvement and learning within the team. Each retrospective not only improves the team's process, but it also deepens trust and understanding among colleagues. Over time, a team using Scrum can become high-performing (reaching that "Performing" stage of team dynamics perhaps faster than it otherwise would) because it has built strong habits of communication and reflection. In high-stakes creative work, as we noted earlier, managing the team journey is as important as managing the design process. Scrum gives a concrete way to manage that journey, complementing the psychological and social insights from Chapter 9 with a procedural roadmap.

It is worth noting that Scrum is just one agile framework, not *the* one. There are others, and many organisations adapt them in various hybrids (you might hear terms like Kanban, Lean UX, etc.). The spirit common to all is iterative development and human collaboration. We suggest Scrum because of its wide adoption and proven effectiveness in coordinating DT teams tackling complex problems. It is a solid starting point for most teams. Once comfortable, teams often tailor their own hybrid to best fit their context, which is perfectly fine. The ultimate goal is not to be good at Scrum but to consistently deliver value and innovation through effective teamwork. Scrum is a means to that end, not a straitjacket.

As you integrate Scrum into your personal innovator toolkit, remember to maintain the mindset that both Design Thinking and Scrum encourage empathy, experimentation, and adaptation. Empathy not just for end-users, but for your teammates: understanding each other’s strengths, workloads, and viewpoints (Scrum ceremonies like Retrospectives allow airing those). Experiment in your process. Try a new way of doing stand-ups or a new sprint length to see what works best for your team. As project conditions change or as your team matures, tweak the process accordingly.

In a world where problems are increasingly complex and interdisciplinary, the ability to work creatively and systematically in teams is a true superpower that every personal innovator should learn. By going beyond individual design thinking and embracing a team framework like Scrum, the personal innovator elevates not only their own impact but also empowers their collaborators. It transforms a group of talented individuals into an innovation team with direction and cohesion. To conclude, Scrum as a lightweight framework provides the flow and rhythm that can carry a Design Thinking team through the turbulence of creative work to the delivery of meaningful results. It ensures that the path of the personal innovator can be expanded into a path for teams of innovators: structured enough to keep on track, open enough to welcome creativity at every turn. And so, armed with Post-it notes and a backlog (that you call Plan instead), stakeholder maps, and a Scrum Board, you and your team can confidently tackle challenges that none of you could solve alone, marching forward in well-coordinated, innovative sprints towards your collective goal.

11. The PILOT Method

The personal innovator will often encounter complex decision situations and needs to be able to handle them in a structured and efficient manner. We will in this chapter take a look at how easily decisions can also be made using only pen and paper by following the PILOT method. PILOT is not an acronym but refers to being in the pilot's seat when making real-life decisions if the method is followed. The method is based on observations on what information people can easily provide and handle with reasonably preserved quality. As such, the method does not rely on unrealistic assumptions about the personal innovator's time and resources to achieve impeccable decisions. The PILOT method allows the innovators to sit in the driver's seat when they are going to make decisions that require reflection, either alone or as part of a group. In other words, become their own decision pilots with the ability to control decision situations without too much risk of making mistakes.

The PILOT method consists of five decision stages that we will now look into. We will provide a worked-through example to illustrate the stages. The example comes from a personal decision but this does not mean that the PILOT approach is any less suited to decisions for businesses or organisations than to personal decisions. Not at all. The example was chosen so that most readers can recognise themselves and be able to relate to the different decision stages easily. Note that the terms *alternative* and *option* will be used interchangeably in this description to lighten up the presentation. They refer to the same thing.

The method comes in two versions comprising four and five stages, respectively. The four-stage method (PM4) considers the cost aspect from the very start of the process, while the five-stage method (PM5) focuses on functionality in the first four stages and devotes the fifth stage entirely to the cost aspect. The versions are identical if there is no cost aspect involved in the decision. The first three or four stages of the method ensure that we gradually work towards better and better decision information in terms of the features and functions of our options for action. The last stage is the final evaluation of our information base. At each stage, our information base gains in quality. But already after the first stage, we will have a fully operational decision-making basis, and for some decisions, we might decide to make do with that. Time is money, and there are many decision situations where the most preferable option becomes clear relatively early in this process. In that case, there is no need to continue with more stages or analyses. The PILOT method is divided into the following stages:

1. First, we create a pro-and-contra (P-C) list for each alternative. Such a list includes the advantages and disadvantages we can see. We might already make our decision at this point.
2. Otherwise, we record the important characteristics of the possible options in the decision. This may be enough to make our decision.
3. Otherwise, we rank all the options under each criterion separately. This will give us an excellent basis for our decision.
4. If we continue, we assign weights to the criteria according to a ranking order. In PM4, we are now ready for our final decision, while in PM5, we are almost there.
5. In PM5 only, the fifth stage manages the trade-off between cost versus functionality and features of our available options.

Why two versions of the method? There are two distinct types of decision situations. On the one hand, there are decisions where we choose between alternative courses of action either primarily based on their functional properties or where cost is a perspective among others, albeit often the highest-ranked criterion. For these decisions, using PM4, the outcome (decision) is settled at the latest after the fourth stage or earlier after as many stages as we see the need to complete. On the other hand, some decisions primarily focus on the most cost-effective option, often not the option with the best functional properties, but instead combines reasonably good properties with a low cost. Using PM5, the fifth stage is separate and necessary for this type of decision in order to find the most cost-effective option, no matter how many of the previous four stages we carry out to rank the options functionally. As examples of the latter type of decision, procurements come to mind.

