What is quality?

- Quality, simplistically, means that a product should meet its specification.
- This is problematical for software systems
  - There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.)
  - Some quality requirements are difficult to specify in an unambiguous way
  - Software specifications are usually incomplete and often inconsistent

The quality compromise

- We cannot wait for specifications to improve before paying attention to quality management.
- We must put quality management procedures into place to improve quality in spite of imperfect specification.
Scope of quality management

- Quality management is particularly important for large, complex systems.
- The quality documentation is a record of progress and supports continuity of development as the development team changes.
- For smaller systems, quality management needs less documentation and should focus on establishing a quality culture.

Quality management activities

- Quality assurance
  - Establish organisational procedures and standards for quality.
- Quality planning
  - Select applicable procedures and standards for a particular project and modify these as required.
- Quality control
  - Ensure that procedures and standards are followed by the software development team.
  - Quality management should be separate from project management to ensure independence.

Process-based quality

- More complex relation for software because:
  - The application of individual skills and experience is particularly important in software development.
  - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality.
- Care must be taken not to impose inappropriate process standards - these could reduce rather than improve the product quality.
**Principal product quality factors**

- Development technology
- Process quality
- Product quality
- People quality
- Cost, time & schedule

**Practical process quality**

- Define process standards such as how reviews should be conducted, configuration management, etc.
- Monitor the development process to ensure that standards are being followed
- Report on the process to project management and software procurer

**Quality assurance and standards**

- Standards are the key to effective quality management
  - **Product standards** define characteristics that all components should exhibit e.g. a common programming style
  - **Process standards** define how the software process should be enacted

**Product and process standards**

<table>
<thead>
<tr>
<th>Product standards</th>
<th>Process standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design review form</td>
<td>Design review conduct</td>
</tr>
<tr>
<td>Requirements document structure</td>
<td>Submission of documents to CM</td>
</tr>
<tr>
<td>Method header format</td>
<td>Version release process</td>
</tr>
<tr>
<td>Java programming style</td>
<td>Project plan approval process</td>
</tr>
<tr>
<td>Project plan format</td>
<td>Change control process</td>
</tr>
<tr>
<td>Change request form</td>
<td>Test recording process</td>
</tr>
</tbody>
</table>
Standards development

- Involve practitioners in development. Engineers should understand the rationale underlying a standard.
- Review standards and their usage regularly -- standards can quickly become outdated and this reduces their credibility amongst practitioners.
- Detailed standards should have associated tool support.

ISO 9000

- An international set of standards for quality management.
- Applicable to a range of organisations from manufacturing to service industries.
- ISO 9001 applicable to organisations which design, develop and maintain products.
- ISO 9001 is a generic model of the quality process that must be instantiated for each organisation using the standard.

ISO 9000 certification

- Quality standards and procedures should be documented in an organisational quality manual.
- An external body may certify that an organisation’s quality manual conforms to ISO 9000 standards.
- Some customers require suppliers to be ISO 9000 certified although the need for flexibility here is increasingly recognised.

ISO 9000 and quality management

![Diagram showing ISO 9000 and quality management process]

- ISO 9000 quality models
- Organisation quality manual
- Project 1 quality plan
- Project 2 quality plan
- Project 3 quality plan
- Project quality management
- Supports
Documentation standards

- Particularly important - documents are the tangible manifestation of the software
- Documentation process standards
  - Concerned with how documents should be developed, validated and maintained
- Document standards
  - Concerned with document contents, structure, and appearance
- Document interchange standards
  - Concerned with the compatibility of electronic documents

Document standards

- Document identification standards
  - How documents are uniquely identified
- Document structure standards
  - Standard structure for project documents
- Document presentation standards
  - Define fonts and styles, use of logos, etc.
- Document update standards
  - Define how changes from previous versions are reflected in a document

Quality planning

- A quality plan sets out the desired product qualities and how these are assessed and defines the most significant quality attributes
- The quality plan should define the quality assessment process
- It should set out which organisational standards should be applied and, where necessary, define new standards to be used
Software quality attributes

<table>
<thead>
<tr>
<th>Safety</th>
<th>Understandability</th>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Testability</td>
<td>Usability</td>
</tr>
<tr>
<td>Reliability</td>
<td>Adaptability</td>
<td>Reusability</td>
</tr>
<tr>
<td>Resilience</td>
<td>Modularity</td>
<td>Efficiency</td>
</tr>
<tr>
<td>Robustness</td>
<td>Complexity</td>
<td>Learnability</td>
</tr>
</tbody>
</table>

