

Kanban vs Scrum

A practical guide

Deep Lean, Stockholm
May 19, 2009

Henrik Kniberg - Crisp AB

Agile coach & Java guy

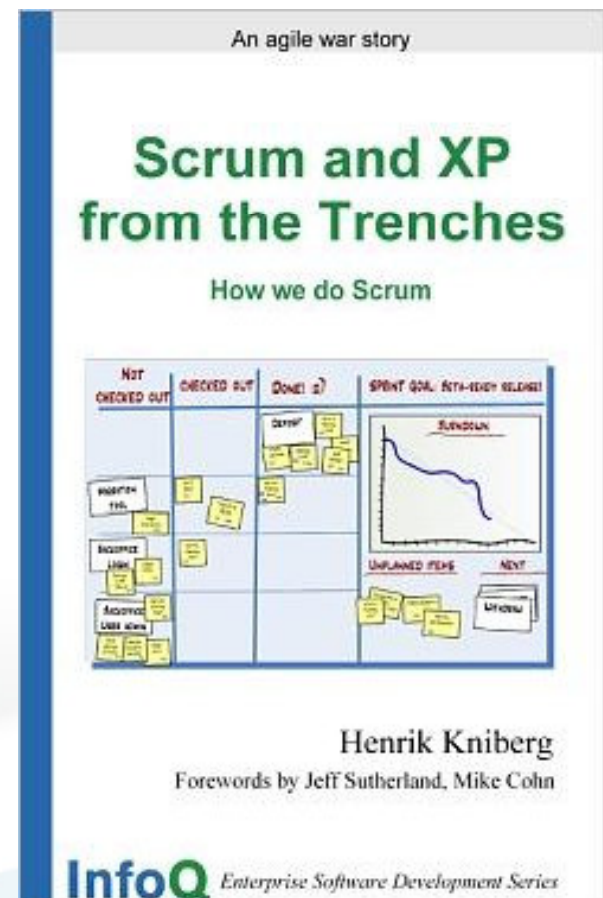
Cofounder / CTO of Goyada (mobile services)
30 developers

Lead architect at Ace Interactive (gaming)
20 developers

Chief of development at Tain (gaming)
40 developers

Agile coach at various companies

henrik.kniberg@crisp.se
+46 70 4925284



Introduction

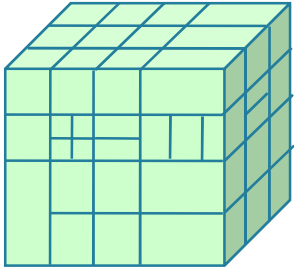
Purpose of this presentation:

Clarify Kanban and Scrum by comparing them

...so you can figure out how these may come to use in your environment.

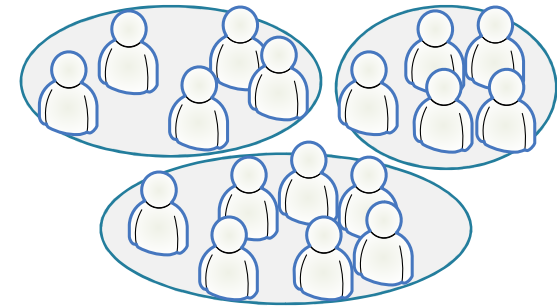
Scrum in a nutshell

Split your product

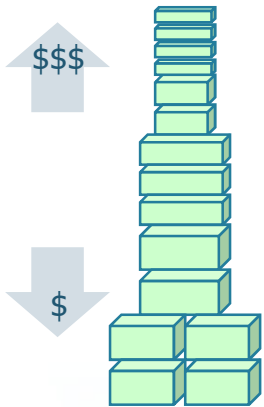


~~Large group~~ spending ~~a long time~~ building a ~~big thing~~
Small team spending a little time building small thing
... but integrating regularly to see the whole

