- Focus on the tasks is ensured with a limit on work-in-progress WIP.
- The Pull approach enables resources to complete the tasks at hand before a new task is taken up.
- Optimizing lead-time $cycletime$ results in faster delivery.
- Visualization of the workflow with Kanban board draws immediate attention to any bottlenecks that can be resolved immediately.
- Empowerment of the team makes the team accountable for the success.

KANBAN - PROJECT MANAGEMENT

Kanban is adapted to software development as a project management approach. Kanban in software development supports a continuous workflow, termed as Value Stream.

Value Stream

The Value Stream consists of all actions required to bring a project from creation to completion.

The actions can –

- Add Value to the project
- Add no Value, but unavoidable
- Add no Value, avoidable $termedaswaste$

Elimination of Waste

Anything that does not add any value to the project is known as Waste. Kanban facilitates elimination of waste.

In software development, there are three types of waste –

- Waste in code development
- Waste in project management
- Waste in team potential

Waste in Code Development

Waste in code development is due to the following reasons –

- **Partially completed work** – The partially completed work can become outdated and unusable. It can be eliminated with iterative cycles and with modular code that completes within the iteration.
- **Defects** – In developing a code, correction and retesting requires time and resources. It can be eliminated with up-to-date test suite, completing testing within the iteration and continuous customer feedback.

Waste in Project Management

Waste in project management is due to the following reasons –

- **Extra Processes** – Unnecessary documentation that requires time and resources. It can be eliminated with –
  - Pre-planning of what processes are relevant and necessary.
  - Documentation review, that ensures relevant and necessary processes are followed.
- **Code Handoffs** – means passing the work from one person or team to another, after the first person's work is complete. It may give rise to lack of knowledge. It can be eliminated by keeping the flowcharts and wireframes visible and clear.

- **Extra Functions** – These are features that are not required by the customer. Effort and time are wasted in developing the functions required to implement the features that the customer does not want. It can be eliminated with continuous interaction with customer and testers involving in the requirements gathering as they can better visualize the scenarios and expected behavior of the system.

Waste in Team Potential

Waste in team potential is due to the following reasons –

- **Task Switching** – It leads to the danger of multi-tasking, which is a waste. It can be eliminated with focus on a task with every release. Large process steps are segmented into tasks to –
  - Improve visibility
  - Reduce dependencies
  - Enable easy flow of work
  - Focus on the cycle-time of delivered work
  - Give a way to detect and resolve bottlenecks
- **Waiting** – Time for getting instructions or information – Team is subjected to sit idle if the decisions are not made by the team, or if the information provided to the team $developers, testers, etc.$ are expensive resources. It can be eliminated by allowing the team members $developers, testers, etc.$ to –
  - Take decisions so that they do not have to wait for instructions
  - Have access to information so that it can be used as and when required

KANBAN - AGILE

Agile Kanban is Agile Software Development with Kanban approach. In Agile Kanban, the Kanban board is used to visualize the workflow. The Kanban board is normally put up on a wall in the project room. The status and progress of the story development tasks is tracked visually on the Kanban board with flowing Kanban cards.

Kanban Board

Kanban board is used to depict the flow of tasks across the value stream. The Kanban board –

- Provides easy access to everyone involved in the project.
- Facilitates communication as and when necessary.
- Progress of the tasks are visually displayed.
- Bottlenecks are visible as soon as they occur.

Advantages of Kanban board

The major advantages of using a Kanban board are –

- **Empowerment of Team** – This means –
  - Team is allowed to take decisions as and when required.
  - Team collaboratively resolves the bottlenecks.
  - Team has access to the relevant information.
  - Team continually communicates with customer.
- **Continuous Delivery** – This means –

- Focus on work completion.
- Limited requirements at any point of time.
- Focus on delivering value to the customer.
- Emphasis on whole project.

