Lean Software’s Main Idea

Reduce Waste

Waste
- Anything not delivered to the customer
- Artifacts that do not deliver a business value

Lean software inspired by
- Toyota’s “lean manufacturing” industrial production
- Just-in-time production
Lean Software’s Principles

1. Eliminate Waste
   - Anything not delivering business value
   - E.g. requirements documents, partially done work, rarely used features (bloat), bugs, task switching, waiting

2. Amplify learning
   - “Try-it, test-it, fix-it” rather than “do it right the first time”
   - Short iteration cycles

3. Decide as late as possible
   - Avoid up front design decisions, make choices when information is available

4. Deliver as fast as possible
   - Working system at every iteration, fast feedback cycle
Lean Software’s Principles

5. Empower the team
   - Motivate the team, self-organization
   - “find good people and let them do their own job”

6. Build integrity in
   - Maintain the consistency of a system’s design
   - E.g. through refactoring, automated tests, complete build system

7. See the whole
   - Focus on overall progress of the project
   - Strong common sense
Lean Software Summary

- Software development can benefit from industrial production recipes
- However, software has no production, only design
- “Lean” can be seen as more philosophy than method
- Reminder to look out for waste of any kind

“Lean” is applicable in many sectors, e.g. lean startups
Kanban
(看板)
At full capacity, there is little throughput (flow is constricted by bottlenecks)
Kanban’s Main Idea

Minimize Work In Progress

- “Stop Starting Start Finishing”
- Ensure just-in-time production
- Kanban: literally “signboard” or “billboard” in Japanese
- Inspired by Toyota
- Visual process-management approach (“Kanban boards”)
Core Kanban Practices

1. Limit work in progress
   - Limit amount of tickets per column
   - Focus on most productive task for the project
   - Pull work from previous columns
   - Reduce context switching (waste)

2. Visualize
   - Shared Kanban-Board with process steps as columns
   - Requirements (tasks, user stories,...) travel as notes from left to right.

3. Manage flow
   - Measure length of queue, average cycle time and throughput
   - Identify bottlenecks and allow planning

There are many solutions for digital Kanban boards, it feels significantly different to move physical post-its though.
Core Kanban Practices

4. Make policies explicit
   - Create explicit shared understanding of rules and assumptions
   - E.g. what columns mean, Definition of Done, which ticket to pull next

5. Implement feedback loops
   - Process of continuous improvement ("kaizen" in Japanese)
   - Don’t wait for feedback, build it into the process

6. Improve collaboratively, evolve experimentally
   - Try things out, evaluate
Flow: from backlog to delivery

Kanban limits

Pull

Subdivisions by subject / swim lanes
One day in Kanban Land

(A comic by Henrik Kniberg)
A & B are the 2 most important things right now!

Great! Next is C & D!

Hmm... might need J+K+L as well

Ah, something to deploy!

We'll do A

And we'll do B

Crap! A doesn't build!
Metrics — Cumulative Flow Diagram

**Lead Time:**
Time from ticket being placed on board to ticket delivered

**Cycle Time:**
Time from starting work on ticket to finishing
References