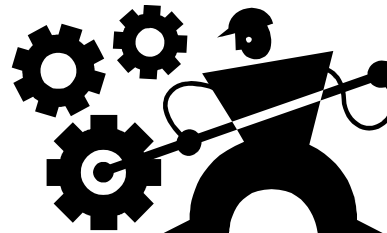
Split your organization



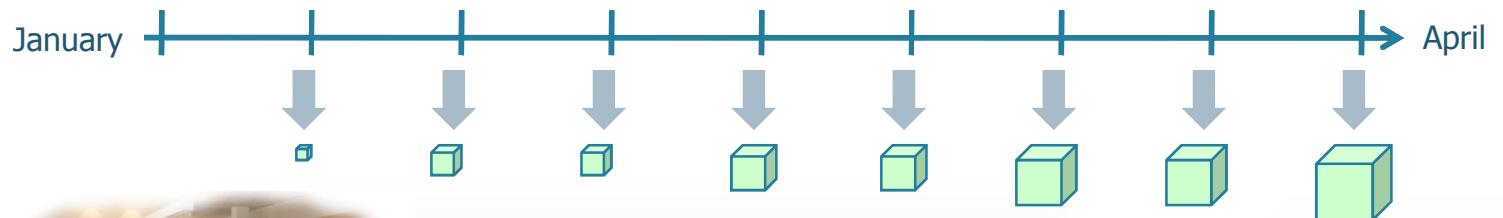
Optimize business value



Optimize process



Split time

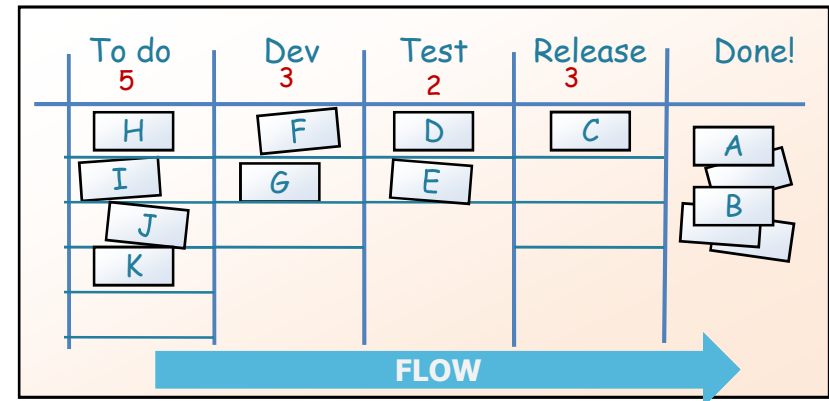


Henrik Kniberg

crisp

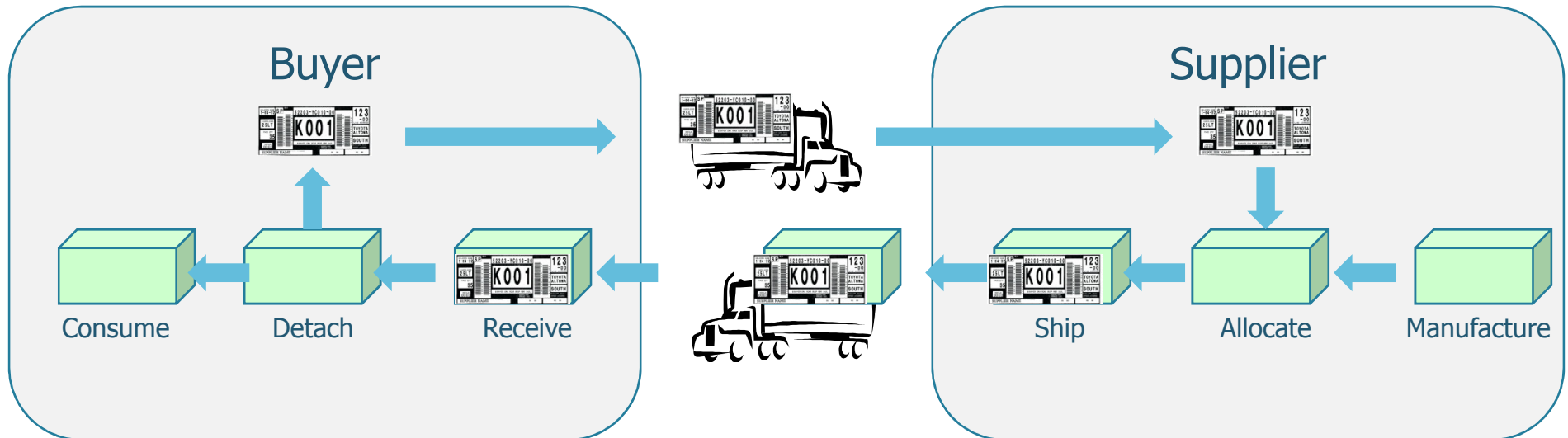
Kanban in a nutshell

- **Visualize the workflow**
- **Limit WIP** (work in progress)
- **Measure & optimize flow**



Roots of Kanban (Toyota)

看板
Kan Ban
"Visual Card"