A decision analysis using the PILOT method has two equally important effects. The first is reaching clear and, as far as possible, accurate results from the completed procedure. The results give a good indication of which decision to make. However, we should always bear in mind that decision analysis is the basis for the decision and that a human decision-maker always makes the real decision. The second effect, which is just as important, is the increased awareness and understanding of the options and the entire context for the decision, which is achieved by illuminating the decision problem. Options are what we consider when choosing between what we are going to do. The second parameter is what we consider important, that is, our criteria for choosing. These must be clear and understandable. The effects of these insights are important, not least when a group should make a decision or when investigators prepare a basis for a decision that needs to be communicated to policy-makers or to a management team.

Stage 1 – Pro and Contra Lists

So, how should we approach a decision problem? The first stage is to find out which options are available. Sometimes, it is relatively easy to list these, but sometimes, they are harder to identify. In many cases, creating a process similar to brainstorming is favoured in which creative options of action are produced without the restriction that they must be guaranteed to be realistically feasible. In other cases, there are numerous possible options, but having more than ten in an analysis is rarely advisable. With many more options, it is best to divide the analysis into two phases. In the first phase, representative and particularly attractive options are included for each type or cluster of options. When the first phase is completed, the analysis will indicate one or two most attractive types. Then, in the second phase, more options from these preferred types can be included in a more refined analysis. If such a division cannot be made, then there is no formal reason not to include a large number of options in an analysis with only one phase, but in practice, it might become cumbersome.

Example: Lilly and Larry live with their son Fido and their dog Smilla in a small apartment in the town centre. Fido is soon to start school and needs his own room. For some time, the family has been considering moving from the city centre to a larger apartment in a suburb. However, there are many suburbs, and the choice is not easy. Lilly and Larry have looked at about fifty apartments in the past year but have not been able to decide, and now the start of the school year is approaching rapidly. Above all, buying a larger apartment seems both expensive and a bit scary, but also difficult because there is such a huge selection available and the market is so capricious and nothing quite feels like value for money. They decide to use the PILOT method to determine which apartment to buy.

They begin with Stage 1 by writing down the apartments they have looked at and liked for whatever reason. It turns into a rather long and confounding list, but when they group them by residential- or rather, the type of area, a pattern emerges. After some contemplation, they have identified eight apartments that well characterise what they have looked at during the past year. They write each apartment's address on a piece of paper and begin writing arguments for and against each apartment. The paper is quickly filled with comments like "great floor plan", "afternoon sun", "feels cramped", "close to school", "graffiti by entrance gate", and so on. However, by grouping those into arguments for and against, Lilly and Larry soon see that two of the apartments are out of the question. One is simply too expensive even if it looks great with a terrace and designer kitchen, and the other is so far away that commuting would take unreasonably long. The addresses of the remaining six apartments are A-street no.1, B-alley no.2, C-road no.3, D-crescent no.4, E-avenue no.5, and F-square no.6. The first three seem to be the best at first glance, but they want to continue with Stage 2 of the method with all six options without trying to decide yet. Parts of the lists from the analysis of the first three options are shown in Table 1. The other options are dealt with similarly.

A-street no.1	*B-alley no.2*	*C-road no.3*	…
Pros: Cosy living room Fido has more space Super & small school …	Pros: Fantastic terrace Own study …	Pros: Charming block Good restaurants …	…
Contras: Rather run-down area Far from town but fast access …	Contras: Big anonymous school Far from town and slow access …	Contras: Cramped room for Fido Small balcony facing north …	…

Table 1. The output of Stage 1 of the PILOT method showing the three first options

This procedure can be described more generally. Let us assume that there is a set of lists with an easily manageable number of options, maybe five to ten. We need to develop a pro and contra list (P-C list) for each option. This list includes the advantages and disadvantages we see in each option. The lists may become relatively long, and the same aspect need not be present on all the lists. When the lists are complete, they are inspected for unacceptable drawbacks. An option with any drawback impossible to live with is rejected no matter how attractive other features of that option may be. Mark all such options and eliminate them from the continuation of the method. We now have a possibly purged set of options that we could accept as a good basis for a decision. If we intend to continue with the next stage in the PILOT model, we are now finished with Stage 1.

Finishing after Stage 1

But if we do not want to continue with more stages and rather want to try to make a decision already now, one finishing step remains, which is to inspect the P-C lists. Sort the pros and cons for each option. Lay out the P-C lists with all the options and weigh the advantages against the disadvantages. Try to find options that are entirely worse than at least one other and eliminate them immediately. If there are also options that are worse than doing nothing, then eliminate all of these too. Continue this process until there are only two options left. Now is the time to take a break, and upon returning, try to convince yourself (or the group if this is a group decision) that this particular option should be selected. The option of the two that prevails by that line of reasoning is the one to choose from Stage 1. We have now completed the first analysis. If that seems sufficient, do not continue with Stage 2 or any other stages (except Stage 5 if you use PM5). For example, suppose a large terrace was crucial for Lilly and Larry, but they do not care much about the school, so they select option 2, the apartment at B-alley no.2.

This may seem a bit rough and ready, and if the last part of Stage 1 seems relatively difficult, it is because it often is. This is precisely why four more stages in the PILOT method help find the best option in a decision-making situation. However, sometimes Stage 1 suffices.

Stage 2 - An Argument Matrix

In Stage 1, we produced a set of options and a P-C list with pros and cons for each. We also ensured that no options with unacceptable characteristics remained. In Stage 2, it is time to start thinking about values. Which properties of the possible options are fundamental in this decision situation? Which perspectives on these options are the most important? These perspectives should be grouped under several criteria, each representing any focus on one or more important perspectives on the decision. The P-C lists from the previous stage are often of great help in finding these criteria. It would be strange if the pros and cons of the options did not relate to what we consider to be essential properties of the criteria we will use to make our decision.

