

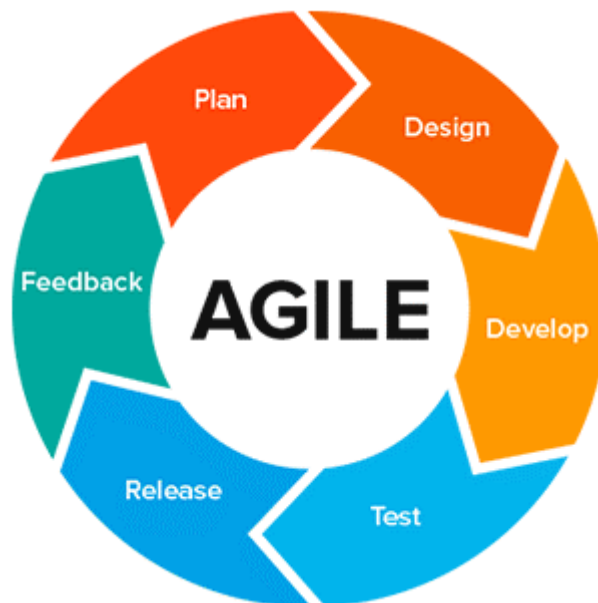
AGILE

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds.

Or

The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

Agile cycle involves:



Agile Methodologies

1. Scrum
2. Kanban
3. Crystal Methodologies
4. Dynamic Software Development Method (DSDM)
5. Feature Driven Development (FDD)
6. Lean Software Development
7. Extreme Programming (XP)

1. SCRUM

Scrum is one of the many types of agile methodology, known for *breaking projects down into sizable chunks called “sprints.”* Agile scrum methodology is good for businesses that need to finish specific projects quickly.

Agile scrum methodology is a project management system that relies on incremental development. Each *iteration consists of 2 to 4 week sprints*, where each sprint aims to build the most important features first and come out with a potentially deliverable product.

Roles of Agile



1. ScrumMaster

Master is responsible for setting up the team, sprint meeting and removes obstacles to progress

