**INTORDUCTION OF KANBAN**

Kanban is a work management system designed to help you visualise your work, limit work in progress and maximise efficiency, which we call flow. Kanban is a Japanese word for visual signal.

Kanban is a way to get thinks done and it also works great in conjunction with scrum, a Kanban is a lean scheduling system,

Kanban was used in manufacturing settings to control inventory throughout the supply chain, using a practice called just-in-time (JIT) manufacturing.

A Kanban system utilizes visual cues that tell you what to produce, how much to produce and when to produce it.

the Kanban methodology adapts the same concept by ensuring that the amount of required work is the same as the work capabilities of the team.

It continuous improvement, flexibility in [task management](https://kissflow.com/project/team/guide-to-task-management/), and enhanced workflow. With this illustrative approach, the progress of the whole project can be easily understood in a glance.

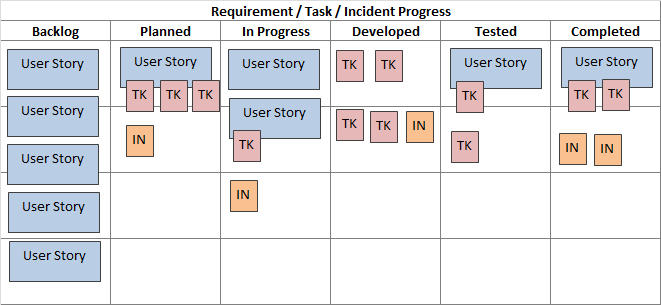
**WORKING PROCESS**

Lean Software Development is a set of principles that helps teams deliver software in a more efficient manner. In a lean environment, activities or processes that result in the expenditure of effort and/or resources towards goals that are not producing value for the customer should be eliminated.

Essentially, lean development is centered on preserving value with less work. Lean approaches are the building blocks of Six Sigma and Just In Time (JIT) development.

When used for software development, Kanban uses the simplified stages in the software development lifecycle (SDLC) to represent the different stages in the manufacturing process.

The aim is to control and manage the flow of features (represented by Kanban cards) so that the number of features entering the process matches those being completed

[](https://www.inflectra.com/GraphicsViewer.aspx?url=Methodologies/Kanban.xml&name=wordml://03000001.png)

Kanban is an [agile methodology](https://www.inflectra.com/Methodologies/Agile-Development.aspx)that is not necessarily iterative. Processes like [Scrum](https://www.inflectra.com/Methodologies/Scrum.aspx)have short iterations that mimic a project lifecycle on a small scale, having a distinct beginning and end for each iteration.

Kanban allows the software to be developed in one large continuous development cycle.

Despite this, Kanban is an example of an agile methodology because it fulfils all twelve of the principles behind the Agile Manifesto, because while it is not iterative, it is incremental.

**FEATURE DRIVEN DEVLOPMENT (FDD)**

FDD is a framework in the [Agile methodology](https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/) The workload is divided into short [Agile iterations](https://www.wrike.com/agile-guide/faq/what-is-agile-iteration/), where developers repeat steps until the final deliverable for release.

FDD is an iterative software development methodology intended for use by large teams working on a project using object-oriented technology.

FDD is a solid alternative to the more well-known [Agile frameworks](https://www.wrike.com/project-management-guide/project-management-frameworks/) such as [Scrum](https://www.wrike.com/project-management-guide/faq/what-is-scrum-in-agile/) and [Kanban](https://www.wrike.com/blog/kanban-methodology-ultimate-guide/#What-is-Kanban). communicate rather than [daily meetings](https://www.wrike.com/blog/stand-up-meetings-best-practices/), which can be time-consuming.

The methodology description includes some prescription about what tasks should be done and what roles should be doing them, So many do not consider it a truly agile methodology.

FDD is suitable for large-scale, long-term projects, as it enables teams to manage

FDD is good for organizations that are transitioning from a phase base approach to an iterative approach but are not comfortable getting rid of all task and role assignments

**FIVE BASIC PROCESSES OF FDD**

1.Develop an overall model

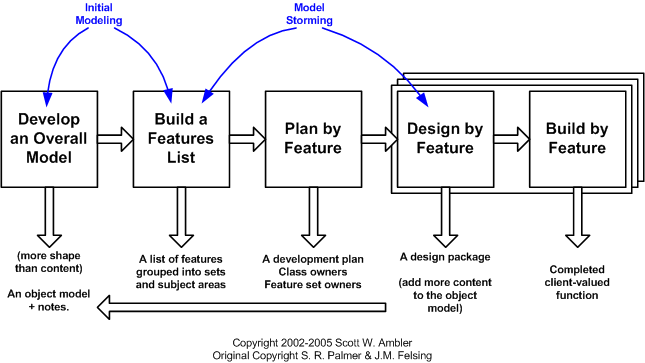
2.Build a features list

3.Plan by feature

4.Build by features

5.Design by features

**The FDD Project lifecycle.**



**Develop an overall model**

Here, an FDD team will determine the project scope. Multiple models will be proposed and merged to create one overall model.

**Build a features list**

Next, the team members will outline the customer-focused features to be developed. They will be small functions that can be completed in a short period of time.

(An example could be to create an automatic reminder for subscription to renewal.)

**Plan by feature**

The team will assess the individual features in the list and arrange them in the appropriate order. Then, the features will be assigned to team members.

**Build by features**

At this stage, the team’s chief programmer will choose which features to develop within a two-week period.

A design package will be created for each feature, and team members will conduct a review before building commences.

**Design by features**

Developers work to build the code for the aforementioned features. This code will be tested before the final version is created