Quality reviews

- This is the principal method of validating the quality of a process or of a product
- A group examines part or all of a process or system and its documentation to find potential problems
- There are different types of review with different objectives
  - Inspections for defect removal (product)
  - Reviews for progress assessment (product and process)
  - Quality reviews (product and standards)

Software measurement and metrics

- Software measurement is concerned with deriving a numeric value for an attribute of a software product or process
- This allows for objective comparisons between techniques and processes

Software metric

- Any type of measurement which relates to a software system, process or related documentation
  - Lines of code in a program, the Fog index, number of person-days required to develop a component.
- Allow the software and the software process to be quantified
- May be used to predict product attributes or to control the software process
- Product metrics can be used for general predictions or to identify anomalous components

Covered on the course TEME/ID2003
**Internal and external attributes**

- Maintainability
- Reliability
- Portability
- Usability

**Number of procedure parameters**

- Cyclomatic complexity
- Program size in lines of code
- Number of error messages
- Length of user manual

**The measurement process**

- A software measurement process may be part of a quality control process
- Data collected during this process should be maintained as an organisational resource
- Once a measurement database has been established, comparisons across projects become possible

**Data accuracy**

- Don't collect unnecessary data
  - The questions to be answered should be decided in advance and the required data identified
- Tell people why the data is being collected
  - It should not be part of personnel evaluation
- Don't rely on memory
  - Collect data when it is generated not after a project has finished

**Measurement analysis**

- It is not always obvious what data means
- Analysing collected data is very difficult
- Professional statisticians should be consulted if available
- Data analysis must take local circumstances into account
Process improvement

- Understanding existing processes and introducing process changes to:
  - improve product quality
  - reduce costs
  - accelerate schedules
- Most process improvement work so far has focused on defect reduction. This reflects the increasing attention paid by industry to quality

The process improvement cycle

- Process measurement
  - Attributes of the current process are measured. These are a baseline for assessing improvements
- Process analysis
  - The current process is assessed and bottlenecks and weaknesses are identified
- Process change
  - Changes to the process that have been identified during the analysis are introduced
Principal product quality factors

Quality factors

- For large projects with "average" capabilities, the development process determines product quality.
- For small projects, the capabilities of the developers is the main determinant.
- The development technology is particularly significant for small projects.
- In all cases, if an unrealistic schedule is imposed then product quality will suffer.

Process classification

- Informal
  - No detailed process model. Development team chose their own way of working
- Managed
  - Defined process model which drives the development process
- Methodical
  - Processes supported by some development method such as the RUP
- Supported
  - Processes supported by automated tools

Process applicability

- Informal process: Prototypes
  - Short-lifetime systems
  - 4GL business systems
  - Small/medium-sized systems
- Managed process: Large systems
  - Long-lifetime products
- Methodical process: Well-understood application domains
  - Re-engineered systems
Process choice

- Process used should depend on type of product which is being developed
  - For large systems, management is usually the principal problem so you need a strictly managed process
  - For smaller systems, more informality is possible
- There is no uniformly applicable process which should be standardised within an organisation
  - High costs may be incurred if you force an inappropriate process on a development team
  - Inappropriate methods can also increase costs and lead to reduced quality

Process tool support

- Informal process
- Managed process
- Methodical process
- Improving process
- Generic tools
- Configuration management tools
- Project management tools
- Analysis and design workbenches
- Specialised tools

Process measurement

- Wherever possible, quantitative process data should be collected
  - However, where organisations do not have clearly defined process standards this is very difficult as you don’t know what to measure. A process may have to be defined before any measurement is possible
- Process measurements should be used to assess process improvements
  - But this does not mean that measurements should drive the improvements. The improvement driver should be the organisational objectives

Process change

- Involves making modifications to existing processes
- This may involve:
  - Introducing new practices, methods or processes
  - Changing the ordering of process activities
  - Introducing or removing deliverables
  - Introducing new roles or responsibilities
- Change should be driven by measurable goals
Process change stages
- Improvement identification
- Improvement prioritisation
- Process change introduction
- Process change training
- Change tuning

The CMMI framework
- The CMMI framework is the current stage of work on process assessment and improvement that started at the Software Engineering Institute in the 1980s
- The SEI’s mission is to promote software technology transfer particularly to US defence contractors
- It has had a profound influence on process improvement
  - Capability Maturity Model introduced in the early 1990s
  - Revised maturity framework (CMMI) introduced in 2001

