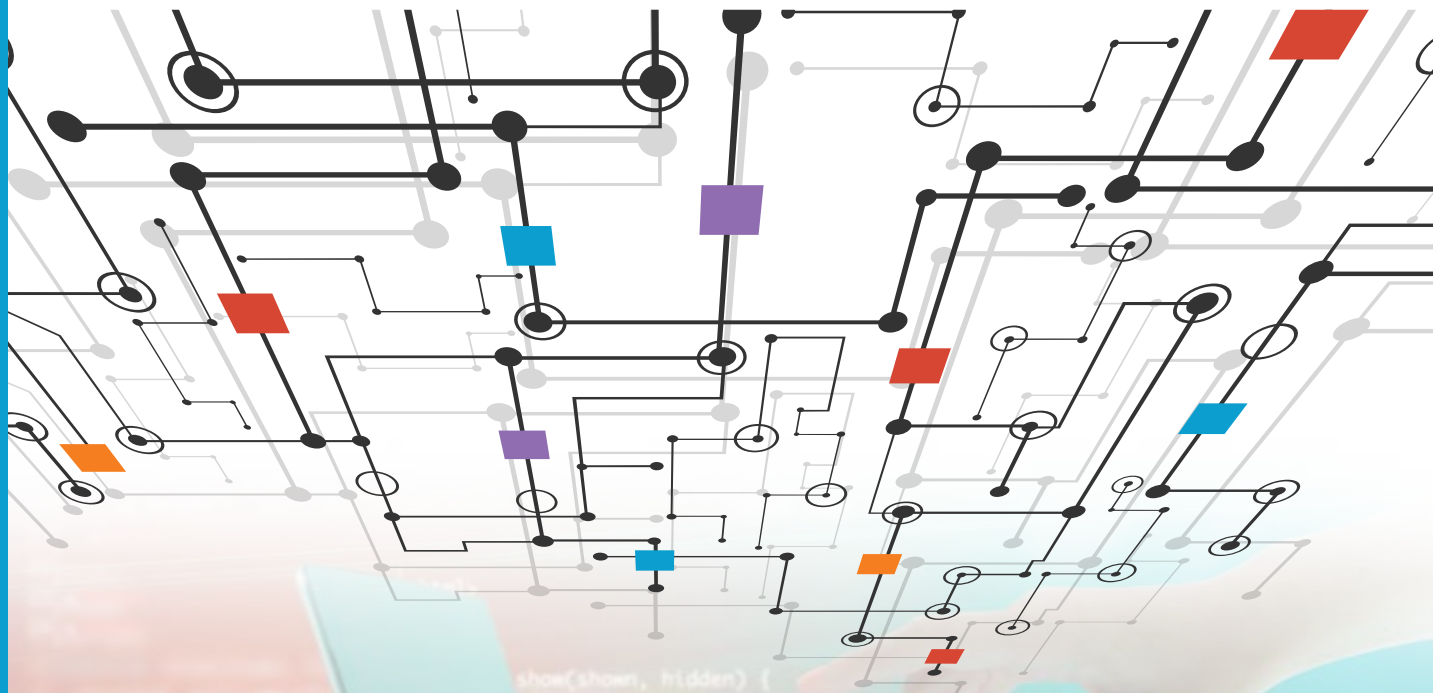
NIBM WORLD WIDE



ebook

SOFTWARE ENGINEERING DSE

AGILE SOFTWARE DEVELOPMENT

Lesson 04 – Agile Software Development

What is Agile?

Agile Software Development is a group of Software Development methodologies which based on the Iterative and Incremental development.

It is a conceptual framework that promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. Agile is Rapid, incremental delivery of software by active and continuous communication between developers and customers.

Agile Manifesto

Agile Manifesto is a formal documentation that describe the Agile. The Agile Manifesto is a brief document built on 4 values and 12 principles for agile software development.

Some of the authors formed the Agile Alliance (a non-profit organization) that promotes software development according to the manifesto's principles.

Figure 4.0.1 Agile Manifesto

The agile has 4 overarching values differentiating it from traditional software development processes.

- ***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 Agile Alliance defines 12 agility principles for those who want to achieve agility:

Agile Principle 1

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Agile Principle 2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Agile Principle 3

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile Principle 4

Business people and developers must work together daily throughout the project.

