

Origins of Lean

- This method came from Toyota. Developed a lean production system in the 1950s
- Poppendieck and others applied principles of lean production to create Lean Software Development
- Piggly Wiggly story: first self-service grocery store only reorders goods when customers had almost depleted current stock. Toyota realized that they could use this just in time strategy for manufacturing cars

Lean development

- Continuously improve, create value for the customer, eliminate waste, optimize value streams, empower people
 - **Principles of Lean:** Eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, and see the whole
1. Eliminate Waste: anything that does not add value to the customer is waste (unnecessary features)
 2. Amplify learning: learn and apply improvements as soon as possible; reading and refactoring code help; short iterations provide helpful feedback from customers, so both developers and customers learn
 3. Decide as late as possible: premature decisions may need to be undone later, which creates waste
 4. Deliver asap: just in time production can be applied in software development; allow teams to self-organize so that they can deliver asap
 5. Empower the team: developers should provide their own estimates of effort; choose their own process, choose their own tools. Management should facilitate, not dictate
 6. Build integrity: invest time and effort to build a good product rather than providing a bad customer experience
 7. See the whole: don't isolate

Lean Practices

- **Value stream mapping:** diagram the flow of goods and/or information. Current map and future map. Identify waste that should be removed from the current map and changes needed to establish the future map
- **Set based development:** choose several designs that will satisfy the customer requirements; invest some time exploring all the alternatives, perhaps even implementing prototypes; eliminate alternatives as you gain experience and feedback
- **Kanban Method:** anything that slows the software delivery pipeline is wasteful. Kanban helps identify waste and optimize processes; helps the team prioritize work and focuses on process improvement.
- **Kanban Boards:** visual display of items at each stage of the process. Pull a task through the flow when capacity is available. Don't push a task through; prioritize to limit the # of items in each queue at any time and focus on flow of value: delivering items with little value quickly doesn't help
- Successful kanban: visualize the workflow, limit WIP, identify bottlenecks, manage lead time (the average time to pull an item to working on it)
- Focus on overall flow of work through the process rather than individual team member utilization

- Kanban key principles: foster leadership at all levels of the org, start with what you do now; pursue incremental changes to existing process;
- Make process policies explicit - improve collaboratively and continuously
- Scrum vs kanban: scrum and kanban may be complementary or competitors



Scrum vs Kanban Method

Kanban	Scrum
No prescribed roles	Product Owner, Scrum Master, Developers
Continuous delivery	Time boxed sprints
Pull work through system	Pull work through in batches
Changes can be made at any time	Define sprint, then don't allow changes
Measure cycle time	Measure velocity
Ideal for high variability	Ideal for batch deliveries

-
- Scrumban: scrum features: Product owner, scrum master, include kanban techniques in your scrum process
- Scrum planning may be inefficient and wasteful
-