

# Agile SDLC

## Agile manifesto

- Individuals & interactions over process and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following the plan

## Scrum

Scrum is a lightweight framework that helps teams and organizations generate value and is based on transparency, inspection and adaptation.

## Scrum team/ Accountabilities

**Scrum Master:** (促进团队合作, 沟通, 减少障碍): Fostering Communication, Agile Coaching, Protecting the Team, Team support, Remove obstacles

**Product Owner** (manage requirement, manage product backlog, adjust priority, decide what can be deliverable) Terminating a sprint, Participating in daily scrums, Managing product backlog, Anticipating client needs

**Dev Team:** Self organization, Cross functional, Estimates possible release date of features

## Other Agile frameworks

**Kanban:** Visualising work, Limiting work progress, Maximising efficiency, Focus on reducing the time taken to take a project to finish

**Extreme Programming:** Frequent releases in short development cycles, Intended to improve productivity and introduce checkpoints, **Pair programming**

## Project Schedule

Project Schedule is used and maintained throughout the project to monitor and track project progress – it is a living document.

Project schedule contains 1. Duration and dependencies for each task 2. People and physical resources required by each task 3. Milestones and deliverables 4. Project Timeline

## Waterfall (Predictive) Process

- **Constraint:** scope (requirement). They are usually defined and fixed at the start of the project
- **Plan driven** This process is plan-driven, meaning that a detailed plan is made at the beginning of the project.
- **Estimates:** The plan creates estimates for the cost and time required to complete the project based on the *fixed scope*

## Agile (Adaptive) Process

- **Constraint:** In agile, the constraints are **Cost and time**

- **Value/Vision:** The focus is on delivering the most valuable features according to the business needs, rather than a pre-defined set of requirements.
- **Estimates** Agile work within fixed time and cost, with the scope being adjusted based on what can be delivered in that time.

Key Comparison

- **Water fall** The scope is fixed, cost and time estimates are derived from that fixed scope
- **Agile** Time and cost are fixed, while the scope is flexible and can be adjusted based on value delivered.

Examples of Task Dependencies

- Finish-to-Start: Predecessor must finish before Successor can start
- Start-to-Start: Predecessor must start before Successor can start
- Finish-to-Finish: Predecessor must finish before the Successor can Finish
- Start-to-Finish: Predecessor must start before the Successor can finish

Outsourcing/Procurement/Contracts

**Outsourcing :** The practice of engaging an external party (under contract) to perform services or create goods that are traditionally performed in-house by the company's own employees.

- Onshoring: Relocating activities inside national borders to access targeted benefits. (Sydney - Brisbane)
- Nearshoring: Activities relocated to another country with close proximity e.g. New Zealand, Indonesia.
- Offshoring: Activities relocated to another country irrelevant of geographical location and time zones.

**Procurement** buy (outsource) any products or services from outside the organisation

Procurement Phase	Key Activities
Plan	1. Consulting key stakeholders to define the real need
	2. Analyzing how the supply market works
	3. Assessing risks
	4. Defining the best procurement strategy to meet organizational requirements
Source	1. Identifying and engaging suppliers to provide best value for money
	2. Ensuring procurement is conducted with probity and fairness
	3. Determining the appropriate sourcing method (e.g., tendering, negotiation, or direct procurement)
Manage	1. Understanding and managing the rights and responsibilities of all parties involved
	2. Ensuring ongoing supplier relationship management after contract signing

Procurement Phase	Key Activities
	3. Delivering the best outcomes for the organization through effective contract management and continuous monitoring

## Quality Management (week 6)

### Cost of Software Quality Management

**Conformance** : Delivering products that meet requirements and are fit for purpose.

**Nonconformance** :Refers to taking responsibility for failures or not meeting quality expectation

**Quality Management Processes:** 1. Performing Quality Assurance 2. Quality Control 3. Performing Quality Control

### Quality Assurance

Verification and Validation

**Verification:** Ensuring you are building the system right (the right way).

**Validation:** Ensuring that you are building the right system (to meet stakeholder needs).

Quality Assurance –Types of Testing

#### Unit Test

- Used to test each individual component (often a program) to ensure it is defect-free.
- Performed before commencing integration testing.

#### Integration Testing

- Between unit testing and system testing, ensures that subset(s) of the overall system work together correctly.

#### System Testing

- Tests the entire system as one entity. Ensures that the entire system is working correctly.

#### User Acceptance Testing

- Testing performed by end users prior to accepting the delivered system.

Quality Assurance -Software Standards and Systems

Advantages of standards

- Provide a framework around which the quality assurance process may be implemented. -Customers sometimes require a particular quality standard/level when choosing a software vendor.