Our example again (Stage 2): Lilly and Larry now have six apartments left, each on its own piece of paper listing its respective pros and cons. In Stage 2, it is now time to think about and decide which features and characteristics (criteria) are the most important to their decision. Lilly and Larry begin by writing these down in an unsorted list. The list is growing rapidly: “cosy neighbourhood,” “many cafes,” “lake view,” “good school for Fido,” “open floor plan,” “balcony facing south,” “good state of repair,” “close to work,” “neat indoors,” “easy to park,” and so on. It soon becomes unmanageable and Lilly and Larry try to group the desired characteristics into four main groups and one residual group of miscellany. They find this a little bit tricky, but it also affords clarity to the process to have a grouping as a goal. After some thought, they arrive at the following groupings: Area/Location, Planning/Indoor comfort, School, Commuting/Accessibility, and Miscellaneous. Both Larry and Lilly agree that these criteria embrace the most important aspects of the decision situation while some less important ones need to go into the Miscellaneous category. Further, they feel that this order between the criteria corresponds well to how important they perceive their respective criteria to be. Fido is also allowed to have his say too, but primarily so that he feels included.

The next task for Lilly and Larry is to draw up a matrix (table) with the options (apartments) as rows and characteristics (groups of criteria) as columns. They then fill the boxes with text by picking pros and cons from the P-C lists. Most of the boxes get filled, but after they have gone through and checked off all the lists, a few empty boxes remain. The last thing they do in this stage is to complete the empty boxes by filling in their value assessments there too. Feeling quite satisfied, Lilly and Larry look at the matrix (partially shown in Table 2) to ensure they agree with its contents. They feel they have acquired a much better overview and structure for their decision. They also feel that this could be the basis for their decision but decide to subsequently forge ahead with Stage 3 in the method.

	Area/Location	*Planning/Indoor comfort*	…
A-street no.1	Rather run-down area Far from town but access seems fast …	Cosy living room Plenty of room for Fido …	…
B-alley no.2	Far from town and access seems slow …	Fantastic terrace Own study …	…
…	…	…	…

Table 2. The argument matrix

There are a few things to consider here. Since Lilly and Larry both think that cost is one of the most important criteria, they will use PM5 and defer dealing with cost until Stage 5. In the first four stages, they will consider each option's functional criteria and actual characteristics.

Another important point is that it has long been known that people find it difficult to simultaneously keep more than 5-7 things in their minds. For this reason, but also because in practice only a few criteria dominate most decision-making situations, they will limit themselves to four criteria (or groups of criteria), with an additional miscellaneous group for any remaining criteria, as well as an intuition criterion that we will return to later. Thus, there are six criteria in all. While this is not the method's limit, it is a general recommendation not to exceed six.

So, the purpose of Stage 2 is to find four main criteria for the current decision. If we regard the previous stage as a brainstorming process, then this stage can be regarded as a process where we ask ourselves what we really want. What do we actually value and appreciate about an option that is a candidate solution to our decision problem? Here, in Stage 2, we construct a matrix in which the options form rows and criteria form columns.

Next, we place each argument from the P-C lists in a box (row and column intersection). If an argument does not fit in any of the regular criteria columns, place it in the miscellaneous column. A test that the criteria are properly selected is that most of the arguments from the lists fit into one of the four criteria columns and that few or relatively insignificant arguments end up in the miscellaneous column. When the arguments from the P-C lists are categorised, carry out the following completion measure. One or more boxes in the four criteria columns may be empty, in which case they need to be filled in with how we value the respective options under that criterion. (The miscellaneous column need not be filled in the same way.)

After this procedure, there is a more complete basis for decisions in which all options are valued under each relevant criterion. There is now a matrix (table) with options acceptable as final choices, and which have been assessed using all the criteria. If the intention is to continue with the next stage in the PILOT method, then Stage 2 is now completed.

Finishing after Stage 2

But if you already want to try to make your decision at this stage, then one finishing step remains. This step is to pitch the options of the matrix against each other in a way similar to Stage 1 but with more and better-structured information. That they are already in the form of a matrix makes it considerably easier to find an option worse than all of the others (if there is one) and then eliminate it. If you find several inferior options, eliminate them all in the same way as in Stage 1. Continue this process until you only have two options left. Next, do the same as in Stage 1 and take a break. When you return, try to find convincing arguments supporting that this option should be chosen. The option that clearly wins this challenge is the option you should choose from Stage 2. If Lilly and Larry are beguiled by the living room and are happy that Fido has more space, but care neither about the terrace nor the surroundings, they should choose A-street no.1.

If you also find the last process in this stage relatively tricky that is because it is too, albeit somewhat less. That is why there are three more stages in the PILOT method that help you find the best option.

Stage 3 - Ranking Alternatives

In Stage 1, we produced a set of options and, for each option, a list of pros and cons (the P-C lists). In Stage 2, we continued with value assessments. Each option was valued under the four criteria that we considered most important for the decision situation. The P-C lists support this process, which we documented in matrix form (tabular form) where we reviewed each option against each criterion.

In Stage 3, we will rank all the options within each criterion separately. Usually, an option we consider the best under one criterion is not the best under all other criteria. If any option were the best under all criteria, the decision would be easy, but this is rarely the case. And in those rare cases, the best option is usually obvious without us needing to conduct any decision analysis at all.

