

CSC301

Kanban

Kanban

- Kanban is another agile process
 - Designed to improve (and maintain a high) level of production
 - Lighter and less-prescriptive than Scrum
 - Gaining popularity amongst software teams

Origins

- Kanban originates from Toyota's efforts to perform Just In Time manufacturing.
- “Pull” more inventory when a cell runs low.
 - Helps keep inventory to a minimum
- Jargon:
 - Parts kept in bins on shop floor, factory inventory, and at supplier

Just in Time Manufacturing

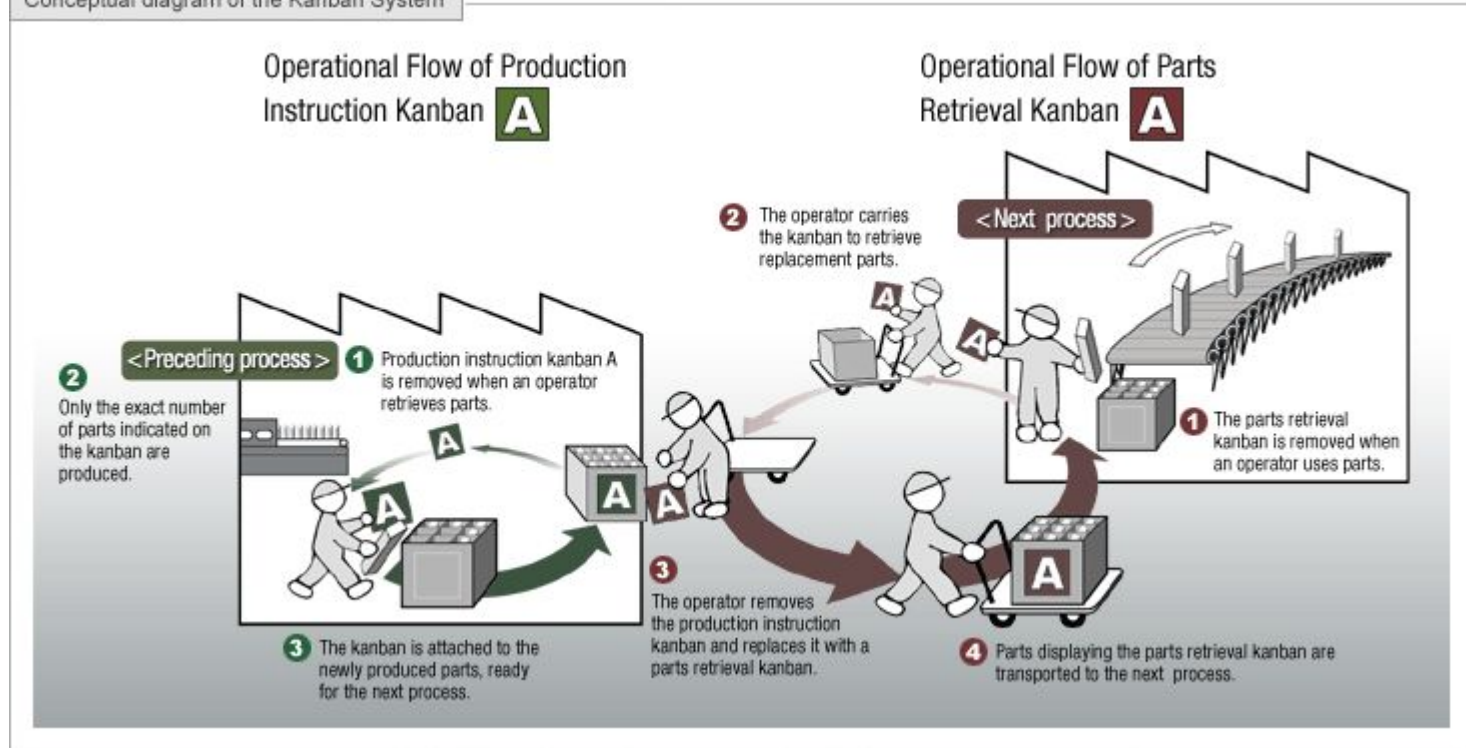
- When bin on floor runs out, fetch bin from storage
- When storage depleted, fetch bin from supplier
- When supplier inventory runs out, make more
- Kanban “card” sitting at bottom of each bin
 - with product details