The tasks and stories are represented by Kanban cards. The current status of each task is known by displaying the cards in separate columns on the board. The columns are labeled as **To Do, Doing**, and **Done**. Each task moves from **To Do** to **Doing** and then to **Done**.

Kanban Board is updated on a daily basis as the team progresses through the development.

WIP Limit

The label in the Doing column also contains a number, which represents the maximum number of tasks that can be in that column at any point of time. i.e., the number associated with the **Doing** column is the WIP $Work - In - Progress$ Limit.

Pull Approach

Pull approach is used as and when a task is completed in the Doing column. Another card is pulled from the To Do column.

Self-directing

In Agile Development, the team is responsible for planning, tracking, reporting and communicating in the project. Team is allowed to make decisions and is accountable for the completion of the development and product quality. This is aligned to the characteristic of empowerment of the team in Kanban.

Continuous Flow

In Agile development, there is no gate approach and the work flows across the different functions without wait-time. This contributes in minimizing the cycle time characteristic of Kanban.

Visual Metrics

In Agile Kanban, the metrics are tracked visually using –

- Kanban Board
- Burndown Chart

Uses of Kanban board

Kanban Board is used to –

- Measure the cycle times, that can be used to optimize average cycle time.
- Track WIP limit to eliminate waste.
- Track resource utilization to eliminate waste.

Uses of Burndown chart

Burndown chart is used to capture –

- The current status of the tasks and stories.
- The rate of progress of completing the remaining tasks.

As Kanban Board is updated daily, it contains all the information that is required by the Burndown charts.

KANBAN - LEAN AND AGILE

In Agile Kanban, the user stories are broken into tasks and Kanban cards are used to track the tasks on the Kanban board. Agile Kanban has a concept of iteration that is not present in Kanban. Further, no processes are considered.

Kanban in Value Stream

Kanban is defined to be executed in value stream with focus on delivery of value. Kanban in software development can be visualized as the features flowing across the value stream. All the Kanban characteristics $ReferChapter - CharacteristicsofKanbaninthisTutorial$ are met in the Kanban approach for software development.

Feature Kanban Board

Feature Kanban Board is used to track the Feature Driven Development with Kanban Approach. Each Feature is assigned to a particular release. The columns in the Kanban board represent releases. Hence, each column contains all the features assigned to the release represented by it.

Each feature is broken into stories. Each release is broken into iterations. The iteration is executed in an Agile Development approach. This can be treated as a sub-stream in the value stream, with the stories to be completed within that iteration assigned to it.

Agile Kanban in Sub-stream

Agile Kanban approach is followed within each sub-stream that is implemented as an iteration. Each story is broken into tasks in the iteration. Task Kanban board is used to track the status and progress of the story development tasks. The current status of each task is known by displaying the cards in separate columns on the board. The columns are labeled as To Do, Doing, and Done. Each task moves from To Do to Doing and then to Done.

Continuous Delivery

Continuous delivery to the customer is ensured with features tracked on feature Kanban board and stories representing features tracked on task Kanban board.

Delivery through a release is accomplished by –

- Continuous tracking
- Constant communication with the customer
- Adjusting development plan as required
- Focusing on delivery of value to the customer

Agile development as well as Kanban maintain team collaboration. This, in turn helps in identifying and resolving Bottlenecks immediately as required by Kanban. This results in accomplishment of all the needed tasks within the iteration to deliver quality product, which meets customer expectations.

Continuous Process Improvement

Kanban supports process improvements to enhance the delivery approach continuously.

Consider a requirement that is a change or addition to the product. In such a case, Kanban cards can be used to visualize the requirement passing through the processes of analysis, design, development, product integration and testing. This is different from the Waterfall approach in the sense that it does not require completion of one process for all the requirements to flow to the next process in the sequence.

Such an implementation of Kanban in product maintenance allows maintainability, reliability and integrity of the product. The required process improvements are gathered at regular intervals and implemented on a continuous basis.

KANBAN - SCRUM

In this chapter, we will learn the similarities and differences between Kanban and Scrum. These similarities and differences will help you in choosing the correct method for your project.