Agile Principle 5

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Agile Principle 6

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Principle 7

Working software is the primary measure of progress.

Agile Principle 8

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Agile Principle 9

Continuous attention to technical excellence and good design enhances agility.

Agile Principle 10

Simplicity—the art of maximizing the amount of work not done—is essential.

Agile Principle 11

The best architectures, requirements, and designs emerge from self– organizing teams.

Agile Principle 12

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Teams

We are the most important people in AGILE Development.

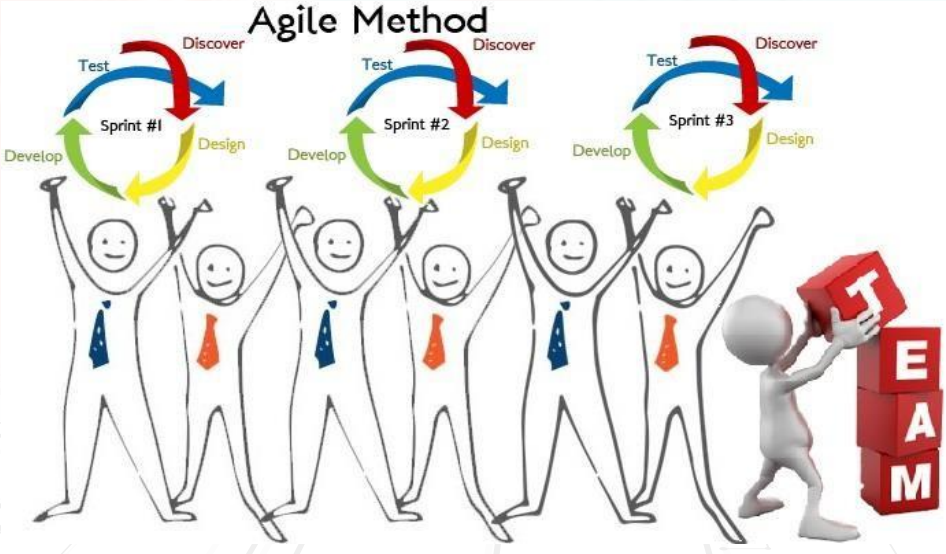


Figure 4.0.2 Most Important people in Agile Development

Figure 4.0.3 Agile Teams

Key Qualities among the People on an Agile Team

Agile team is the most important thing in Agile Development. Normally 6-7 self-organizing, cross functional people are among the team.

- **Competence:** Master in One but knows everything.
- **Common focus:** All are focused on one goal—to deliver a working software increment to the customer within the time promised.
- **Collaboration**: Communication and creating information that will help all stakeholders understand the work of the team.
- **Decision-making ability:** Any good software team (including agile teams) must be allowed the freedom to control its own destiny.
- **Fuzzy problem-solving ability and Self Organize:** Agile team will continually have to deal with issues and must solve those issues in a better manner.

- **Mutual trust and Respect**: The agile team must become "jelled" team. A jelled team exhibits the trust and respect for each other.

Agile Techniques

- Agile Scrum Methodology
- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development (DSDM)
- Crystal
- Feature Driven Development (FDD)
- Test Driven Development (TDD)
- Lean Software Development (LSD)
- Agile Unified Process (AUP)