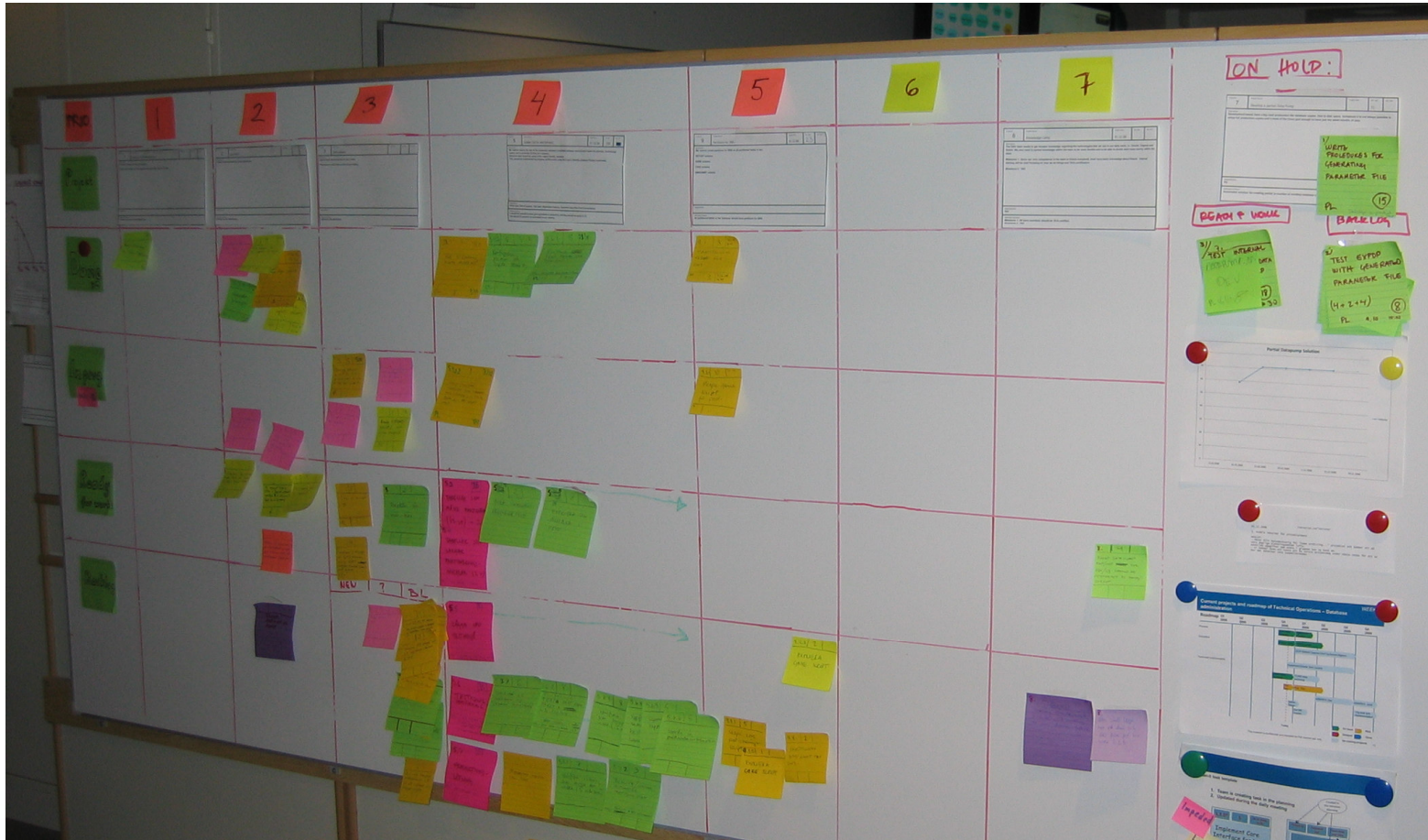
The two pillars of the Toyota production system are just-in-time and automation with a human touch, or automation.

The tool used to operate the system is kanban.