The SEI capability maturity model
- Initial
  - Essentially uncontrolled
- Repeatable
  - Product management procedures defined and used
- Defined
  - Process management procedures and strategies defined and used
- Managed
  - Quality management strategies defined and used
- Optimising
  - Process improvement strategies defined and used

Problems with the CMM
- Practices associated with model levels
  - Companies could be using practices from different levels at the same time but if all practices from a lower level were not used, it was not possible to move beyond that level
- Discrete rather than continuous
  - Did not recognise distinctions between the top and the bottom of levels
- Practice-oriented
  - Concerned with how things were done (the practices) rather than the goals to be achieved
The CMMI model

- An integrated capability model that includes software and systems engineering capability assessment.
- The model has two instantiations
  - Staged where the model is expressed in terms of capability levels;
  - Continuous where a capability rating is computed.

CMMI model components

- Process areas
  - 24 process areas that are relevant to process capability and improvement are identified. These are organised into 4 groups.
- Goals
  - Goals are descriptions of desirable organisational states. Each process area has associated goals.
- Practices
  - Practices are ways of achieving a goal -- however, they are advisory and other approaches to achieve the goal may be used.

CMMI process areas

1

<table>
<thead>
<tr>
<th>Process management</th>
<th>Organisational process definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Organisational process focus</td>
</tr>
<tr>
<td></td>
<td>Organisational training</td>
</tr>
<tr>
<td></td>
<td>Organisational process performance</td>
</tr>
<tr>
<td></td>
<td>Organisational innovation and deployment</td>
</tr>
<tr>
<td>Project management</td>
<td>Project planning</td>
</tr>
<tr>
<td></td>
<td>Project monitoring and control</td>
</tr>
<tr>
<td></td>
<td>Supplier agreement management</td>
</tr>
<tr>
<td></td>
<td>Integrated project management</td>
</tr>
<tr>
<td></td>
<td>Risk management</td>
</tr>
<tr>
<td></td>
<td>Integrated teaming</td>
</tr>
<tr>
<td></td>
<td>Quantitative project management</td>
</tr>
</tbody>
</table>

2

<table>
<thead>
<tr>
<th>Engineering</th>
<th>Requirements management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Requirements development</td>
</tr>
<tr>
<td></td>
<td>Technical solution</td>
</tr>
<tr>
<td></td>
<td>Product integration</td>
</tr>
<tr>
<td></td>
<td>Verification</td>
</tr>
<tr>
<td></td>
<td>Validation</td>
</tr>
<tr>
<td>Support</td>
<td>Configuration management</td>
</tr>
<tr>
<td></td>
<td>Process and product quality management</td>
</tr>
<tr>
<td></td>
<td>Measurement and analysis</td>
</tr>
<tr>
<td></td>
<td>Decision analysis and resolution</td>
</tr>
<tr>
<td></td>
<td>Organisational environment for integration</td>
</tr>
<tr>
<td></td>
<td>Causal analysis and resolution</td>
</tr>
</tbody>
</table>
CMMI assessment

- Examines the processes used in an organisation and assesses their maturity in each process area
- Based on a 6-point scale:
  - Not performed
  - Performed
  - Managed
  - Defined
  - Quantitatively managed
  - Optimising

A process capability profile

The staged CMMI model

- Comparable with the software CMM
- Each maturity level has process areas and goals. For example, the process area associated with the managed level include:
  - Requirements management
  - Project planning
  - Project monitoring and control
  - Supplier agreement management
  - Measurement and analysis
  - Process and product quality assurance
SW-CMM Key Process Areas

- There are key process areas (KPAs) for each level
- Level 2 KPAs include:
  - Requirements management
  - Project planning
  - Project tracking
  - Configuration management
  - Quality assurance

Experience

- It takes:
  - 3 to 5 years to get from level 1 to level 2
  - 1.5 to 3 years from level 2 to level 3
  - SEI questionnaires highlight shortcomings, suggest ways to improve the process

SW-CMM Praise

- It has arguably been successful in this role, even reputedly causing some software sales people to clamour for their organisations to "implement CMM."
- Economic development agencies in India, Ireland, Egypt, Syria, and elsewhere have praised the CMM for enabling them to be able to compete for U.S. outsourcing contracts on an even footing.
- The CMM provides a good framework for organisational improvement. It allows companies to prioritise their process improvement initiatives.

