

# System Development Methods

Erik Perjons

# Questions to answer

- What is a system development method?
- Why do you need system development methods?
- Which different types of system development methods exists and which are their benefits and drawbacks?

# System Development Methods

# System Development is Complex

- System development, sometimes called software development, is complex
- **Many system development projects fail** (around 25 percent), **others are delayed or do not produce the required functionality** (around 45 percent) – mainly due to this complexity of the projects

# System Development is Complex

Factors that makes system development complex are:

- **System development project are large project** with many different people involved from different groups, departments and organizations. These people have different roles, and thereby, different views and expectations
- Many times there are **shortcomings in requirement engineering**, that is, requirements are lacking and/or have low quality. **Requirements can also change during the projects**
- The system to be developed also needs to **be integrated with business processes and other IT systems** in the IT system architecture of an organization, and this is a complex process

# What is a System Developments Method?

- A **system development method** is method that provide structure to the development of a system
- A system development method does often contain the following phases/activities:
  - **Requirement engineering** – define the requirement of the system (this phase/activity is sometimes called **analysis**)
  - **Design** – define the overall architecture and structure of the system
  - **Implementation** – code the functions and build the database system
  - **Test and deploy** – test that the implemented system fulfill the requirements, and introduce the system in the organisation and integrate it with processes and other systems

# Why System Developments Methods?

The **goals of system development methods** are:

- to provide a **better control of the project**, such as better resource allocation and management of delays
- to ensure that **requirements are defined and fulfilled**
- to **communicate information about the project throughout the organization – to all stakeholders** (i.e. people and roles that has interest in the system, such as future users, developers, managements, owners)
- to **produce necessary documentation for maintaining the system after it has been deployed**

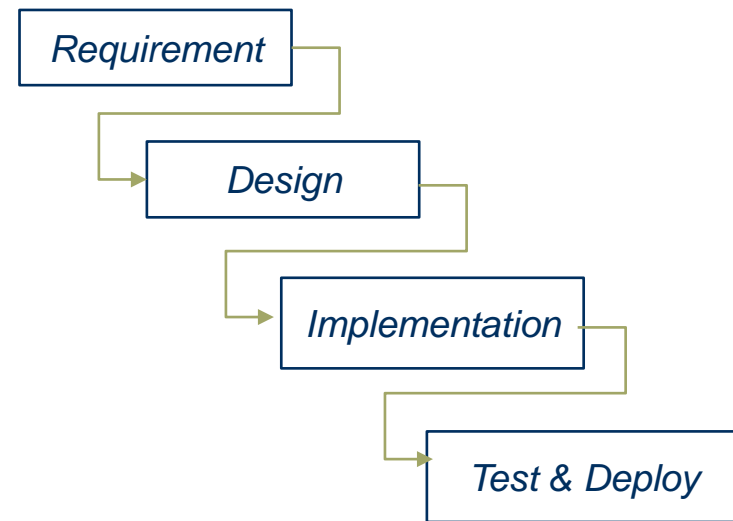
# Categorizing System Development Methods

- System development methods can be categorized into methods based on the **waterfall principle** or the **iterative principle**
- **Example of methods based on waterfall principle** – DOD-STD-2167A , MIL-STD-1521, both are standards from the military domain
- **Example of methods based on iterative principle** – Rational Unified Process (RUP), XP, Scrum