Our example continues (Stage 3): Lilly and Larry were pretty drained after the two initial stages, which entailed a considerable effort when they needed to find a complete set of pros and cons for each of the six apartments. During the coffee break before they started with Stage 3, they speculated which of the apartments would probably turn out to be best when they were finished with the functional analysis. It is important to remember that this yet only includes functions and properties, not costs, since they use PM5. They concluded that C-road no.3 and E-avenue no.5 would probably lead, but it was impossible to say which of them had the advantage. They intuitively ranked the two options equally. They guessed A-street no.1 as the next one, followed by D-crescent no.4 and F-square no.6, with B-alley no.2 last. This order was just their gut feeling ensuing the first two stages once they had familiarised themselves in depth with their options.

Before we begin to rank the alternatives, we should, therefore, try to make use of this kind of subconscious information. Sometimes, it is not easy to completely describe a decision situation with a set of regular criteria. Even if you are relatively satisfied with the descriptions in the argument matrix in Stage 2, there may be a sense that something is missing. Sometimes there is this sense, but sometimes there is not. This will be different for people with different levels of awareness of their thought processes and may also be different depending on the decision situation. The PILOT method is an opportunity to ensure that all such information is exploited. You can choose to use this opportunity but it is not required. Anyone who thinks this sounds vague or does not feel comfortable with it can skip this step. Others should do the following: try to construct an overall ranking of the alternatives based on your gut feeling – what you think or guess the outcome of the functional analysis of the decision will turn out to be. This ranking is called the intuition criterion.

Assuming that various options are best under different criteria, we must rank them for each criterion. We can construct a hierarchy by studying our evaluations from Stage 2 in the matrix one column at a time. In this hierarchy, we expect to decide which options are better than which others, but a draw is also permitted and indicated by two or more options being ranked with the same placement in the order. After each criterion has been treated separately, you will have four rankings, one for each of the four criteria, plus the miscellaneous criterion.

Our example again: Lilly and Larry have created a matrix (table) describing how they value their six prospective apartments under the six criteria relevant to this decision. Now it is time to look at each criterion separately and rate the options accordingly. They begin with area/location, the six alternative apartments are located in different areas and with different locations in these areas. Some are more centrally located, others are closer to the water. Still others are closer to the socially significant presence of cafes, restaurants, cinemas, and so on. After some discussion, they succeed in ranking the apartments. They rate C-road no.3 best in terms of Area/Location followed by E-avenue no.5 and F-square no.6, followed by the other three apartments ranked in decreasing attractiveness. Then they do the same with each of the other three criteria: Planning/Indoor comfort, School, and Commuting/Accessibility. The same apartment will not lead in all criteria. For example, C-road no.3 is the penultimate for Floor plan but is first for Area and location. Finally, they rank the remaining factors which did not fall under the four main criteria. At this stage, Lilly and Larry feel that their criteria have become stable clusters of aspects and they rename them accordingly as Neighbourhood, Floor plan, School, and Travel respectively.

Now it is time to score the rankings of the alternatives. This is entirely mechanical and involves no opinion or consideration. All that is needed is pen and paper or an Excel spreadsheet that is produced in a matter of minutes. In the matrix that contains the option rows and criteria columns, points are awarded systematically so that under each criterion the worst option gets one point, the second worst two points and so on up to the best option.

Back to our example: Draws between options are allowed, but for Lilly and Larry there are no draws in which two options are ranked the same except under the intuitive criterion. You can see their rankings in Table 3. As a very preliminary result, the options' scores are summed across the rows, which puts C-road in first place followed by E-avenue and A-street. This summary does not consider how the criteria are of different importance, but rather considers all aspects as equally important. This is not something that Lilly and Larry actually agree on. The fact that no option is best under all criteria, but rather that options, so to speak, cross over under the different criteria, means that Lilly and Larry decide to proceed to the next stage in the method. Thus, Table 3 shows the intermediate result of this stage. Note that rows and columns switch places compared to Stage 2 since this stage is a numerical one.

Stage 3	*Neigh-bourhood*	*Floor plan*	*School*	*Travel*	*Misc.*	*Intuition*	*Result*
A-street no.1	1	4	6	3	4	3	21
B-alley no.2	3	6	1	1	2	1	14
C-road no.3	6	2	4	4	6	4	26
D-crescent no.4	2	5	3	2	3	2	17
E-avenue no.5	5	3	5	6	1	4	24
F-square no.6	4	1	2	5	5	2	19

Table 3. Stage 3 in the PILOT method – ranking alternatives

More generally, in Stage 3, the lowest ranked item under each criterion receives 1 point, the second-lowest ranking receives 2 points, and so on, up to the highest ranking, which receives as many points as there are alternatives. An exception is if two or more options are ranked equally under any criterion. Then they get the same score but a higher ranking will still only get one point more than the option ranked immediately below. You will now have a column with points for all options under each criterion. However, note here that we have not taken into account how important the criteria are. If you intend to do this by continuing with Stage 4 in the PILOT method, you are now finished with Stage 3.

Finishing after Stage 3

But if you already want to try to make a decision in Stage 3, then one finishing step remains. This step sums the options by row. We summarise each option's scores across all criteria and obtain a total. The option with the highest total is the option that the PILOT method indicates as the best, but since we are only at Stage 3, it is good to take the results with a pinch of salt. At least retain the two or three best options and try to reason which option is preferable using a procedure similar to the previous two stages. Remember, so far we have ranked only the alternatives, not the criteria. Ranking the criteria, which comes next, is an important component of the PILOT method. However, Strata Decisions sold an MCDM decision-making software tool that was bought and used by over 1000 hospitals in the US. It contained a number of criteria that were by default set at equal weights (i.e. in essence no conscious weighting) and supposed to be altered by each hospital individually according to their particular preferences and priorities. In reality, it turned out that less than 10 of those over 1000 hospitals actually changed the weights at all (in our terminology, went beyond stage 3) and declared themselves satisfied with the decision support they had received that far. This is not to say that you should not move on to the next stage if your decision is not yet finalised, you definitely should, but rather that there are considerable knowledge gains at every stage of the PILOT method.