Scrum Process

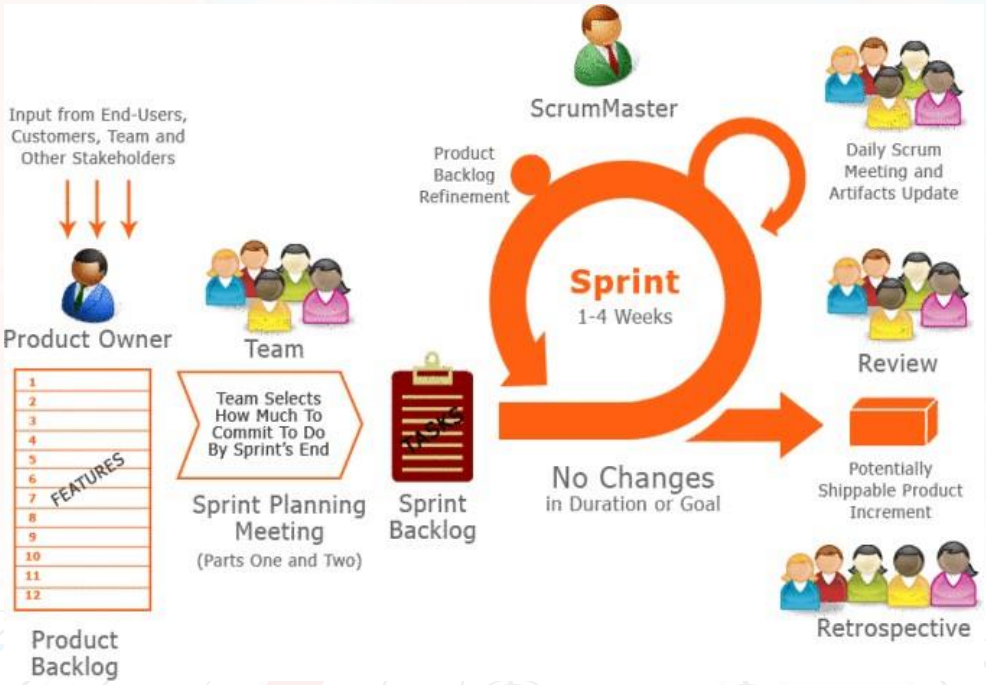


Figure 4.0.4 Scrum Process

Main Roles in Scrum

Scrum Master

Is the person who ensures the process is followed and removes impediment.

Figure 4.0.5 Scrum Master

A good Scrum Master shelters the team from outside distractions, allowing team members to focus maniacally during the sprint on the goal they have selected.

Product Owner or Champion

Product Owner is the Business Analysis. PO represent the stakeholders and the business.

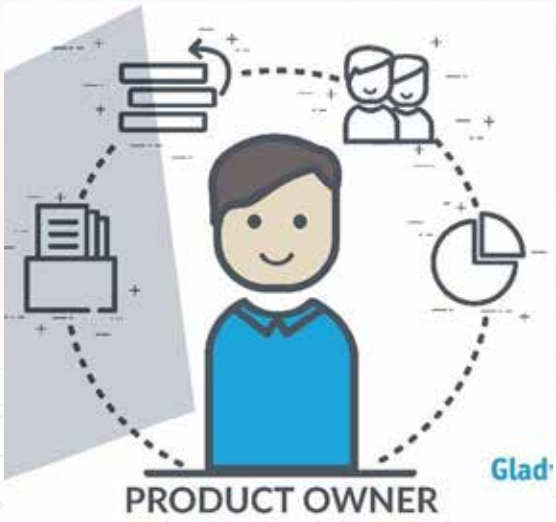


Figure 4.0.6 Product Owner

The Product Owner is responsible for prioritizing the backlog during Scrum development, to ensure it's up to par as more is learned about the system being built, its users, the team and so on.

Development Team

A group who does the coding, implementation, testing etc.

Figure 4.0.7 Development Team

In Scrum, individuals are expected to work beyond their preferred disciplines whenever doing so would be for the good of the team.

Product Back Log & Sprint Back Log

Product Owner get all the requirements and put in the Product Back Log (PB). Those items called as **"User Stories".**

The Product Backlog is a continuously improved list, with the initial version listing only the most preliminary and well-known requirements (no necessary well understood).

Product Backlog evolved based on changes in the product and development environment.

User Stories

Figure 4.0.8 User Story

PRODUCT BACKLOG EXAMPLE

ID	As a...	I want to be able to...	So that...	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators	Must	2	Work in Progress
10	Member	update my email preferences	I'm not bombarded with junk email	Should	2	Work in Progress
11	Member	share content to social networks	I can promote what I find interesting	Could	2	Work in Progress
12	Visitor	create an account	I can benefit from member discounts	Must		To be started
13	Visitor	login	I can post new entries	Must		To be started
14	Visitor	add comments	I can have a say	Must		To be started
15	Visitor	suggest improvements	I can contribute to the site usability	Should		To be started
16	Visitor	contact the Administrators	I can directly submit a query	Could		To be started
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting	Should		To be started
18	Visitor	view a member's profile	I can know more about a member	Must		To be started
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from	Must		To be started

Techno-PM
Project Management Templates

Figure 4.0.9 Product backlog example

- During the Sprint Planning Meeting team and Product Owner determine which backlog items go into the sprint.