Taiichi Ohno
Father of the Toyota Production System

Kanban in software development



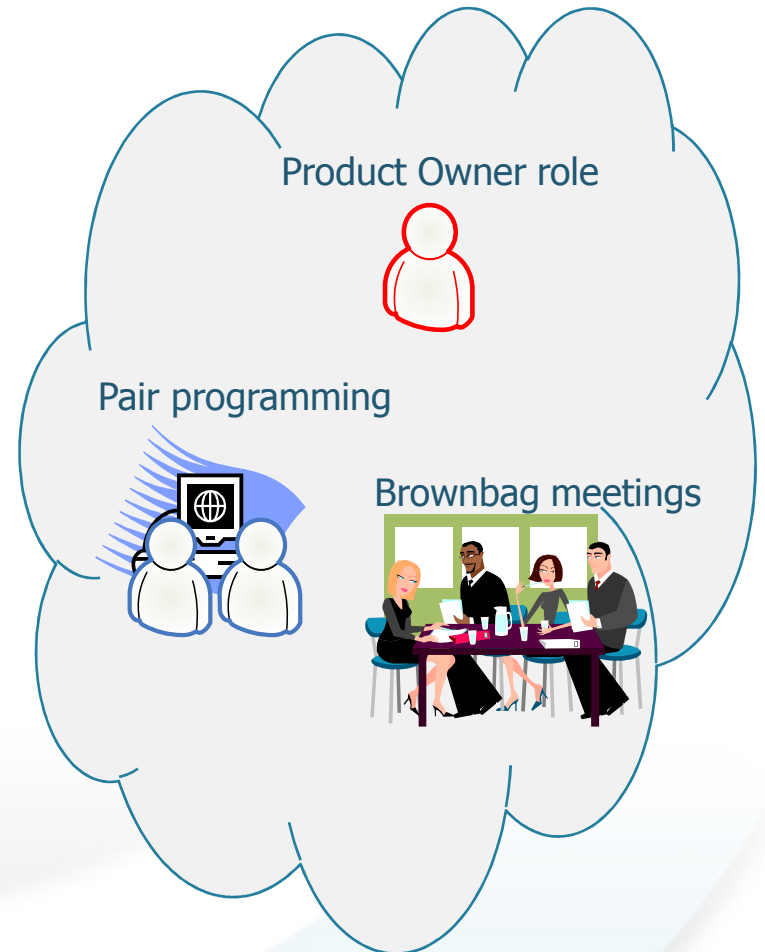
Kanban and Scrum are both process tools

Physical tools



Process tools

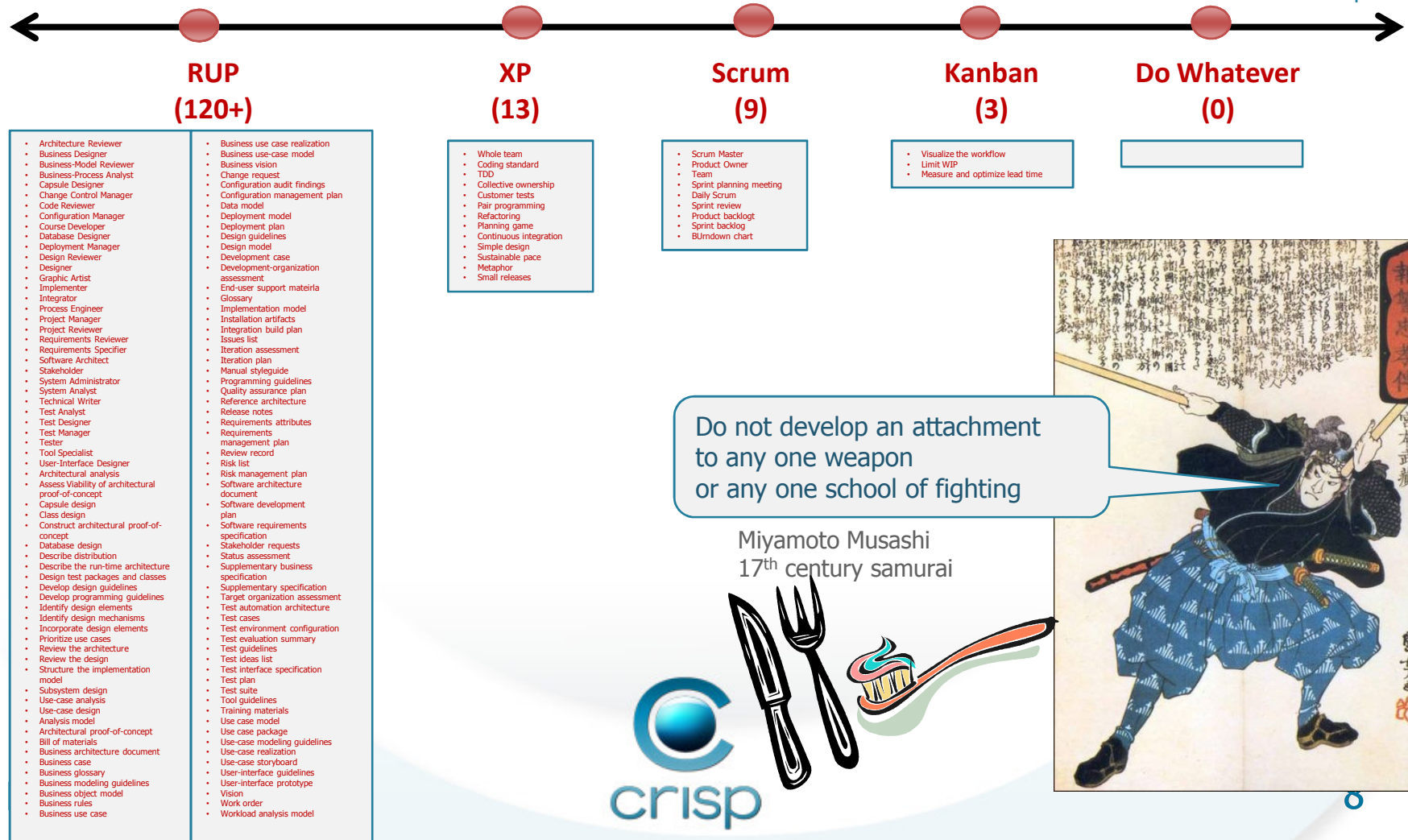
a.k.a. "organizational patterns"