Stage 4 - Ranking the Criteria

In Stage 1, we produced a set of options and developed a list of pros and cons for each option. In Stage 2, we continued by assessing the options. Each option was valuated under the four criteria we selected for the current decision situation. All this was documented in the form of a matrix in which we ensured that each option was judged under each criterion. In Stage 3, we then ranked the alternatives within each criterion so that we had as many rankings as we had criteria.

The result from Stage 3 was a scored matrix where each option under each criterion has a score that indicates exactly how this option has been ranked under the current criterion. A higher score indicates that the item is ranked higher, while a score of one point indicates that the option is ranked last of all the options under this particular criterion. However, the summation made in Stage 3 did not take into account that some criteria are more important than others. Therefore, time is nigh for ranking the criteria, not unlike the procedure we did for the options in Stage 3.

Our example again (Stage 4): Lilly and Larry have now done most of the work evaluating their options under their selected criteria. As we have seen, both of them felt that they had listed the criteria roughly in their order of importance: Neighbourhood, Floor plan, School, Travel and Miscellaneous. However, when using the PILOT method, they need to decide exactly what their thoughts are about the criteria *in the current decision situation*. They must decide how important the different criteria are *in this particular case*.

When Larry and Lilly look at the six options, they feel that their locations are actually all quite ok. Although there are differences, they are not extremely large. The same goes for the floor plans and indoor comfort. They realise that within their price range, they will have neither a big living room nor a recently modernised kitchen, so the differences are not so great between the options they have selected and are currently considering. However, the schools in different areas clearly differ substantially, and both Lilly and Larry are keen that Fido will get a good education throughout elementary school. Larry works as an IT consultant, so he is periodically leased out to customers that can be virtually anywhere in the city. If the apartment they choose is too far from the beaten track, Larry risks having to make some very long commuting trips, and when they look again, they realise that this is an essential difference between the various options.

After rethinking this, Lilly and Larry realise that the difference between the best and worst options for the School criterion is the most important in this particular decision situation followed by Travel, Neighbourhood, and Floor plan in that order. Again, this order does not mean that Floor plan is less important than School in any absolute sense, only that Lilly and Larry have taken a stand specific to the current situation. Thus, their real criteria ranking differs markedly from what they initially thought it would be. They had not realised that such a ranking must be relative to the options – it cannot be absolute in any sense.

It is important to note two crucial differences compared to Stage 3. First, criteria are ranked only once, not numerous rankings as with the options. Second, this ranking is relative, which is a very important point, formally called the scale/weight duality. The statement "criterion A is more important than criterion B" is irrelevant in this form because we do not know what options are available under these criteria. Suppose someone says that for computer hard-disk drives that "price is more important than storage capacity". But if the prices of three disks under consideration are $50, $55, and $60 with storage capacities 1000 GB, 2000 GB, and 3000 GB, the decision is completely different than if prices were $50, $70, and $90 for hard drives with storage capacities 1300 GB, 1400 GB, and 1500 GB. Basically, no matter how we weigh price in relation to capacity, we choose the last hard drive in the first of these two examples and the first one in the second. So the key is to *rank the criteria according to the differences between the best and worst options in each criterion.*

In the first example, only $10 distinguishes 2000 GB of storage capacity and in the second $40 distinguishes 200 GB. It is these differences we must pitch against each other, not the absolute values themselves. "Price is more important than storage capacity" is therefore insufficient information to proceed with in a decision analysis. Such a statement will lead you completely astray. When we have to rank the criteria, it is hence important to rank the respective ranges between the best and worst options under each criterion. It is exactly here that many decision-makers fail, so this stage deserves to be taken very seriously. This also entails that should you go back to Stage 3 at any point and change the rankings of options under one or more criteria, the criteria ranking in this stage must subsequently be revisited.

Our main example again: Lilly and Larry have agreed on the ranking of criteria: School, Travel, Neighbourhood, and Floor plan in that order based on the actual differences between their available options, not based on any absolute truth or order regardless of the options, simply because no such truth can exist. Therefore they assign weights as follows: Floor plan one point, Neighbourhood two points, Travel three points, and finally four points to School. Then they multiply the options' points with their respective weights and sum for each option, see Table 4. In this way, they gain an overall score for each option and that score is their complete evaluation of each option. The highest score thus indicates the option that Lilly and Larry should prefer if they had a free choice, i.e. if there were no costs involved. In Table 4, we can see that E-avenue no.5 and C-street no.3 have changed places compared with Stage 3.