- During the meeting, the product owner informs the group of the items in the PB that he or she wants to completed and the team then determines how much of they can commit to complete during the sprint and records in the sprint backlog.

Product Backlog Grooming

It is an on-going activity in Sprint rather than a time box event, together with product owners and development teams. Often, development teams have domain knowledge that they can refine themselves.

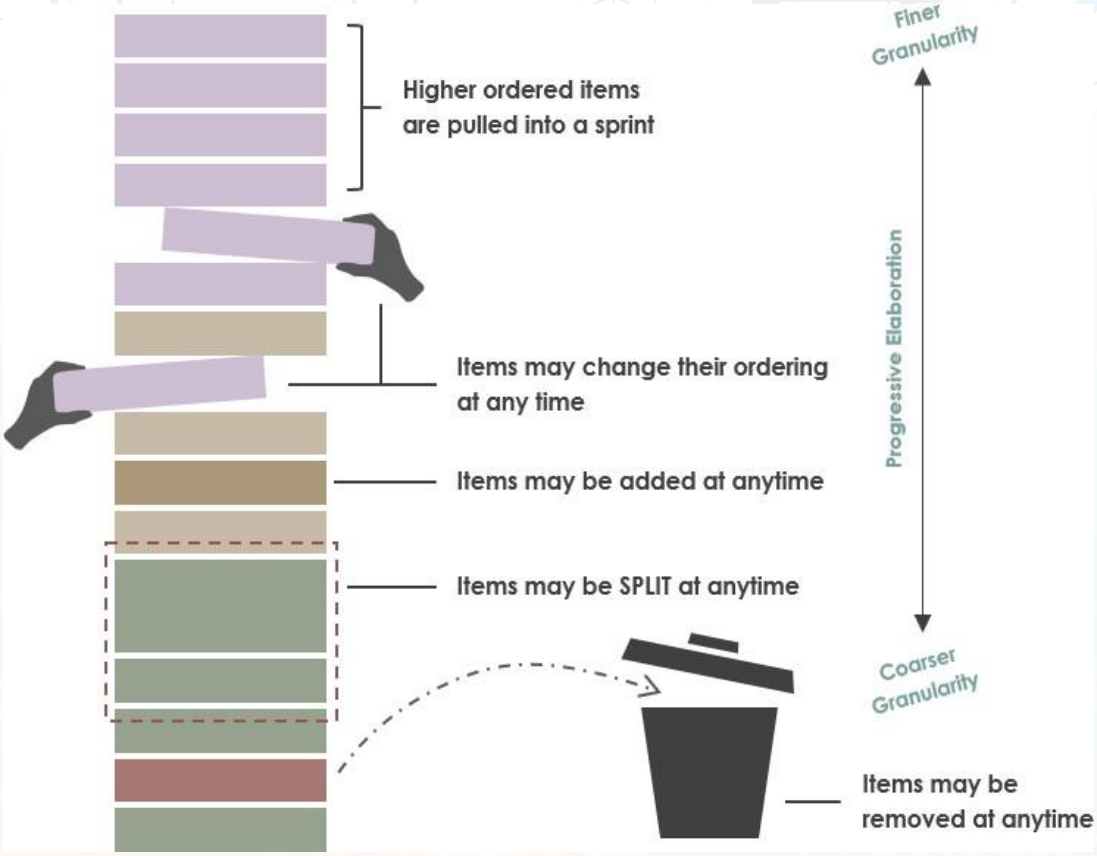


Figure 4.0.10 Product Backlog Grooming

Sprint in the Scrum

Sprint is one time boxed (One week to Four week) iteration of a continuous development cycle. Within a Sprint, planned amount of work has to be completed by the team and made ready for review. Each sprint is preceded by a **Sprint Planning Meeting** where the tasks for the sprint are initiated for the sprint goal.