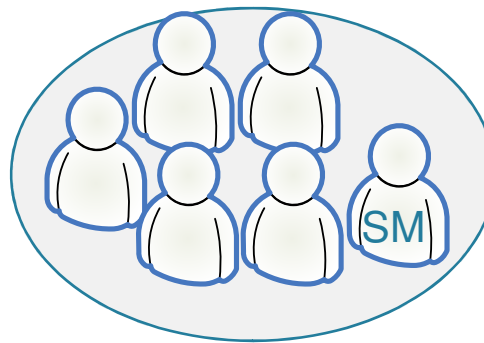
Prescriptive vs adaptive

More prescriptive

More adaptive



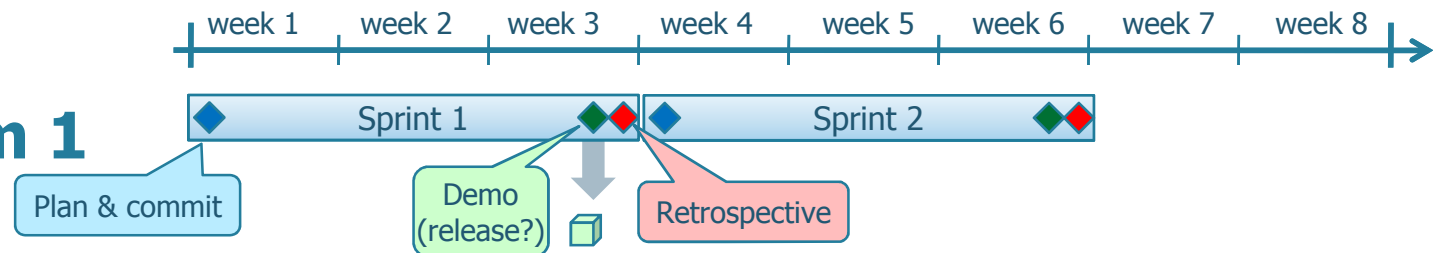
Scrum prescribes roles



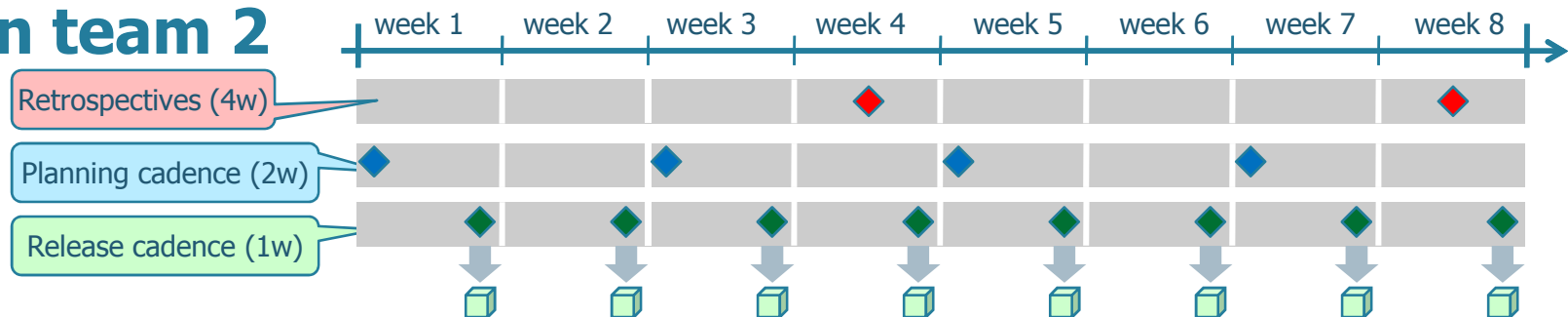
Scrum prescribes iterations

Scrum team

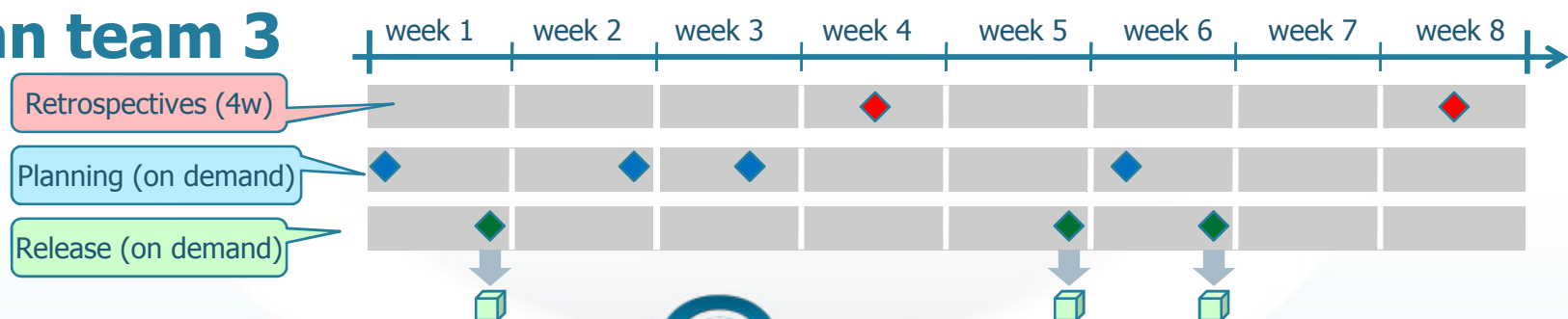
Kanban team 1



Kanban team 2