Relative weights	2	1	4	3	1	0	
Stage 4	*Neighbourhood*	*Floor plan*	*School*	*Travel*	*Misc.*	*Intuition*	*Result*
A-road no.1	2	4	24	9	4	0	43
B-alley no.2	6	6	4	3	2	0	21
C-street no.3	12	2	16	12	6	0	48
D-crescent no.4	4	5	12	6	3	0	30
E-avenue no.5	10	3	20	18	1	0	52
F-square no.6	8	1	8	15	5	0	37

Table 4. Ranking criteria without the intuition criterion

In general, after ranking the criteria it is time to score them (in the previous stage we assigned points to options, not criteria). The least important of the criteria receives weight one; the next, weight two; up to the most important, which receives the highest weight. If two criteria are deemed equally important, assign the same weight as we did with points for the options in the previous stage. Once the criteria are assigned weights, sum up each option's total score as we did in Stage 3. But before summing, each option's score in the table is multiplied by the weight that each criterion received. If you have a Miscellaneous criterion in which you have a number of smaller aspects that you still want to include in the analysis, assign the weight one to Miscellaneous, otherwise assign zero. For example, the score for A-road no.1 is calculated as

$$2 \cdot 1 + 1 \cdot 4 + 4 \cdot 6 + 3 \cdot 3 + 1 \cdot 4 + 0 \cdot 3 = 2 + 4 + 24 + 9 + 4 + 0 = 43.$$

In doing this, something remarkable happens. From many years of research and development of decision methods, including algorithm development and simulations of all kinds of decision situations, as well as numerous real-life decision analyses, this relatively simple rating method just explained imposes a strongly discriminatory (decisive) effect on the decision analysis. One could think that specifying exact percentages for weights would be important or identifying them more precisely should be. But a straight ranking order has proved to have properties that are close to as good – with a lot less effort. As we discussed earlier, it is generally very difficult or even impossible to give such weights with any real precision, and in such cases ranking proves to be the superior method for indicating the importance of various criteria.

The total score for each option in this process is the final ranking of the functional quality and capacity of the options being considered in the decision situation if you follow PM5 and of the entire option for PM4. Stage 4 is hereby completed, and if we follow PM4 or if there is no cost component in the analysis, then we have reached a final decision. Otherwise, we need to proceed to Stage 5.

Finishing after Stage 4

But before we do, those who took the opportunity to set up an intuition criterion may use it now, simply by comparing its ranking with that resulting from Stage 4 in which the intuition criterion was assigned the weight zero, as in the example with Larry and Lilly in Table 4. If the rankings are consistent or almost consistent with each other, there seems to be no significant difference between the gut feeling and the formal results of the analysis. If the rankings are not consistent, there is subliminal information that partly contradicts the analysis results. Such a discrepancy does not mean that the analysis is wrong. Either the conception represented by gut feeling is misplaced, which is common, or it indicates that some criterion has been overlooked or that an option has been badly ranked. The analysis should then go back to Stage 1 or Stage 2 to see if there is any reason to re-evaluate the work of those stages. But before going back, it is advisable to check the size of the deviation. This is done by increasing the weight of the intuition criterion in increments of one until reaching five. At weight five, gut feeling weighs more than the main criterion and if the analysis still has not flipped to the expected result then we can say with great certainty that our gut feeling is playing tricks on us.

Back to our example: Lilly's and Larry's intuitive ranking corresponded fairly well with the formal analysis, but there were some small differences with options that they could not distinguish and yet which clearly differed in the analysis. Because they believe that buying an apartment is a decision that should be both close to optimal and also feel right, they choose in Table 5 to include their intuition criterion and assign it a weight of one, thus including it in the result of the stage.

Relative weights	2	1	4	3	1	1	
Stage 4	*Neighbourhood*	*Floor plan*	*School*	*Travel*	*Misc.*	*Intuition*	*Result*
A-road no.1	2	4	24	9	4	3	46
B-alley no.2	6	6	4	3	2	1	22
C-street no.3	12	2	16	12	6	4	52
D-crescent no.4	4	5	12	6	3	2	32
E-avenue no.5	10	3	20	18	1	4	56
F-square no.6	8	1	8	15	5	2	39

Table 5. Ranking criteria with the intuition criterion activated

Stage 5 – Separate Cost Analysis

In its larger PM5 form, the PILOT method consists of five stages, so this is the last one. In Stages 1 and 2, we produced a set of options and evaluated each of them. Each option was evaluated under the four criteria selected for the decision situation. Then in Stage 3, options were ranked under each criterion and in Stage 4 criteria were weighted in relation to each other. The total score each option received in Stage 4 was the final ranking of the functional quality of the options. This leaves only the matter of cost to analyse in Stage 5.

Returning to our example: Lilly and Larry have conducted an analysis of six apartments according to the previous stage. For each option, they have calculated a monthly expense based on the monthly fee plus interest on the loans they would need to take. They calculate using a fixed rate for the next few years in order to obtain a secure budget. The cost per month for the six options is shown in Table 6.

Stage 5	*Cost/month*	*Cost increase*	*Score difference*	*Dominance*

~~A-road no.1~~	1590	310	+24	dominated	
B-alley no.2	1280	0	0		← base case
C-street no.3	1550	270	+30		
~~D-crescent no.4~~	1490	210	+10	dominated	
E-avenue no.5	1710	430	+34		
F-square no.6	1430	150	+17		

Table 6 Trading cost and functionality

Lilly and Larry want to keep costs down, so they start with the least expensive option. They take that option as the basis of their analysis and thus make B-alley no.2 their so-called base case. In two columns in Table 6, they then work out how much more than the base case each option will cost and how many more points these have. It is possible that an option can have fewer points than the base case, which would yield a negative point difference, but this is not so in their case. Before they begin with the monetary analysis, they look to see whether any options are dominated, that is, any options that score lower in points for a higher cost than some other option. They see fairly quickly that A-road no.1 is both more expensive than, and inferior to, C-street no.3 and likewise that D-crescent no.4 is both more expensive than, and inferior to, F-square no.6. Two options can therefore be rejected before the analysis in this stage has even begun.