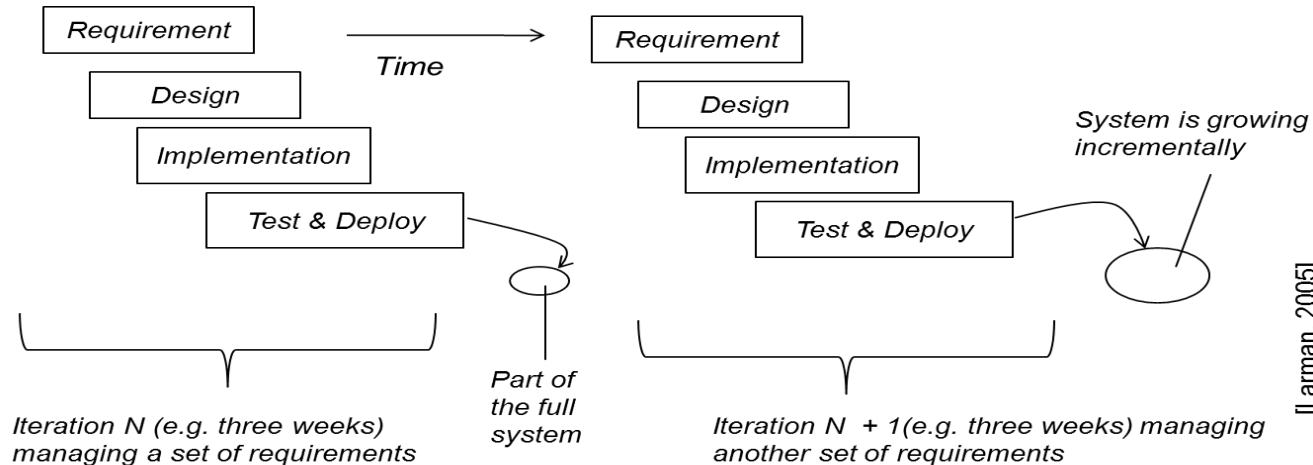
# Waterfall Principle

- **Waterfall principle** - is based on the idea that **all requirements should be defined before the overall design of the whole system is started**. Moreover, the design, in its turn, needs to be finished before implementation/coding is started. Finally, first when the coding is finished, the system can be tested and deployed in the organization. That is, it is strictly sequential.



# Iteration Principle

- **Iteration principle** is based on the idea that a system should be developed in a set of mini project. Each miniproject, called an iteration, involve a limited set of requirements, which are designed, implemented and deployed to receive early user feedback



# Waterfall Principle Evaluated

- **Benefit of the Waterfall principle:**
  - easy to plan and estimate costs early in the project
  - the plan is easy to understand, and control/follow up
- **Drawback of Waterfall principle:**
  - many systems developed using the waterfall principle are not successful, mainly due to limited customer feedback during the project
  - **problem to handle changed requirements** during development
  - often results in extensive production of documents that are not used (i.e. easily become bureaucratic)

# Iteration Principle Evaluated

- **Benefit of the Iteration principle:**
  - **more successful results** than projects based on the waterfall principle
  - the **project receive early and rapid feedback from user**
  - **support replanning based on user feedback** and **changed requirements during the project**
- **Drawback of Iteration principle:**
  - hard to plan and estimate costs early
  - hard to introduce iterations and early user feedback in practice

# Incremental/Evolutionary Development

- Closely related to the **interaction principle** are incremental and evolutionary development
- **Incremental development** – the system is growing (in number of functions) step-by-step
- **Evolutionary development** – the system is developed based on feedback, that is, the system is adapted given the feedback

# Prototype

- Before developing a system, a prototype can be developed.
- There is no agreement on what a prototype is, but it can be seen as an early form of the final system, focusing on certain aspects of it. For example, a prototype of a system can be a drawing on a paper, or a software system focusing on the system's user friendliness.
- The prototype enables the designers to better understand design challenges for developing the final artefact, and the prototype is often constructed to obtain early user-feedback.

# Agile Methods

- **Agile methods** are a **group of iterative system development methods** that share a set of **values** and **principles** that should govern the system development – presented in the Agile Manifesto
- Agiles methods also **provide a set of practices** to use for supporting the values and principles, for example pair programming and time boxing
- **Most well-known agile methods are:** eXtreme Programming (XP), Scrum, Crystal, Lean Software Development

# Agile Values

- “Individuals and interactions over processes and tools”
- “Working software over comprehensive documentation”
- “Customer collaboration over contract negotiation”
- “Responding to change over following a plan”

[Citations from Manifesto of Agile Software Development (<http://agileManifesto.org>)]



# Agile Principles - Examples

- “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software”
- “Working software is the primary measure of progress”
- “Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done”
- “At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly”

[Citations from Manifesto of Agile Software Development (<http://agileManifesto.org>)]

# Agile Practices – Examples

- **Pair programming** – means that two programmers work together during the coding of the system. The first one, called the driver, carries out the coding, while the other, the observer, gives advices both regarding both the strategy and details
- **Time boxing** – means that the time frame for each each iteration is specified early in the project and should not be changed. The time frames for each iteration force the development team to really develop the system incrementally using iterations

# Agile Methods Evaluated

- **Benefits of Agile methods:**

- popular among system developers
- very adaptive to new requirements – support constant replanning
- the bureaucracy is limited – instead focus on individuals and their collaboration

- **Drawbacks of Agile methods:**

- provide limited governance support in large project
- provide insufficient risk management

# Rational Unified Process

- **Rational Unified Process (RUP)** - also often called Unified Process
- RUP is iterative and use case driven
- RUP provides a loose structure of activities and instruments (including a number of UML models) that can be used
- RUP needs to be configured for each projects, that is, it need to be specified which activities and instrument to be used and combined

# RUP Evaluated

- **Benefits of RUP:**

- can be configured to suit a specific project
- better in managing risk than agile methods
- is use case driven which supports iterations

- **Drawbacks of RUP:**

- RUP is often criticized for being too bureaucratic if too many activities and instruments are used when configuring RUP for a project

# Questions to answer

- What is a system development method?
- Why do you need system development methods?
- Which different types of system development methods exists and which are their benefits and drawbacks?