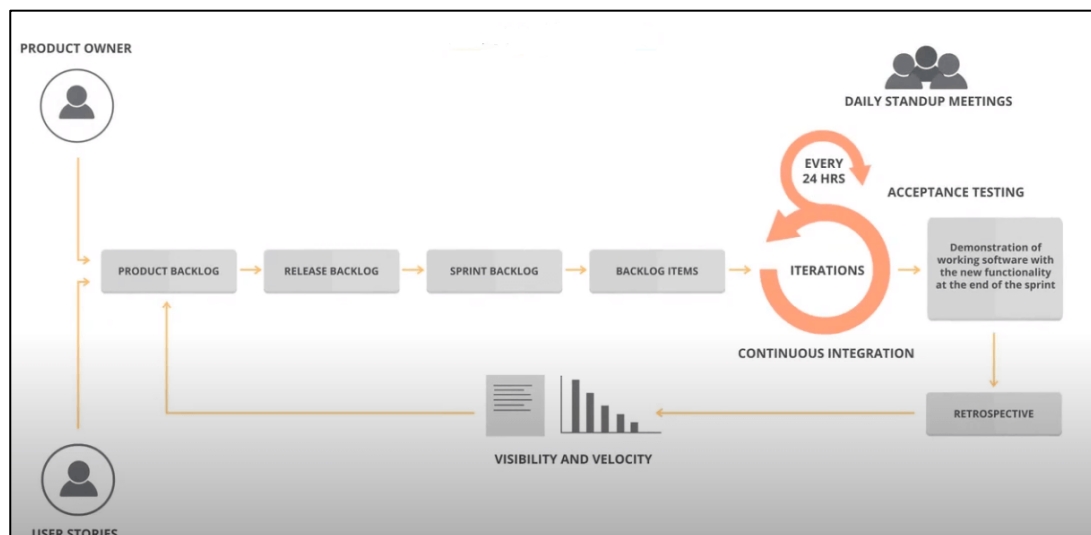
2. Product Owner

The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration

3. ScrumTeam

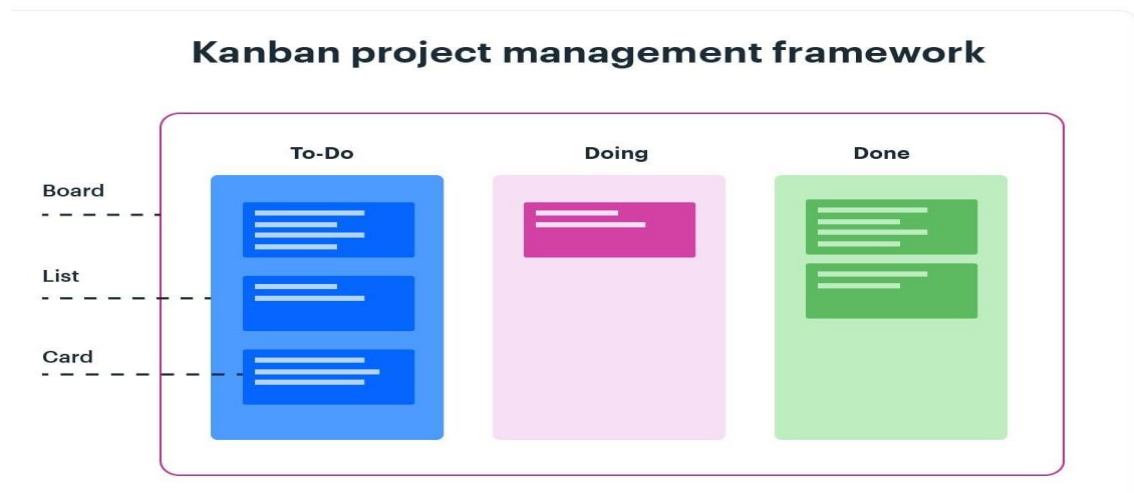
Team manages its own work and organizes the work to complete the sprint or cycle

Scrum Process Flow



2. KANBAN

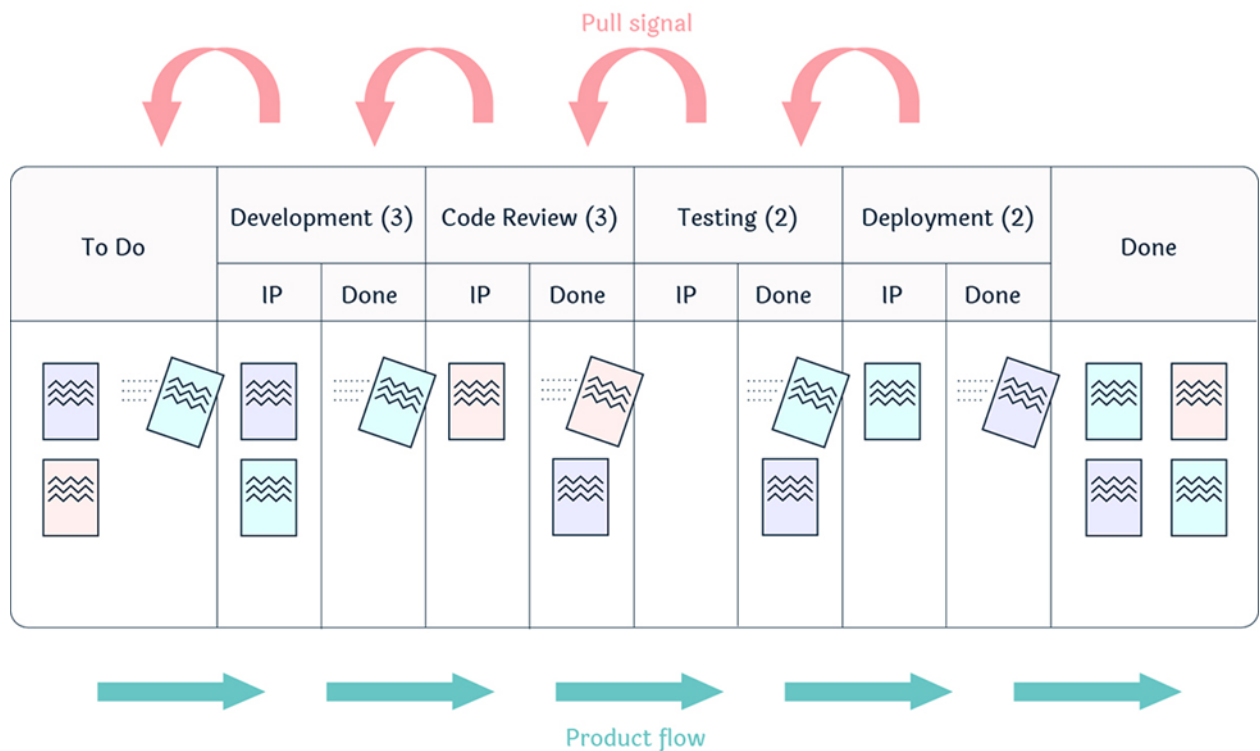
Kanban is a popular workflow management method for defining, managing, and improving services that deliver knowledge work. It helps you visualize work, maximize efficiency, and improve continuously. Work is represented on **Kanban boards, allowing you to optimize work delivery across multiple teams and handle even the most complex projects in a single environment.**