Starting with the base case B-alley no.2, Lilly and Larry now analyse what they can get for their money if they decide to invest more. They look at the options in ascending order of cost. For $150/month more than the base scenario, they can live on F-square no.6, which is an increase of 17 functional points. When they pitch these two options against each other, F-square seems much more interesting and worth the difference in cost, so they decide to keep it and reject the base case B-alley no.2. Next, they compare F-square no.6 to the cheapest remaining option that has not yet been rejected, which is C-street no.3. For a further increase of $120/month, a total of $270 above the (rejected) base case, they get an apartment they found to have 30 more points, a further increase of 13 functional points. Here, Lilly is a little hesitant, but Larry is more positive. After a discussion, they find that even this improvement is worth taking, so C-street becomes their new choice. There is now only one option left to consider, E-avenue no.5. At $160/month more than C-street, and a total of

$430/month above the base case, they can acquire an apartment they valued 34 points better than the base case, a meagre 4 points more than C-street. That difference between the two options does not seem that great to them, and altogether C-street appears to have the most value for money.

The decision really could only have turned out two ways during this analysis. Both of them were convinced that F-square had considerably more value for money than B-alley, but maybe Lilly could have convinced Larry that they should not have chosen C-street. Both were in complete agreement that they gained a superior overview of the decision situation by using the PILOT method, and that with good factual arguments, it was relatively easy to arrive at the evaluation shown in Table 6 and then at a decision that they were both happy with and feel they understood.

To be a little more general, a mechanism in classical cost-benefit analysis is that costs and benefits are pitched against each other while seeking the greatest possible differences on the benefit side. And the idea behind ABC analysis in decision contexts (Analysis of Benefits and Costs) is the same. You look for the option that provides the greatest difference between functionality and costs, i.e. what colloquially would be called the most value for money. We highlighted the functionality/benefits in the four previous stages of the method and they are indicated by the sum of the points from Stage 4. In order to compare the functionality benefits to the costs, we need to find a way to compare points to monetary terms. After having rejected options that are dominated, i.e. they have a higher cost and worse functionality score than one other option, the following procedure should be followed:

- We cannot reduce the cost below the least expensive option. Therefore, we use that as the base case for the procedure. Let the cost of the least expensive option be M dollars.
- For each option, the increased cost is now calculated. If the cost of an alternative is P dollars, then the increased cost will be P–M dollars. This difference should be set against the corresponding difference for each option in the final functionality score in Stage 4 compared with the score of the base scenario.
- For each such pair-wise comparison, the most cost-effective and affordable option is retained and the other rejected.

- This procedure is repeated for each option that is still not rejected from the lowest to the highest cost.
- When all the options but one have been rejected, only the most cost-effective and affordable option remains, and that is the one to choose.

These bullet points device an approximate procedure because they contain subjective estimates of characteristics that are not easily quantifiable, but this is the method's strength rather than its weakness. It is impossible to accurately determine an objective estimate of functional points. Even if it were possible, such a procedure would take an unreasonably long time to carry out and require considerable resources. Instead, the points should be seen as stable and good indicators for the stated preferences and values, so that when Stage 5 finally pitches costs against features and functions, it does so using real costs against real functionality for the options.

Our example for the last time: After having used the PILOT method, Lilly and Larry decide to select the apartment on C-street. The following week, they sign the contract for the apartment on the top floor of C-street no.3. Then they live happily ever after and their son Fido has a wonderful childhood and in due course a great career as a management consultant. Furthermore, their dog Smilla now avoids the crowded parks in the town centre and has access to large recreation areas.

The PILOT method is an iterative method even if only one cycle has been described in this document. When new information arrives or the set of options change, the stages should be revisited accordingly. Feel free to iterate back to any stage at any time but remember that once a stage is revisited, all ensuing stages must be revisited in sequence before the new iteration is completed. This is especially important for the criteria ranking in Stage 4.