Just in Time manufacturing

- Work is done in a pipeline
- Each stage of pipeline needs inventory
- As inventory runs out, signal preceding stages to get more

Toyota illustration of kanban

Conceptual diagram of the Kanban System



Kanban for software

- No prescribed roles
- No prescribed meetings
- Just one simple concept ...

Kanban

Think of the development process as a pipeline

- Feature requests come in
- Improved software comes out



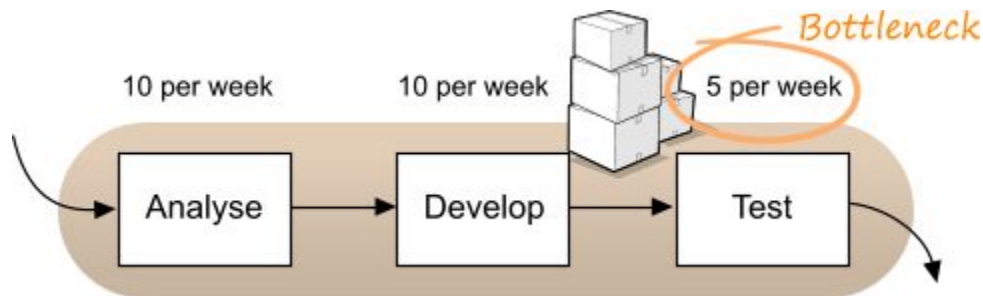
Goal: Maximize throughput

<http://kanbanblog.com/explained/>

Kanban

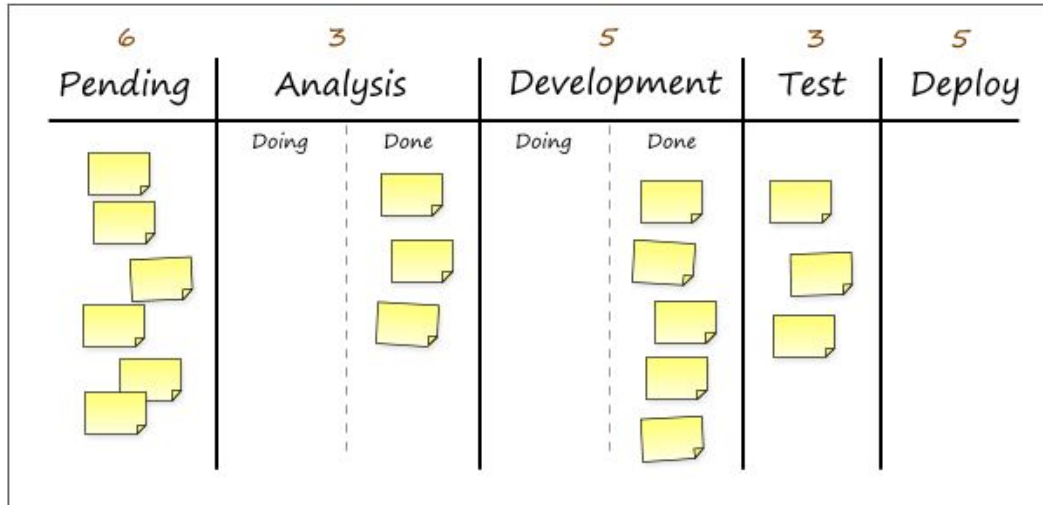
Items go through the pipeline in steps

- The number of steps and their names, can differ between teams and/or projects
- Throughput of the pipe is limited by the bottleneck



