**History of KANBAN:**

In the late 1950th the Company Toyota (Car Manufacturer) wanted to revalise (revaliser) with American Opponents, to fulfil this task, the industrial **NIPONG**, called an Engineer Named **Taiichi Ohno**, who had as mission to optimize car manufacturing for the Toyota company.

By this task he invented the method **KANBAN**, a Japanese word meaning **tickets**. This method was created in view of limiting the amount of inventory tied up in ‘**Work in Progress’** on a manufacturing floor (LIKE**: Inventory waste, Time spent in production**), and also to visualize future expansions or new opportunities.

This method became later a revoke territory by Agile software development teams, and recently started been recognized across various industries.

**What is KANBAN:**

/Kanban is a workflow management method for defining, Managing and improving services that deliver knowledge work. Kanban manages all type of projects.

It replies on a simple approach which represent visually a workflow in the form of tables(**Kanban Table**).

**Core Concept of Kanban:**

* **Visualize Workflow**
  + Split entire work into segments or states of workload, and create column that defines the workflow steps.
  + Attribute each segment to a card and put in a column to indicate it position in the workflow**.**
* **Limit WIP** 
  + Limit the number of items (number of items that can be in progress) assign to a specific segment.
  + i.e. Maximum WIP to limit the number of task or item that can enter a column in the Kanban board.
  + Minimum WIP to specify the minimum amount of task or item that should be attributed to a column.
* **Measure the Lead Time**
  + Lead Time, also known as cycle time is the average time to complete one task or item. Measure the Lead Time and optimize the process to make the Lead Time as small and predictable as possible.

Fig of the **Kanaban Table**:

WIP = work in progress

MAX = maximum

MIN = minimum

|  |  |  |
| --- | --- | --- |
| **TO DO**  WIP MAX = 10 | **Being DONE / IN PRORES**  WIP MAX = 5 | **DONE / Terminated**  WIP MIN = 1 |
| **TASK 4**  **TASK 5**  TASK 6  **TASK 7**  **TASK 8**  **TASK 9**  **TASK 10**  **TASK 11** | **TASK 2**  **TASK 3** | **TASK 1** |

**Benefits of Kanban:**

* Kanban make people to collaborate to optimize workload termination, rather than each person work on it own.
* Increases the visibility on everything that is going on at the company, or in the group.
* Reduces inventory, thereby reducing company costs.
* Overproduction of inventory is avoided, thereby saving resources and time as well.

**Characteristics of Kanban:**

* **Flexibility in Planning:**

Kanban provides improvements in the workflow, With visual representation of the workflow, speed of moving from one task to another is reduced.

* **Limits Work-In-Progress (WIP):**

Explicit limits are assigned to number of items or task that can be in progress at each workflow state, indicated by a column.

* **Pull Approach:**

When you have two teams and the first one is performing better than the second one, it is likely that it pushes more work than the other can actually handle. This often creates friction between the teams. A solution to this is the Pull approach.

In Pull Approach, the next team pulls work only when it is ready for it. Pull Approach is implemented by adding a buffer with limited capacity between the two teams.

* **Minimize Cycle Time:**

The cycle time for each task is measured and the process is optimized to reduce the cycle times. The bottlenecks are identified immediately and resolved collaboratively by the entire team**.**

* **Continuous Delivery:**

Short release cycles result in continuous delivery of growing product at regular intervals.

Continuous interactions with the customer.

Focus is on finishing work than on starting work.

* **Visual Metrics:**

Visually organized workflows (on Kanban Boards).

**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 (termed as waste)

**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** :

* **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 :**

*  **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** :

*  **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 -

o Improve visibility

o Reduce dependencies

o Enable easy flow of work

o Focus on the cycle-time of delivered work

o 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 -

o Take decisions so that they do not have to wait for instructions

o Have access to information so that it can be used as and when required

**AGILE KANBAN**

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 –

o Team is allowed to take decisions as and when required.

o Team collaboratively resolves the bottlenecks.

o Team has access to the relevant information.

o Team continually communicates with customer.

 Continuous Delivery: This means –

o Focus on work completion.

o Limited requirements at any point of time.

o Focus on delivering value to the customer.

o 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 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.

**Kanban and Scrum – Differences**

|  |  |  |
| --- | --- | --- |
| The differences between Kanban and Scrum are as follows –  **Sr. 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 (cycle time) 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. |

**KANBAN TOOL**

Kanban tool is a visual project management tool. Use Kanban cards, colors, swim-lanes, tags and due dates to compose work on Kanban board. Analyze and constantly improve your process to increase business efficiency.

Following are the important features of the Kanban tool -

 Online Kanban Boards

 Insightful analytics

 Visual Project Management

 Online Documents

 Drag & Drop Tasks

 To-Do Lists

**Kanbanery**

Kanbanery is a visual project management tool that helps you work more effectively, alone and together, by visualizing work.

Features of Kanbanery include –

* GitHub integration
* Create or copy task boards easily with templates
* iPhone and iPad apps

 API and several third party apps

* Advanced reporting
* Content-rich tasks
* Work with existing systems
* Real-time updates

**KANBAN EXAMPLE:**

**Kanban project: Date a girl and have sex.**

|  |  |  |
| --- | --- | --- |
| **TO DO**  WIP MAX = 10 | **Being DONE / IN PRORES**  WIP MAX = 5 | **DONE / Terminated**  WIP MIN = 1 |
| **TASK 9: have sex with her**  TASK 7: Ask her out for a date  **TASK 8: bring her home**  TASK 6: Program an outgoing | **TASK 5:Get her phone number**  **TASK 4: make her laugh, make her tell you thing about her**  **TASK 3: greeting and flirt** | **TASK 1: Physical Presentation**  **TASK 2: Find a girl** |