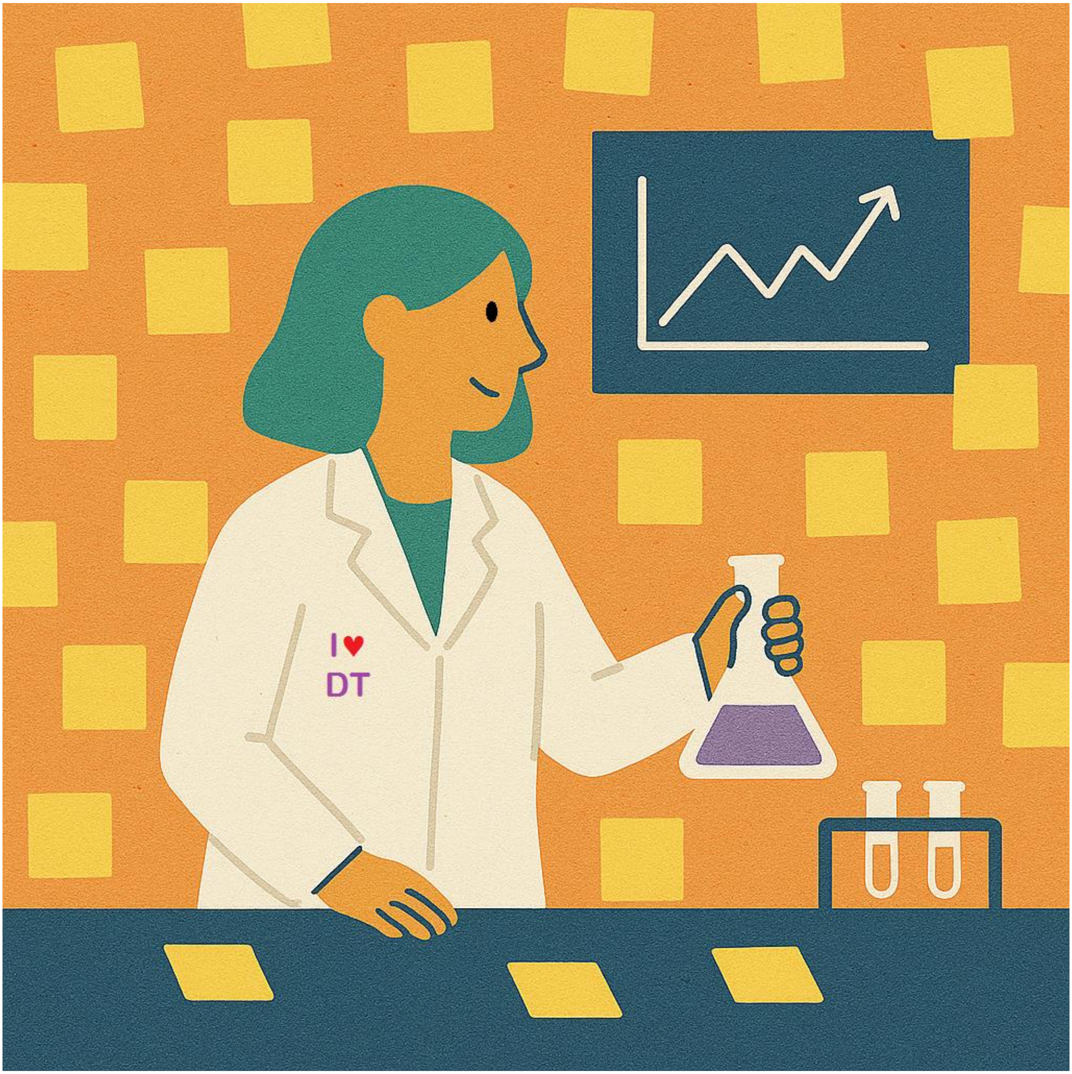
Summary

The PILOT method is one of the easiest decision methods that is still powerful enough, not least procedurally speaking but also in terms of obtaining stable and transparent results. It is the culmination of many years of research and development that has shifted from very powerful but also very complex methods to progressively simpler ones, both in terms of user interfaces and calculations, but without losing too much decision power. The PILOT method has the obvious advantage of being computable by hand or very easily modelled in a computer spreadsheet. More complex methods require either advanced computational software or complex spreadsheet modelling. Thus, one can say that the PILOT method is the most reasonable combination of simplicity and decision-analytic power available.

Note that decision analyses such as the PILOT method can be used in two diametrically opposed ways.

- You can perform a forward analysis based on information about a number of options and their properties and try to arrive at a total ranking where one or possibly several options appear to be the most advantageous.
- Or an inverse analysis can be carried out, where the goal is to make sure that an agreed-upon decision is good enough. The inverse analysis can involve adjusting weights or other judgement values in order for the results to match and make sense.

At first glance, an inverse analysis appears a bit fraudulent since there is a danger that the parameters are adjusted to values that they would not have had in a forward analysis based on known facts. However, since the purpose of an inverted analysis is quite different, we should see the work from a different perspective. The idea here is to try to understand what led to a decision, regardless of whether that means certain parameters take on values that do not appear to be consistent with the worldview that prevails at the time of the analysis; or that of the analyst, or more generally, of the decision-makers and their environment. Through an inverse analysis, we can find out how and what is valued, in a way that in retrospect does not seem to be optimal, or if there simply are real disagreements about certain data inputs. This can be of great value, but it should not be confused with the process in a forward ('normal') analysis. Both analytical methods are supported by the PILOT method and they both work in a completely analogous way. However, in this chapter, we have focused on the more commonly used forward analysis.



Personal innovators use many tools to attain their goals

About the Author

Mats Danielson is a Full Professor in Computer and Systems Sciences and the Vice President of External Relations, Innovation and Information Technology at Stockholm University. He is also a former Dean of the Faculty of Social Sciences. He has a PhD in Computer and Systems Sciences from KTH Royal Institute of Technology as well as university degrees in Computer Science and Engineering (from KTH) and in Economics and Business Administration (from Stockholm University). He was working as an IT and management consultant for more than 20 years before joining academia.

In academia, he has worked with innovations in several capacities, from being the deputy chairman of Stockholm University Holding and Incubator to being the course director and one of the teachers at Openlab Stockholm's innovation courses at both Master's and PhD levels. Both courses use Design Thinking as a primary toolbox for challenge-driven innovations. The Master's course is a full-semester 20-week half-time course involving student teams of 6-8 persons, while the PhD course is shorter and more individually focused on the respective thesis topics.[1]

[1] Visit www.openlabsthlm.se for more information.

Books from Sine Metu in the Computer Science series

Transcending Business Intelligence

K. Borking, M. Danielson, G. Davies, L. Ekenberg, J. Idefeldt, A. Larsson
ISBN 978-91-978-4505-2

A Decision-Analytic Manifesto

L. Ekenberg, M. Danielson
ISBN 978-91-527-5306-4

A Little Book on Error-Free Software

M. Danielson
ISBN 978-91-531-0456-8

DECISION
RIGHT
WRONG

This is not a book about Design Thinking as you know it. The book invites you to use powerful design tools to rethink your everyday work challenges as well as your own life. You will learn how to approach problems with clarity, navigate uncertainty with creativity, and instigate meaningful change, whether you are mapping out a career shift, improving a work process, or simply looking for better ways to move forward. Smart, accessible and grounded in real-world practices, this book is for anyone ready to think more deeply, act more creatively, and design life with intention.

Sine Metu