SW-CMM Criticism

- CMM failed to take over the world.
- CMM is well suited for bureaucratic organisations such as government agencies, large corporations and regulated monopolies.
- The use of auditors and executive reports may influence the entire IT organisation to focus on perfectly completed forms rather than application development, client needs or the marketplace.
SW-CMM Criticism

- The CMM does not describe how to create an effective software development organisation.
- The CMM can seem to be overly bureaucratic, promoting process over substance. For example, for emphasising predictability over service provided to end users.

eXtreme Programming

XP Roles

- **Customer**
  - Writes User Stories and specifies Functional Tests
  - Sets priorities, explains stories
  - May or may not be an end-user
  - Has authority to decide questions about the stories

- **Programmer**
  - Estimates stories
  - Defines Tasks from stories, and estimates
  - Implements Stories and Unit Tests

XP Roles, cont’d

- **Coach**
  - Watches everything, makes sure the project stays on course
  - Helps with anything

- **Tracker**
  - Monitors Programmers’ progress, takes action if things seem to be going off track.
  - Actions include setting up a meeting with Customer, asking Coach or another Programmer to help
XP Roles, cont’d

* **Tester**
  - Implements and runs Functional Tests (not Unit Tests!)
  - Graphs results, and makes sure people know when test results decline.

* **Doomsayer**
  - Ensures that everybody knows the risks involved
  - Ensures that bad news isn’t hidden, glossed over, or blown out of proportion

XP Roles, cont’d

* **Manager**
  - Schedules meetings (e.g. Iteration Plan, Release Plan), makes sure the meeting process is followed, records results of meeting for future reporting, and passes to the Tracker
  - Possibly responsible to the Gold Owner.
  - Goes to meetings, brings back useful information
  - Pays for pizza

* **Gold Owner**
  - The person funding the project, which may or may not be the same as the Customer

XP Practices

XP is based on 12 key practices:

* The Planning Process
  - Release Planning & Iteration Planning
* Frequent, Small Releases
* System Metaphor
* Simple Design
* Test Driven Development
* Refactoring
* Pair Programming
* Collective Code Ownership
* Continuous Integration
* Sustainable Pace
* On-site Customer
* Coding Standard

Stages of an XP project

* **Initiation**
  - User Stories
* **Release Planning**
* **Release** (each Release is typically 1 -6 months)
  - Iteration 1 (typically 1 -3 weeks)
    - Development
    - Deployment
    - Acceptance Testing
  - Iteration 2
    - Development
    - Deployment
    - Acceptance Testing
  - ...

Gathering Requirements

- Responsibilities
  - Key Point: The Customer is responsible for the requirements.
  - Programmers help to gather and clarify requirements. Customers especially need help with non-functional requirements and with working out the details of acceptance tests.
- Documentation
  - User Stories
  - Acceptance Test Cases

User Stories

- A short description of the behaviour of the system from the point of view of the Customer
- Use the Customer's terminology without technical jargon
- One for each major feature in the system
- Must be written by the users
- Are used to create time estimates for release planning
- Replace a large Requirements Document

User Stories cont’d

- Drive the creation of the acceptance tests:
  - Must be one or more tests to verify that a story has been properly implemented
- Different than Requirements:
  - Should only provide enough detail to make a reasonably low risk estimate of how long the story will take to implement.
- Different than Use Cases:
  - Written by the Customer, not the Programmers, using the Customer’s terminology
  - More “friendly” than formal Use Cases

User Stories cont’d

- User stories have three crucial aspects:
  - Card
    - Enough information to identify the story
  - Conversation
    - Customer and Programmers discuss the story to elaborate on the details
    - Verbal when possible, but documented when required
  - Confirmation
    - Acceptance tests to confirm that the story has been properly implemented
Acceptance Tests

- Formal test to determine if a system satisfies its acceptance criteria, i.e. the User Stories!
- Should be automated, but may simply a series of repeatable steps
- At least one Acceptance Test for each Story

Release Planning

- Customer defines the business value of desired features (User Stories)
- Programmers provide estimates of 1, 2 or 3 “points”
- Stories larger than 3 points must be split into smaller stories
- Customer decides which Stories are to be included in a Release
- Focus on completing the Stories with the highest business value and highest risk first

Release Planning cont’d

- Stories for a Release are arranged into 1-3 week Iterations
- Higher risk, and higher priority stories in earlier Iterations
- For new system, the 0:th Iteration defines the basic skeleton of the application and infrastructure required
- The Release and Iterations have fixed dates for completion – dates are fixed, scope is variable
- This is the Release Plan