The 6 Core Kanban Principles

The Kanban method is based on six foundational change management and service delivery.

1. Start with what you do now: Kanban is about continuous improvement, but it starts with an understanding of the current processes and workflows.
2. Agree to pursue incremental, evolutionary change: Rather than attempting a large-scale transformation all at once, Kanban advocates for small, incremental changes that build on each other over time.
3. Encourage acts of leadership at all levels: Kanban is not just for managers or team leads but for everyone involved in the work. Anyone can take leadership and suggest improvements based on their observations.
4. Focus on customer needs and expectations: Kanban promotes understanding the needs and expectations of your customers to elevate the quality of the provided services and the value it creates.
5. Manage the work, not the workers: Kanban respects the existing roles and responsibilities of team members and empowers people's abilities to self-organize around the work.
6. Regularly review the network of services: Kanban encourages collaboration and encourages team members to share their observations, ideas, and feedback for improving the work through regular reviews of the entire network of services.

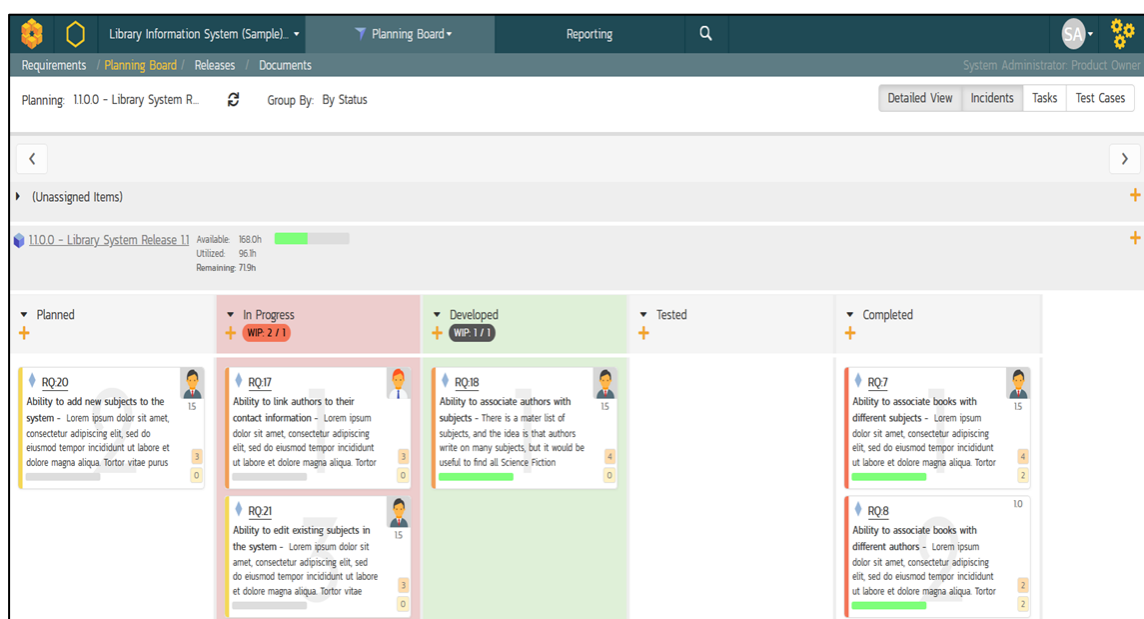


Elements of Kanban:

- **Kanban board:** A Kanban board is one of the Kanban method's key components and is where you visualize all work items. It should be divided into a minimum of 3 columns – Requested, In Progress, Done, representing different process stages.
- **Kanban card:** Kanban cards represent the different work items moving through a Kanban board. They contain important details about the tasks, such as description, deadline, size, assignees, etc.
- **Columns:** They split the Kanban board vertically, and each of them represents a different stage of the workflow. Each Kanban board has 3 default columns: Requested, In Progress, Done. Depending on the complexity of a work process, these three stages can be divided into many smaller sub-columns.
- **Swimlanes:** Horizontal lanes that split a Kanban board into sections. Teams use them to visually separate different work types on the same board and organize homogenous tasks together.
- **Cycle Time:** Cycle time begins at the moment when a new task enters the “in progress” stage of your workflow, and somebody is actually working on it.
- **Lead Time:** Lead time starts at the moment a new task is being requested (it doesn't matter if somebody is actually working on it) and ends with its final departure from the system.
- **Throughput:** The number of work items passing through (completed) a system or process over a certain period. The throughput is a key indicator showing how productive your team is over time.
- **Work in Progress (WIP):** This is the amount of work you are currently working on, and it is not finished yet.

- **WIP limits:** Limiting work in progress means limiting the number of tasks your team can work on simultaneously to avoid overburdening and context-switching.
- **Classes of Service:** Set of policies that help Agile teams prioritize work items and projects.
- **Kanban Cadences:** Cyclical meetings that drive evolutionary change and “fit for purpose” service delivery.
- **Kanban software:** Refers to a digital system that allows the practical application of the Kanban practices and principles to be used by various teams and organizations of all sizes.

Real-Time Example - <https://youtu.be/ea-TH3MixM8>



3. CRYSTAL METHODOLOGY

The crystal method is an agile framework that is considered a lightweight or agile methodology that **focuses on individuals and their interactions**.

The methods are color-coded to significant risk to human life. It is mainly for short-term projects by a team of developers working out of a single workspace.

Two core beliefs of the Crystal method:

- Find your own way and methods to optimize workflow.
- Make use of unique methods to make the project unique and dynamic.

Here concluded that the suitability of the Crystal method depends on three dimensions:

- First, Team size.
- Second, Criticality.
- Third, the priority of the project.