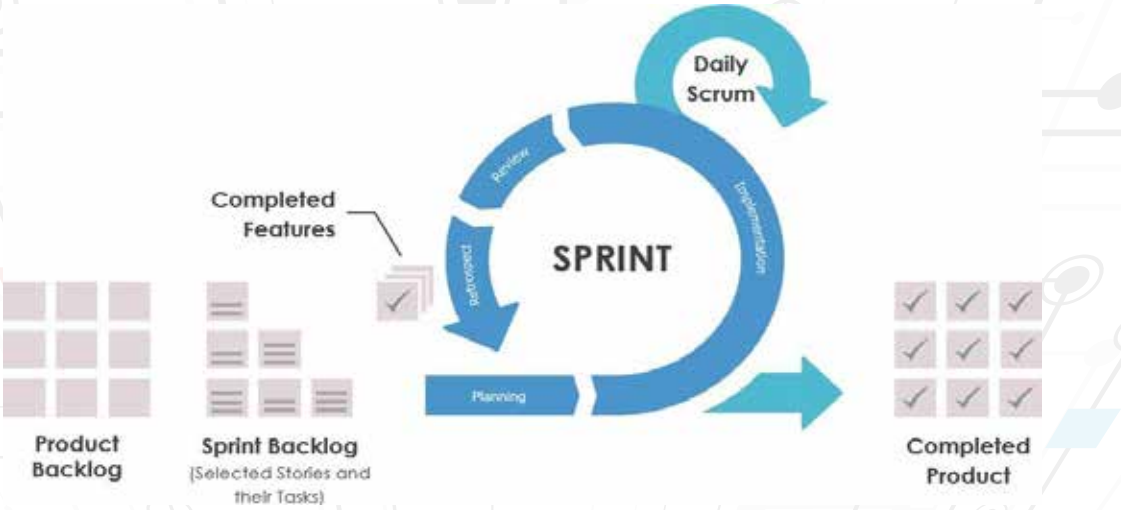


Figure 4.0.11 Sprint

Before the Sprint (Sprint Planning Meeting)

There is a Sprint planning meeting. The length of the sprint planning meeting is proportional to the length of the sprint. This meeting determines what the goals are for that sprint. Based on the team velocity, a set of features are pulled from the top of the product backlog to the Sprint Backlog for the coming Sprint.

Figure 4.0.12 Sprint Planning Meeting

During the Sprint Planning Meeting team and Product Owner determine which backlog items go into the sprint. **"Planning Poker"** is used to estimate how much time is needed for complete the sprint.

Planning Poker

Is used to estimate the effort needed to complete the product backlog items. The product owner need these estimates to find how much time is needed to finish the product backlog items. **"3 Point Estimate"** is used for calculate the most likely time needed for complete the tasks.

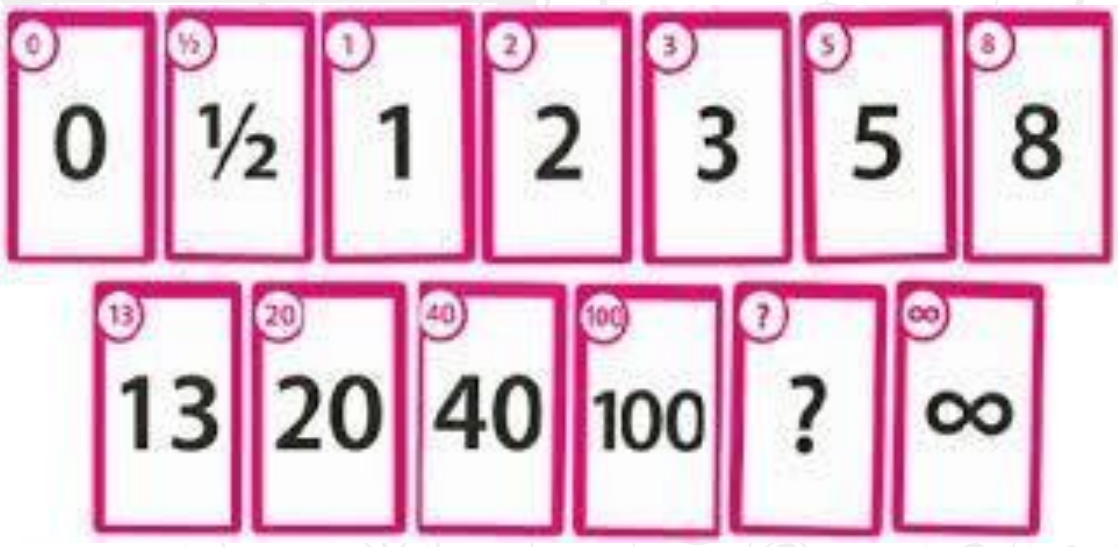


Figure 4.0.13 Planning poker

During Sprint (Daily Scrum Meeting)

No features are added and the sprint goals don't change. The only exception to this is if the team finishes a sprint early. Each day during the sprint, a project status meeting is occurring. This called Daily Scrum or the Daily Standup meeting.

