



# Agile Through the Eyes of a Software Engineer

Author: Peng Shao



# Agile Methodologies in Software Development

- Scrum
- Kanban
- XP (Extreme Programming)

# Kanban history



- Introduced in Toyota
- “To produce only what is needed, when it is needed and in the amount needed.”
- David Anderson popularised it for software
- Pull instead of push



# Kanban principles

- Change Management
  - Start with what you do now
  - Agree to pursue improvement through evolutionary change
  - Encourage acts of leadership at every level
- Service Delivery
  - Understand and focus on customer needs and expectations
  - Manage the work, let people self-organise around it
  - Evolve policies to improve outcomes



# Kanban (看板) practices

- Visualisation
- Limit Work in Progress (WIP)
- Manage workflow
- Feedback loops (stand up, retrospective, planning meeting)
- Explicit policies (documentation)
- Continuous Improvement



# Visualisation and Kanban board

- Managing work rather than people
- Everyone in the team can see what need to be done, and what is in progress

To do	In progress (limit = 3)	Test (limit = 2)	Done
E F G H I J K	C D	B	A



# Kanban board

Jira board

Trello board

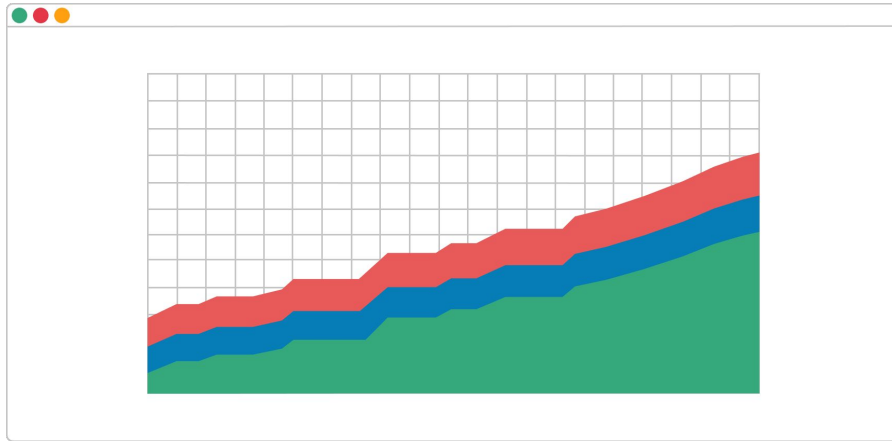
Gitlab board

Physical whiteboard

And many more

# Kanban Metrics

## Cumulative Flow Diagram



<https://kanbanize.com/kanban-resources/kanban-analytics/cumulative-flow-diagram>





# Kanban metrics

- Lead time
- Throughput
- WIP



# Real Life Application of Agile Workflow

- Every team is different
- The clash between the team's work process with the company's
- The agile workflow need to involve at different stages of a team



# Remote Working with Agile

- Advantages
  - Hiring from around the world
  - Flexible working arrangement
  - Results of work vs hours put in
- Disadvantages
  - Onboarding can be difficult
  - Loneliness
  - Mixing personal life with work



# Q & A