Method	Team size	Project size
Crystal Clear	6 people or less	Small projects
Crystal Yellow	7–20 people	Small to medium projects
Crystal Orange	20–40 people	Medium projects
Crystal Red	40–80 people	Medium to large projects
Crystal Maroon	80–200 people	Large projects
Crystal Diamond or Crystal Sapphire	Very large teams (200+ people)	Very large projects with high criticality

Benefits of using the Crystal Agile Framework:

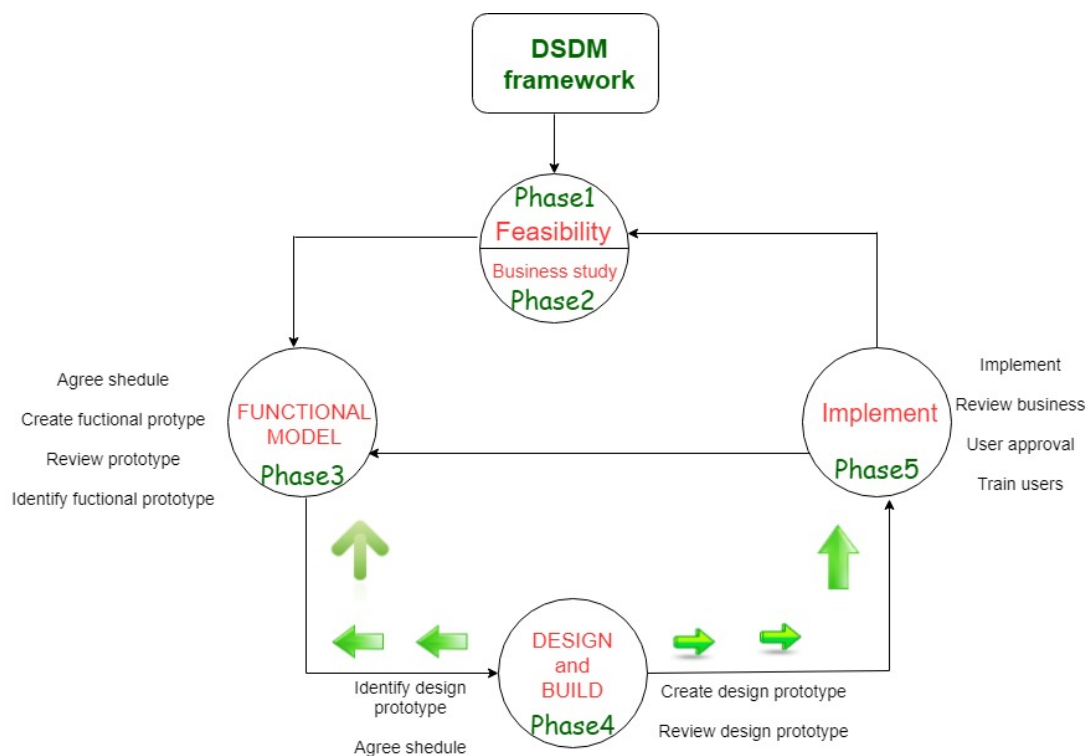
- **Faster delivery** – The framework enables the team to deliver working software faster, which can help gain a competitive advantage in the market.
- **Higher quality** – The framework emphasizes on quality, enabling the team to detect and fix defects early in the development process, resulting in a higher quality product.
- **Improved customer satisfaction** – The framework promotes customer involvement, enabling the team to deliver products that meet customer needs, resulting in higher customer satisfaction.
- **Increased productivity** – The framework enables the team to focus on delivering the highest value features, which can increase productivity and reduce waste.

4. DYNAMIC SOFTWARE DEVELOPMENT METHOD

It is a Rapid Application Development approach to software development that offers an agile project delivery framework. The participants of the team are expected to involve actively in the development activities with necessary decision-making powers vested in them.

DSDM project involves 7 phases of software development –

1. pre-project
2. feasibility study
3. business study
4. functional model iteration
5. design and build iteration
6. implementation
7. post-project.



- **Feasibility Study:**

It establishes the essential business necessities and constraints related to the applying to be designed then assesses whether or not the application could be a viable candidate for the DSDM method.

- **Business Study:**

It establishes the use and knowledge necessities that may permit the applying to supply business value; additionally, it is the essential application design and identifies the maintainability necessities for the applying.