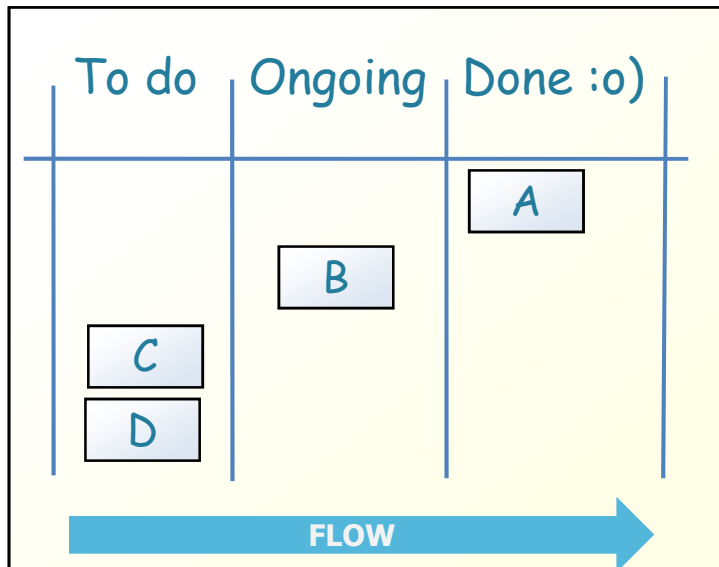


Kanban team 3



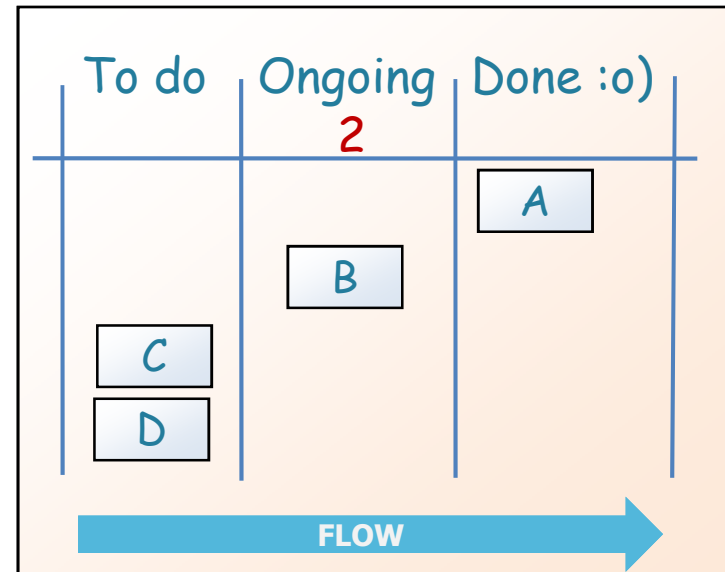
Both limit WIP, but in different ways

Scrum board



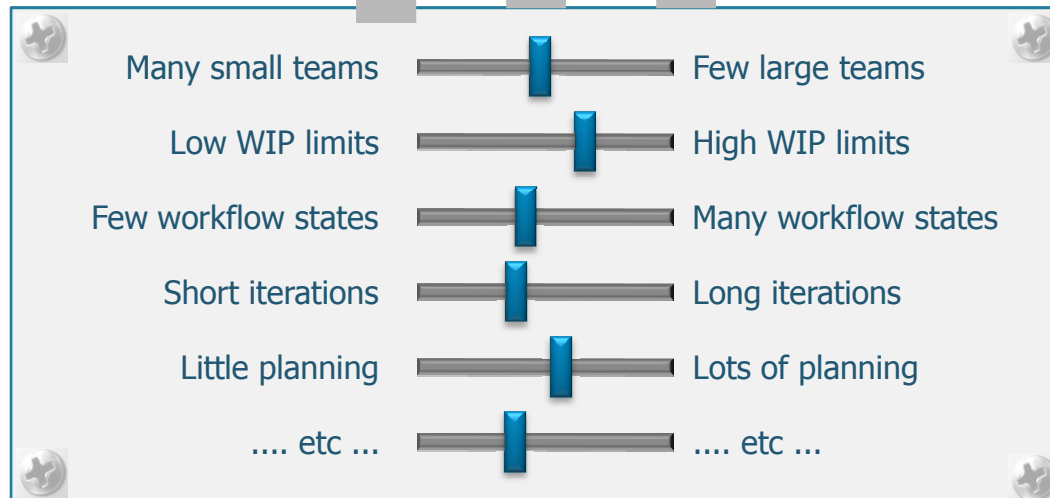
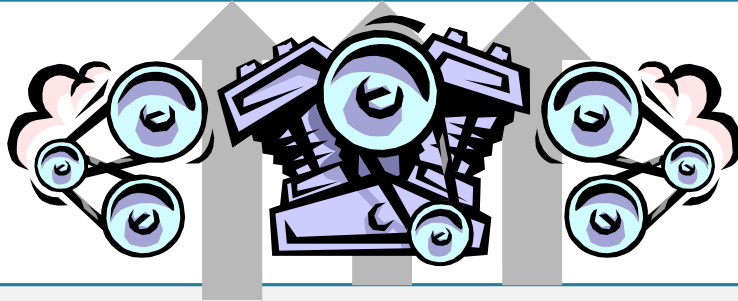
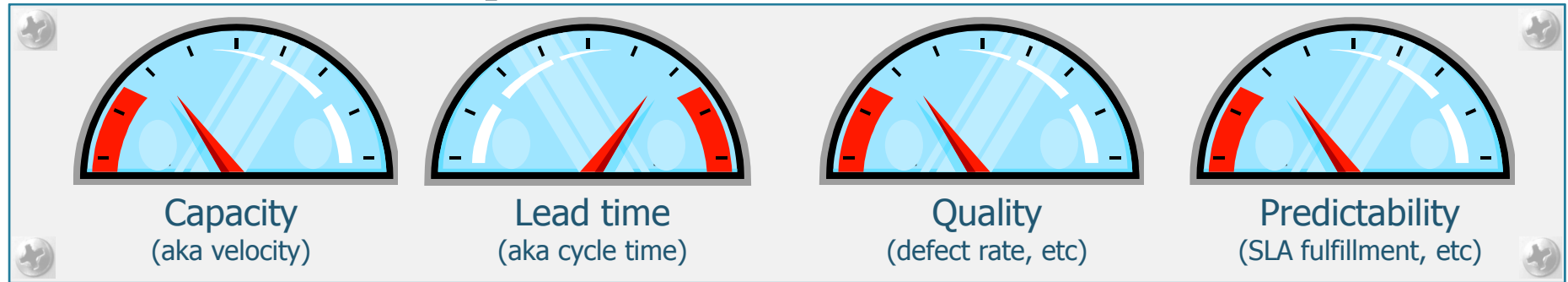
WIP limited per unit of time
(iteration)