Iteration Planning

- Stories for Iteration are broken down into Tasks by Programmers
- Tasks are estimated by all Programmers as a group
- Programmers “sign up” for Tasks, and estimate the time to complete
- Can only sign up for as many points as were completed in the last Iteration
- Once development begins, Project Velocity measures progress
Programmer Tests

- Automated tests written to test the behaviour of individual classes
- Fundamental to XP, and maintaining a flat cost curve
- XP uses a Test First mentality; write the test, then write to code to make the test pass.
  - "Never write a line of code without a failing test." (Kent Beck)
- No code goes into production unless it has associated tests
- Tests are written first
- Tests determine what code you need to write

Programmer Tests cont’d

- Programmer Tests must run at 100% before code is integrated
- At most one test failing at any time
- "Grey-box" testing
- Assist with Refactoring (promote Courage)
- Testing frameworks exist for many languages:
  - JUnit for Java
  - CPPUnit for C++
  - NUnit for all .Net languages

Scrum -- Introduction

- SCRUM is a loose set of guidelines that govern the development process of a product, from its design stages to its completion.
- SCRUM has been successfully employed by hundreds of different companies in many different fields, with outstanding results.
- There are many similarities between SCRUM and XP, but one of the major differences is that SCRUM is a fairly general set of guidelines that govern the development process of a product
  - SCRUM is often used as a "wrapper" for other methodologies, such as XP or CMM, to guide the overall process of development when using these other methodologies.
**SCRUM Values**

- The SCRUM values are derived from the Agile values of software development:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan

**The SCRUM Process**

- The scrum process has 3 main phases:

<table>
<thead>
<tr>
<th>Pre-game</th>
<th>Mid-game</th>
<th>Post-game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and high-level design</td>
<td>Develop</td>
<td>Wrap</td>
</tr>
<tr>
<td>Adjust</td>
<td>Review</td>
<td>Closure</td>
</tr>
</tbody>
</table>

**Sprint Cycle**

**Daily SCRUM Meeting**

- A 15-minute SCRUM meeting is held every day.
- The SCRUM Master asks the three questions, and all members of the team and interested parties take part and give feedback.
- The meeting should be held at the same place every time, so that people know where to go.

1. What have you accomplished since the last meeting?
2. Are there any obstacles in the way of meeting your goal?
3. What will you accomplish before the next meeting?
Sprint

- Before a sprint is begun, a Sprint Planning Meeting is held to determine what features are to be implemented in that sprint.
  - Develop the product further - implement, test, and document.
  - Wrap up the work - get it ready to be evaluated and integrated.
  - Review the work done in this sprint.
  - Adjust for any changes in requirements or plans.

Creating A Backlog

- The product owner compiles all the requests and specifications that are the basis of the changes of the product, e.g. new functions and bug fixes. After the goals have been defined, the entirety is broken down into segments.
- Each segment should in part create business value and in part be sub-deliverable.
- A prioritised list is made at the same time – the product owner personally makes the decisions at this point.
- When it is time to start a new Sprint, the product owner “freezes” the foremost items on the to-do list and summons the SCRUM team to a meeting.

Backlog

1. Product Backlog -- a repository for requirements targeted for release at some point. High level requirements with high level estimates provided by the product stake-holders.
2. Release Backlog - Requirements pulled from the product backlog and identified and prioritised for an upcoming release. Contains more details about the requirement and low level estimate which are usually estimated by the team.
3. Sprint Backlog - A result from each sprint planning is a backlog of requirements/sub-requirements estimated to be completed at the end of the sprint where the requirements from the release backlog are broken down into manageable chunks that can be accomplished typically in 8 - 16 hours.

The Scrum Team

- The SCRUM team consists of 2 groups:
  - the interested team, which consists of people who are interested, but who will not be doing the work
  - the working team -- people who are interested, and will be doing the work on the project
The Scrum Team, cont’d

- A SCRUM is a self-empowered team where everyone had the global view of the product on a daily basis
- The development team should perform as a sport team, every team member working independently but towards the same goal
- A team typically 6-9 working members, although SCRUM has been successfully used with more members
- The team members decide how the work is arranged and how assignments are distributed
- There are no set project roles – everyone should be able to swap tasks with another member

The Product Owner

- The product owner represents the voice of the customer and ensures that the Scrum team works with the right things from a business perspective
- The product owner administers a product Back-log
- The product owner is often a customer, but can also be part of the internal organisation
SCRUM divided into 3 types by Jeff Sutherland

Type A - Isolated cycles of work

Type B - Overlapping iterations

Type C - All at once

Tyranny of the Sprint

End of Today’s Lecture

Thanks for your attention!