The meeting starts precisely on time. The meeting length is set to 15 minutes. The meeting should happen at the same location and same time every day.

Each person on the team is tasked to answer 3 simple questions:

- What did I do yesterday to help achieve the sprint goal?
- What will I do today to achieve the sprint goal?
- What, if anything is impeding or blocking progress toward the sprint goal?

Figure 3.0.14 During Sprint

During Sprint (Task Board / Kanban)

Kanban is Japanese for "visual signal" or "card." Toyota line-workers used a Kanban to signal steps in their manufacturing process. Scrum task board makes the sprint backlog visible by putting it on a scrum task board.

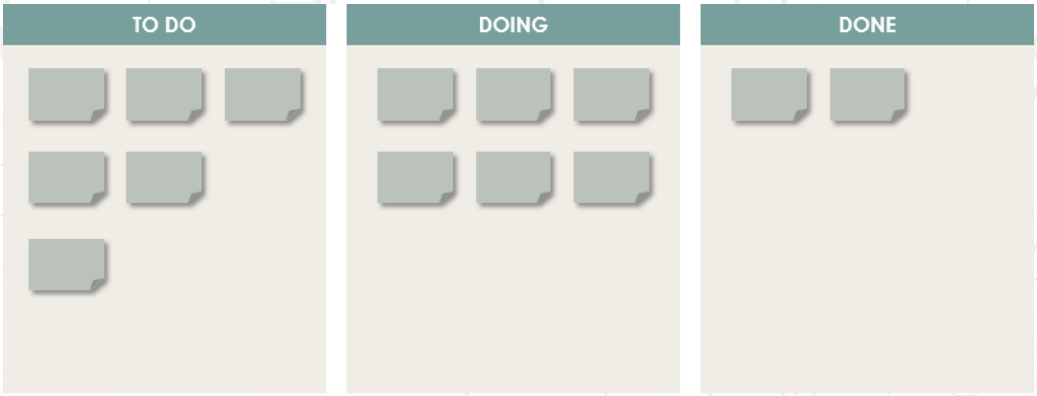


Figure 4.0.15 Task Board / Kanban

Team members update the task board continuously throughout the sprint. Either during or before the daily scrum, estimates are changed and cards are moved around the board. Each row of the scrum task board is a user story.

During Sprint (Sprint Burn Down Chart)

- Daily progress for a sprint over the sprint's length and This chart showing the remaining work in the sprint backlog.
- Updated every day, it gives a simple view of the sprint progress.
- Must update before go to the daily scrum.

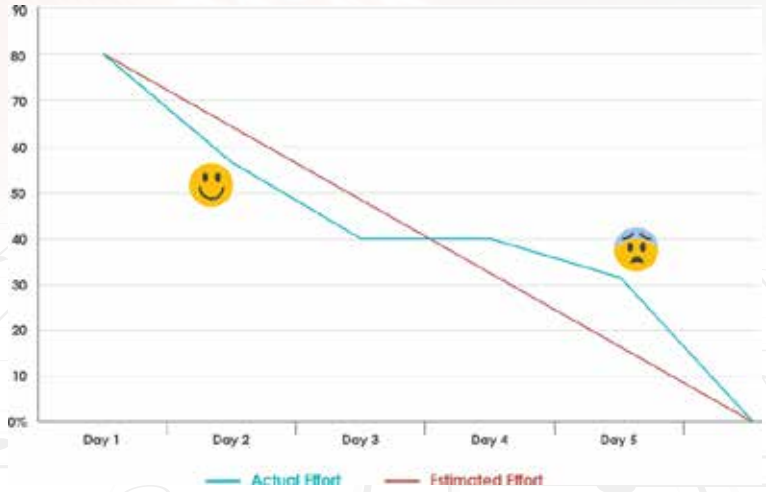


Figure 4.0.16 Sprint Burn Down Chart

Sprint Review Meeting

The sprint review is an informal meeting which the development team, the scrum master, the product owner and the stakeholders will attend. The team gives a demo on the product and will determine what are finished and what aren't.