Kanban board



WIP limited per workflow state

Both are empirical



Henrik Kniberg

crisp

Kanban is more configurable

Great! More options!



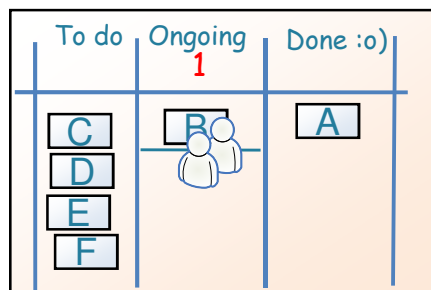
Oh no, more complicated!





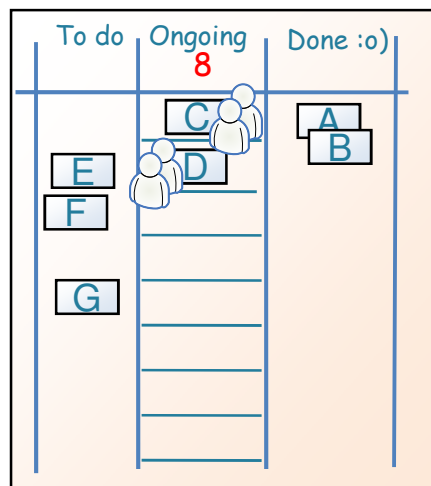
Example: Experimenting with WIP limits

Monday, Week 1

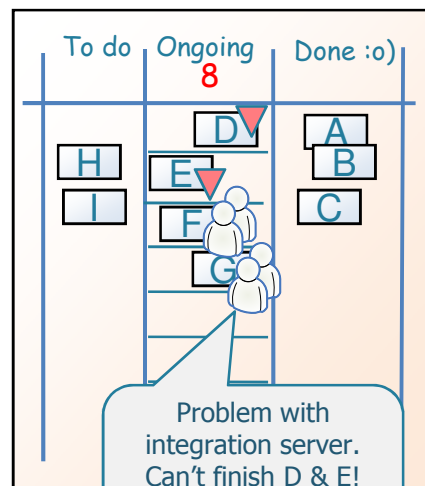


We're idle & bored!
Let's increase WIP
limit to 8!

Monday, Week 2

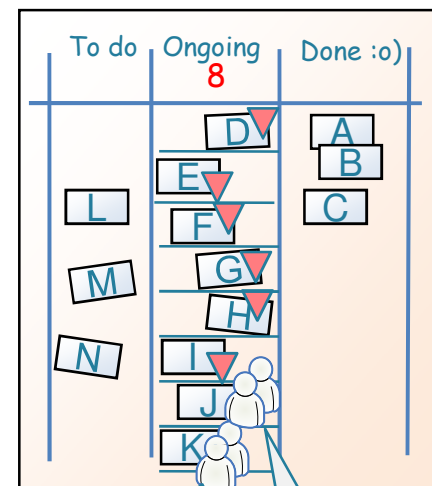


Monday, Week 3



Problem with
integration server.
Can't finish D & E!
We'll work on F & G
instead!

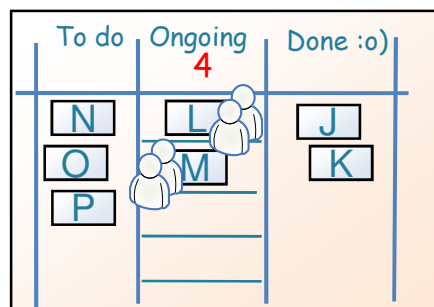
Monday, Week 4



Oops. WIP limit
reached. Now we
HAVE to stop and
fix the integration
server!

