Kanban

Fall 2025

Project management

- Companies have teams and groups.
  - When you have groups, you must divide tasks.
- Without management:
  - projects miss deadlines
  - go over budget and
  - **FAIL**

Key scopes

- **Scope**
  - What exactly needs to be done?
- **Time**
  - When should each task be completed?
- **Resources**
  - Who and what do you have to finish t?
- **Risk**
  - What can go wrong?

Kanban

- A lightweight approach to manage work and improve flow
- Gives structure **without heavy process**
- Helps teams:
  - visualize work
  - communicate clearly
  - finish tasks instead of dealing with many

Kanban

- Developed by *Taiichi Onno* @ **Toyota** in late 1940s.
  - A way to optimize the company's manufacturing processes.
- **Kanban:** visual signal, card.
- Inspired by supermarkets.
  - Supermarkets stock their shelves based on customer demand.
  - They aim to minimize inventory and maximize efficiency.
- It is:
  - A system to visualize tasks
  - A way to manage flow
  - A communication tool
- It is **not**
  - scrum
  - strict meetings
  - heavy documentation

Kanban vs Scrum

- Both Agile frameworks.
- Scrum is based on **fixed-length** iterations (sprints)
  - With pre-defined roles and ceremonies.
- Kanban is more flexible and focuses on continuous flow.
  - For that reason, better for *bugfix* instead of product development.
- Kanban does not prescribe specific roles or meetings.
  - Allows teams to adapt the framework to their unique needs and context

Basic kanban board

- 4 columns:
  - Backlog
  - To Do
  - In Progress
  - Done
- In advanced team settings, we might also see:
  - Review / Testing
  - Blocked
  - etc.

Task Cards

- Each card should include:
  - Task title
  - Short (brief) description
  - Owner (the person assigned to it)
  - Priority or due date
- WIP limits
  - Although these are important in Kanban, I do not expect you to do them in your projects.
  - It forces the team to focus on *finishing* existing tasks before *starting* new ones.
    - **In short: There is a limit on the number of concurrent tasks you can do!**

Acceptance Criteria

- How do you know as task is **done**?
- AC: Define the **conditions** that must be met for a work item to be considered **Done**.
- Set of clear, testable statements that specify the required functionality or performance of a task from the **user's perspective**.
- They answer the question
    - How do we know this task is finished and works correctly?

AC

- It can be multiple.
- Imagine that we have a task:
  - Title: Implement Basic User Login Page
  - Desc: Create an HTML form with fields for Username/Email and Password, a Login button, and simple client-side validation.
  - Owner: Alice
- CA’s
  - CA1: **Given** the user navigates to /login URL. **When** the page loads. **Then** an input field labeled “Username or Email” and an input field labeled *Password* must be visible.
    - When the user navigates to /login and when the page loads, user must see an input field labeled *Username or Email* and an input field labeled *Password*
  - CA2: **Given** the user leaves the password field empty. **When** they click the Login button. **Then** a red error message “Password is required” appears below the password field, and the submission is halted.
  - CA3: **Given** the user provides valid credentials (e.g.test/test). **When** they click the *Login* button. **Then** the system redirects the user to *dashboard* page.

Focus

- Focus on finishing tasks already *in progress*
- Avoid starting new items when existing work is unfinished
    - Usually there is a WIP limit for this reason

Examples

- Check out Trello templates!