The purpose of the Sprint Review meeting is for the team to show the customers and stakeholders the work they have accomplished over the sprint and compare it to the commitment given at the beginning of the sprint.

Figure 4.0.17 Sprint Review Meeting

Sprint Retrospective Meeting

The retrospective session is basically an "improvement" meeting held to find ways and means to identify potential pitfalls, past mistakes, and seek out new ways to avoid those mistakes.

The product owner, scrum master, development team members, and optionally stakeholders will be attended.

Two questions asked in the meeting:

- What went well during the sprint?
- What could be improved in the next sprint?

Decide what to start, what to stop and what to continuous.

Figure 4.0.18 Sprint Retrospective Meeting

Lean Software Development

Lean Software Development is an iterative Agile methodology that focuses the team on delivering value to the customer through effective value stream mapping. It is a highly flexible, evolving methodology without rigid guidelines, rules, or methods.

The main principles of the Lean methodology include:

- Eliminating Waste
- Amplifying Learning
- Deciding as Late as Possible
- Delivering as Fast as Possible
- Empowering the Team
- Building Integrity In
- Seeing the Whole

Lean development eliminates waste by asking users to **select only the truly valuable features** for a system, prioritize those features, and then work to deliver them in small batches.

It relies on rapid and reliable feedback between programmers and customers, emphasizing the speed and efficiency of development workflows.

Figure 4.0.19 Main principles of lean methodology

Extreme Programming (XP)

Extreme Programming (XP), is a disciplined approach for high-quality agile software development, focused on speed and continuous delivery. It is intended to improve software quality and responsiveness in the face of changing customer requirements.

It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks. As an example, code reviews are considered a beneficial practice. Taken to the extreme, code can be reviewed continuously through the practice of **pair programming**.

Figure 4.0.20 Pair Programming

Test Driven Development

- Combination of Test First Development and Refactoring.
- The development team writes tests before they even start to write code.
- They then run the test and if it fails they write the code to ensure the test passes.
- On passing the test, the code is then tied up or refactored and a new test is undertaken.

Figure 4.0.21 Test driven development

Dynamic Systems Development Method

This is an Agile Project delivery framework and an Iterative and Incremental. Uses the **MoSCow** prioritization of scope into Musts, Shoulds, Coulds and Won't haves to adjust the project deliverable to meet the stated time constraint.

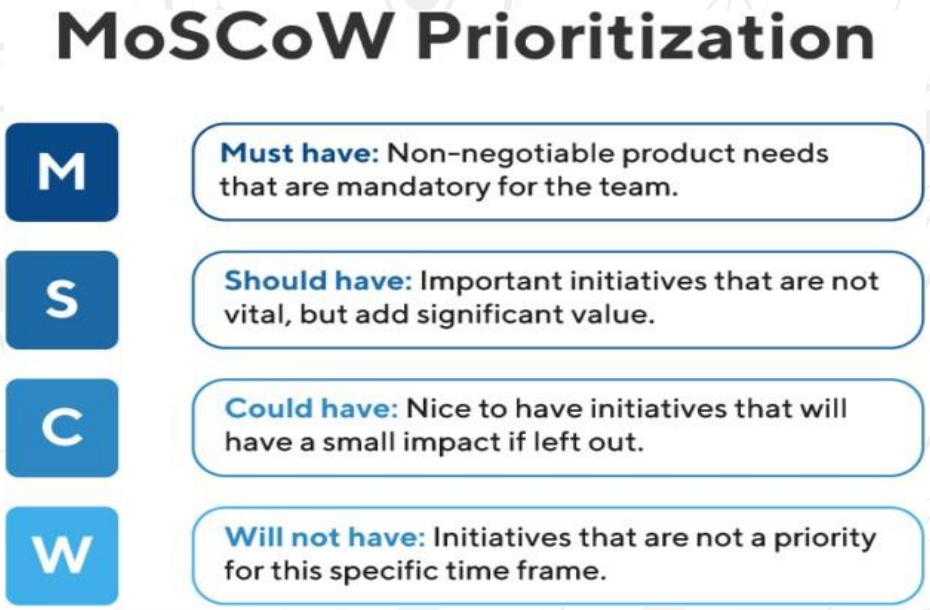


Figure 4.0.22 Dynamic Systems Development Method

Capability Maturity Model Integration (CMMI)