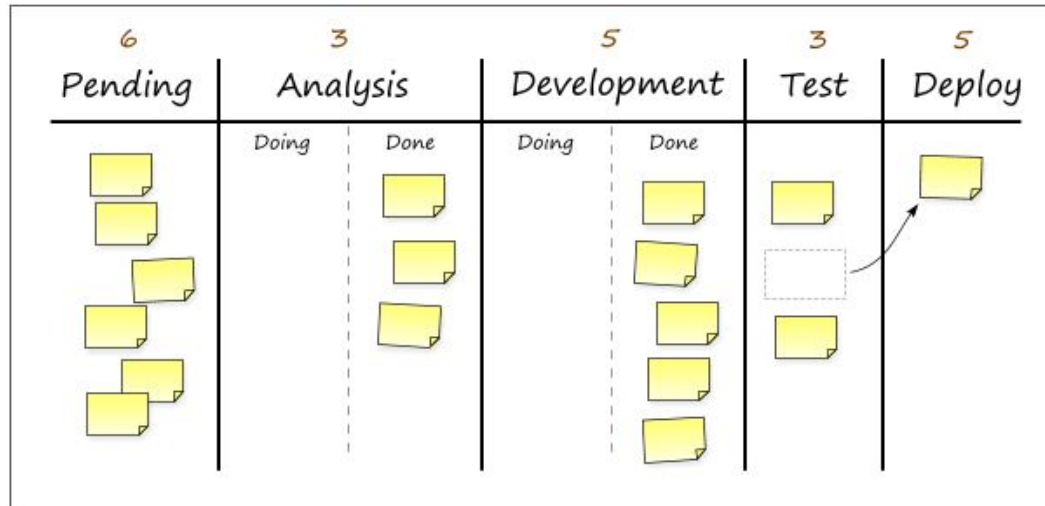
The Kanban Board

- Columns represent pipeline steps
- Sticky notes represent items



The Kanban Board

- Items flow from left to right
- Limit the number of items in each column



The Kanban Board

- Set a limit for the amount of work-in-progress (WIP) that may accumulate at each step.
- When items pile up in a specific step
 - There exists a bottleneck downstream ...
 - Get other team members to help
- We measure *lead time* - The time it takes for an item to make it through the board

The Kanban Board

- Use whiteboard + sticky notes
- Or one of the many available software tools
- Or even something as simple as GitHub issues
 - Labels to indicate columns
 - Search to see which issues are in a column
 - At a given point in time
 - During a time range
 - Developers should respect the WIP limit(s)

Kanban vs. Scrum

- Similarities
 - Self-organizing team
 - Break work into tasks
 - Developers pull tasks
 - Transparent processes
 - Frequent delivery
- Both are agile processes

Kanban vs. Scrum

- Kanban doesn't have prescribed roles
 - What about product owner's responsibilities?
 - Who facilitates the process?
- Up to the team to divide responsibilities
 - Ex: Prioritizing tasks and decide which items go on the Kanban board first

Kanban vs. Scrum

- Scrum uses fixed-length sprints
 - Start with planning meeting (Scrum board is reset)
 - Continue with daily meetings
 - End with review & retrospective meeting
- Kanban is an ongoing process
 - Board is never reset
 - No meetings prescribed
 - The team decides on the frequency, duration and nature of its meetings

Kanban vs. Scrum

- Estimating task size
 - Prescribed in Scrum, optional in Kanban
- Charts
 - Burndown in Scrum, no prescribed chart in Kanban
- Metric
 - Velocity in Scrum, lead-time in Kanban
- Limit Work-In-Progress
 - Implicit in Scrum, explicit in Kanban

Kanban vs. Scrum

- Kanban is lighter & less prescriptive
 - Less prescribed meetings and/or artifacts
 - Less facilitation required
- Kanban fits well with continuous deployment
 - Release when an item makes it to the last column
 - No need to wait until the end of the sprint
- Kanban gives more freedom (and leaves more decisions) to the team

The Trend

- Historic trend
 - From Waterfall to XP
 - From XP to Scrum
 - From Scrum to Kanban
- Process are
 - Becoming lighter
 - Less prescriptive
 - Supporting more frequent delivery