- **Functional Model Iteration:**

It produces a collection of progressive prototypes that demonstrate practicality for the client. The intent throughout this unvarying cycle is to collect further necessities by eliciting feedback from users as they exercise the paradigm.

- **Design and Build Iteration:**

It revisits prototypes designed throughout useful model iteration to make sure that everyone has been designed during a manner that may alter it to supply operational business price for finish users. In some cases, useful model iteration and style and build iteration occur at the same time.

- **Implementation:**

It places the newest code increment (an “operationalized” prototype) into the operational surroundings.

(a) the increment might not 100% complete or,

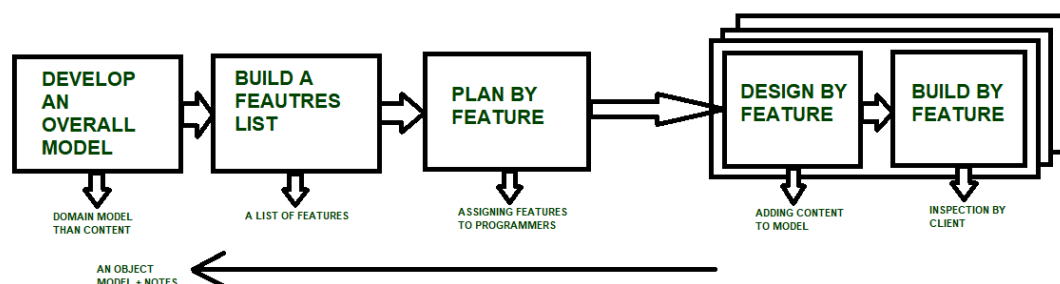
(b) changes are also requested because the increment is placed into place. In either case, DSDM development work continues by returning to the useful model iteration activity.

5. FEATURE DRIVEN DEVELOPMENT (FDD)

- This method emphasises on designing and building features of the product. Feature driven development *defines short cycles of work that needs to be completed for each feature.*
- FDD develops a product considering few aspects like domain object modelling, component ownership, feature teams, inspections, configuration management, regular builds and visibility of progress and results.

FDD breaks projects down into five basic activities:

- Develop an overall model
- Build a feature list
- Plan by feature
- Design by feature
- Build by feature



Characteristics of FDD:

- **Short iterative:** FDD lifecycle works in simple and short iterations to efficiently finish the work on time and gives good pace for large projects.
- **Customer focused:** This agile practice is totally based on inspection of each feature by client and then pushed to main build code.
- **Structured and feature focused:** Initial activities in lifecycle builds the domain model and features list in the beginning of timeline and more than 70% of efforts are given to last 2 activities.
- **Frequent releases:** Feature-driven development provides continuous releases of features in the software and retaining continuous success of the project.

6. LEAN SOFTWARE DEVELOPMENT

- Lean product development is a process for building products faster with less waste. Lean development forces the team to ruthlessly remove any activity that does not bring ultimate value to the product.
- The goal of lean product development is to improve the overall throughput of the system, sustainably, in ways other than just simply throwing more money and resources into it.

This technique is based on '**just in time production**'. The idea is to increase the pace at which a software is being built and reduce cost as a result. Lean development can be categorised into seven parts –

1. Eliminate waste
2. Enhance learning
3. Deferring commitment
4. Early delivery
5. Empower the team
6. Build integrity
7. Optimize the complete process



Key feature of LSD:

- Focus on continuous improvement and waste elimination
- Iterative and collaborative approach
- Development of Minimum Viable Product (MVP)
- Customer involvement and feedback
- Flexible and adaptive approach
- Prioritization of essential functions and minimization of waste

8. EXTREME PROGRAMMING

Extreme programming or XP is a method which **serves a useful purpose when customer demands are rapidly changing**. It facilitates frequent releases of the product under development in short development cycles which is helpful in improving the productivity of the system.

Extreme programming generally involves six phases:

1. Planning
2. Analysis
3. Design
4. Execution
5. Wrapping
6. Closure