Capability Maturity Model Integration (CMMI) is a framework that describes the key elements of an effective product development and maintenance process. The CMM covers best practice for Planning, Engineering and managing product development and maintenance.

CMMI models provide guidance for developing or improving processes that meet the business goals of an organization. A CMMI model may also be used as a framework for appraising the process maturity of the organization.

Immature vs Mature Software Organization

Immature Software Organizations

- Processes are improvised.
- Managers focus on solving crisis.
- Schedules and budgets are exceeded.
- Product Qualities are bad.
- No objective basis for judging quality.

Mature Software Organizations

- Projects are carried out according to the planned process.
- Mandatory processes are consistent with the way work actually gets done.
- Roles and responsibilities are clear.
- Quality is high.

Characteristics of the Maturity Levels

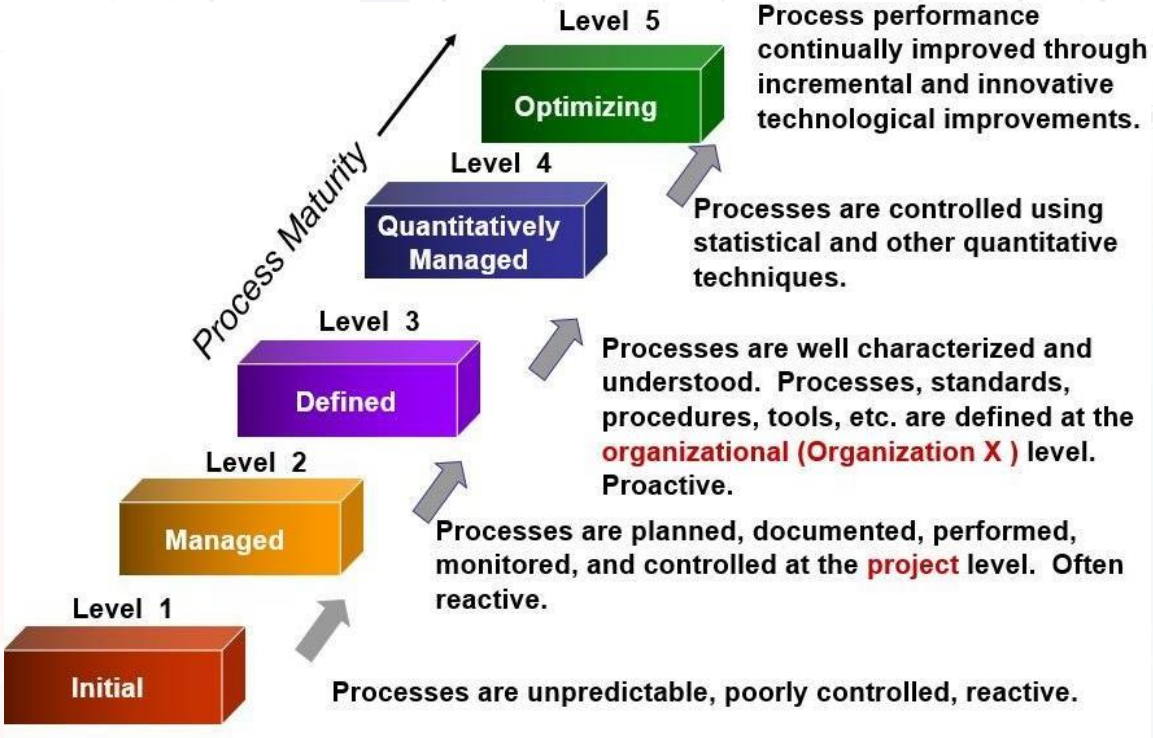


Figure 4.0.23 Characteristics of the Maturity Levels