Kanban and Scrum - Similarities

Similarities between Kanban and Scrum are –

- Both are Agile.
- Both use pull scheduling.
- Both limit WIP, Kanban at task level and Scrum at sprint level.
- Both use transparency across the development.
- Both focus on delivering releasable software early.
- Both are based on self-organizing teams.
- Both require breaking the work into pieces.
- In both the methods, the release plan is continuously optimized based on empirical data $Scrum – Velocity, Kanban – LeadTime/CycleTime$.

Kanban and Scrum - Differences

The differences between Kanban and Scrum are as follows –

S.No	Scrum	Kanban
1	Scrum prescribes roles.	In Kanban, roles are optional.
2	Product backlog is to be prioritized.	Prioritization is optional.
3	Sprints are to be time-boxed. You can choose the length of the sprint, but once chosen, the same length is to be maintained for all the sprints.	Time-boxed iterations are optional.
4	Scrum team needs to commit to a particular amount of work for the sprint.	Commitment is optional.
5	Cross-functional teams are prescribed.	Cross-functional teams are optional. Specialist teams are allowed.
6	Uses velocity as default metric for planning and process improvement.	Uses lead time $cycletime$ as default metric for planning and process improvement.
7	Items such as stories, tests must be broken down so that they can be completed within one sprint.	No particular item size is prescribed.
8	Sprint backlog shows what tasks are to be executed during the current sprint. These tasks are displayed on Scrum board. Scope of the sprint is fixed. WIP is limited per unit of time $WIPlimitisthevelocity$.	Tasks are defined at workflow level. WIP is limited per workflow state.

9	Additions/Changes cannot be done within a sprint.	Additions /changes can be done if WIP limit is not crossed.
10	New Scrum board is set at the beginning of every sprint.	Kanban board is persistent.
11	Daily meetings need to be conducted.	Daily meetings are optional.
12	Burn-down charts are prescribed.	No particular chart is prescribed.

Kanban vs. Scrum

The following advantages can help you choose between Kanban and Scrum –

- You need to choose Kanban if you already have working processes and you want to improve without disturbing the whole system whereas you need to choose Scrum if you want to introduce a new process in the organization.
- You can use Kanban in the product development with Feature Driven Development to track the workflows in the value stream whereas you can use Scrum for the development in each iteration.
- You need to define the WIP Limits in Kanban explicitly whereas you need to define the sprint length in scrum that imposes WIP limits implicitly.
- Both Kanban and Scrum are adaptive but Scrum is more prescriptive than Kanban.
- Kanban imposes only two Rules: Visualize workflow and limit WIP whereas Scrum imposes more constraints such as time-boxed Sprints.
- Kanban leads to organizational process improvements, both in management and development. Kanban also supports maintenance activities. Scrum leads to high throughput in small development teams. It does not contribute to product development and maintenance workflows that are longer in duration with unpredictability on the size of work units and changes. Scrum does not emphasize on optimizing management activities.
- In Kanban, you can choose when to do planning, process improvement, and release. You can choose to do these activities on a regular basis or on-demand. Scrum iteration is one single time-boxed Sprint combining three different activities: planning, process improvement, and release $if required$.

Thus, Kanban and Scrum are effective tools in their specific contexts. You can combine Kanban and Scrum to derive maximum benefits from both.

Adapting Kanban and Scrum Together

You can use Kanban and Scrum together by implementing those characteristics that will suit your needs. The constraints of both need to be considered before adapting them. For instance, Scrum requires Time-boxed Sprints and if you do away with those, you cannot say that you have implemented Scrum. Both give you a basic set of constraints to drive your own process improvement.

KANBAN - TOOLS 1

Several project management tools that follow Kanban approach are available. In this chapter, you can have an overview of the following Kanban Tools –

- Kanban Tool
- Kanbanery
- LeanKit
- JIRA Software
- Earliz