Disadvantages of standards

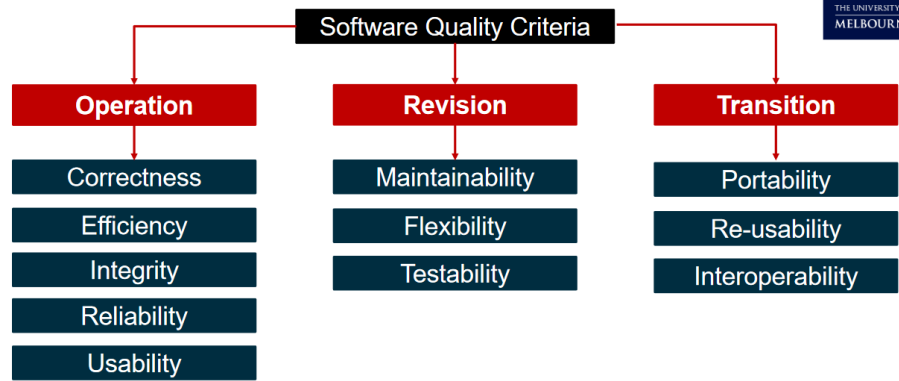
- Involve too much bureaucratic form filling.

Quality Assurance -Capability Maturity Model Integration (CMMI)

Describes an approach for software companies to move from an ad-hoc, immature process to a mature developed process.Organizations are characterised being at a Level from 1-5 based on the processes they follow.

Level	Focus	Key process
1. Initial	Individual effort	
2. Repeatable	Project management	<ul style="list-style-type: none"><li>•Software Project Planning</li><li>•Software Planning &amp; Oversight</li><li>•Software Subcontract Management</li><li>•Software Quality Management</li><li>•Software Configuration Management</li><li>•Requirements Management</li></ul>
3. Defined	Engineering process	<ul style="list-style-type: none"><li>•Organisation Process Focus</li><li>•Organisation Process Definition</li><li>•Peer Reviews</li><li>•Training Programme</li><li>•Intergroup Coordination</li><li>•Software Product Engineering</li><li>•Integrated Software Management</li></ul>
4. Managed	Product & process quality	<ul style="list-style-type: none"><li>•Software Quality Management</li><li>•Quantitative Process Management</li></ul>
5. Optimising	Continual improvement	<ul style="list-style-type: none"><li>•Process Change Management</li><li>•Technology Change Management</li><li>•Defect Prevention</li></ul>

Quality Assurance – McCall Quality Model



Quality planning

Incorporating quality standards into project design

Product Overview Product Plan Quality Goals Process Description Document and Coding Standards Risks and Risk Management

Quality Control and Monitoring

Involves monitoring the software development process to ensure that the quality assurance procedures and standards specified in the Software Quality Plan are being followed.

Three common types of reviews (there are also other types of reviews): 1. Technical Reviews 2. Business Reviews 3. Management Reviews

**Technical Reviews** Reviews of artefacts is performed by peers in the development team but the author/s are involved.

**Business Reviews** The goal of a business review is to ensure that the IT solution provides the functionality specified in the project scope and requirements document.

**Management Reviews** Compares the project's actual progress against a baseline project plan. **Project Manager** is responsible for presenting the project progress and providing a clear picture of the current status. May involve reviewing if the project meets the scope, schedule, budget and quality objectives.

## Agile Centric Software Quality Management

### Test Driven Development (TDD)

1. You are not allowed to write any production code unless it is to make a failing unit test pass.
2. You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
3. You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

### Acceptance-Test Driven Development QA Practices

**Stakeholders** are first engaged to collaboratively discusses acceptance criteria, with stakeholders and team members providing input on what to test before any code is written.

Tests are then written to evaluate these specifications, and only then writing the actual code to implement that feature.

### Behavior-Test Driven Development QA Practices

It focuses on defining the behavior of software through conversation and concrete examples and writing specifications in clear and plain language.

CI / CD meets the first principle of the Agile Manifesto: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software".

Continuous integration requires that every time somebody commits any change, the entire application is built and a comprehensive set of automated tests is run against it.

## Risk analysis(week 3)

**Risk Analysis - Qualitative** risk probability (P) risk impact (I)  $\text{Risk exposure} = P \times I$

**Risk Assessment - Quantitative** Quantitative approaches include **mathematical and statistical techniques**

**Risk Management In Agile Plan:** Risk Registers can be documented in a **Risk Register**

**Identify:** Risk assessment workshops conducted by Product Owner to identify areas. Risks are added as **user stories** and **prioritized in Product Backlog**

**Analyse and Assess (Qualitative and Quantitative):** Team members analyses/assess and plan these risks during sprint planning. They can add new risks as they discover

**Respond (Action):** Mitigation strategies defined for risk items and completed within sprint.

**Monitor and Control:** Product Owners receive regular updates on risk register. Scrum master monitors risks in daily stand ups

## Configuration Management (Week 8)

**A Software Configuration is:** 1. The total of all the artefacts. 2. The current state of all the artefacts 3. The dependencies between all the artefacts.

**The aim of configuration management** is to manage change properly without losing overall consistency through: 1. Establishing processes 2. Setting up repositories 3. Using other appropriate tools and techniques

### CM(Configuration Management)

Configuration Management aims: 1. To identify all items that collectively will make up the configuration. 2. To manage changes to one or more of these items so that the collection remains consistent. 3. To manage different versions of the product. 4. To assure software quality as the configuration evolves over time.