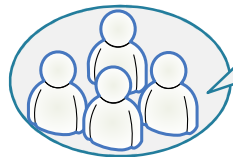
Let's reduce WIP
limit to 4, so we
react earlier next
time!

Monday, Week 5

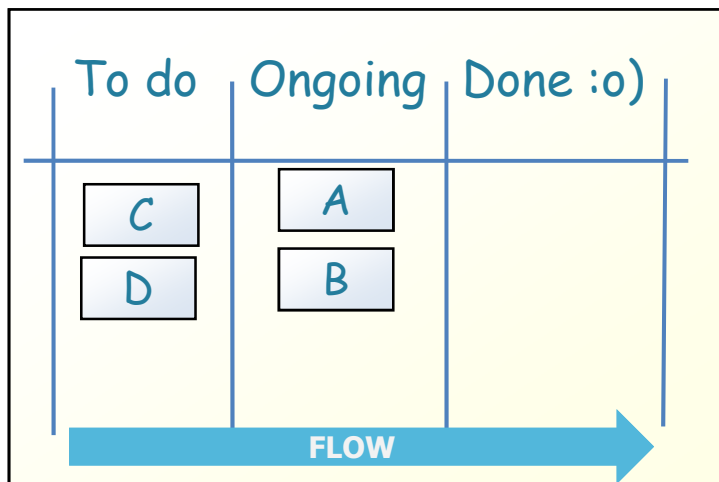


Scrum doesn't allow change in mid-iteration

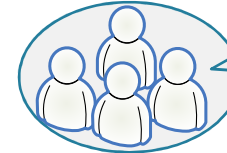
Scrum



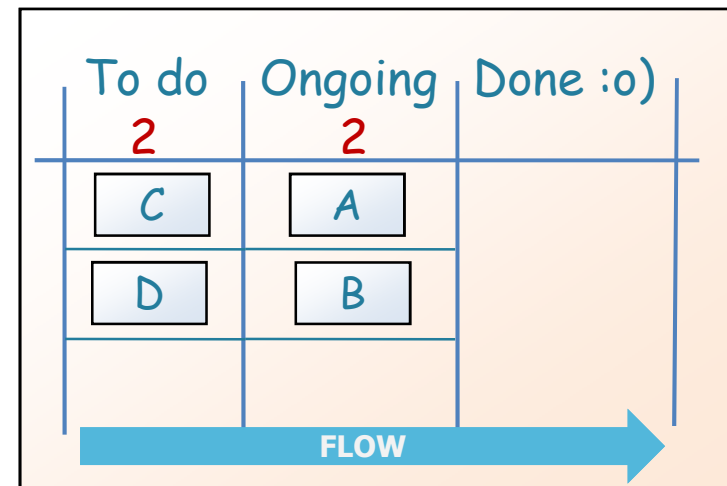
Wait until next sprint!



Kanban



Wait until a To Do slot becomes available!
Or swap out C or D!

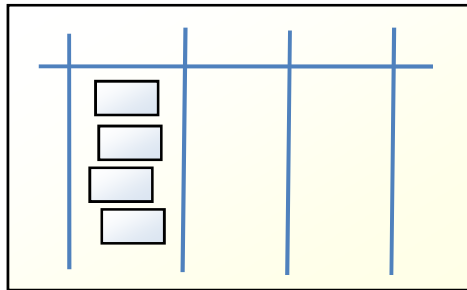


I'd like to have E!

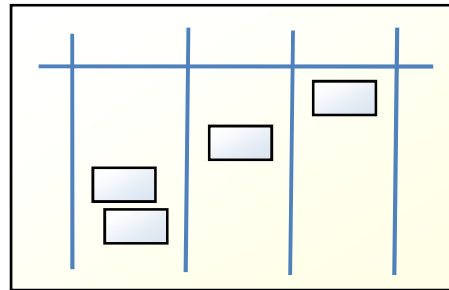
Scrum board is reset between each iteration

Scrum

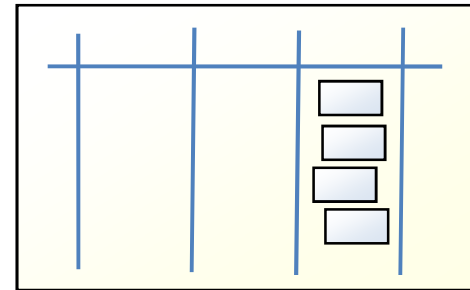
First day of sprint



Mid-sprint

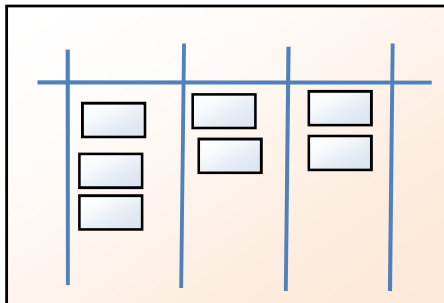


Last day of sprint



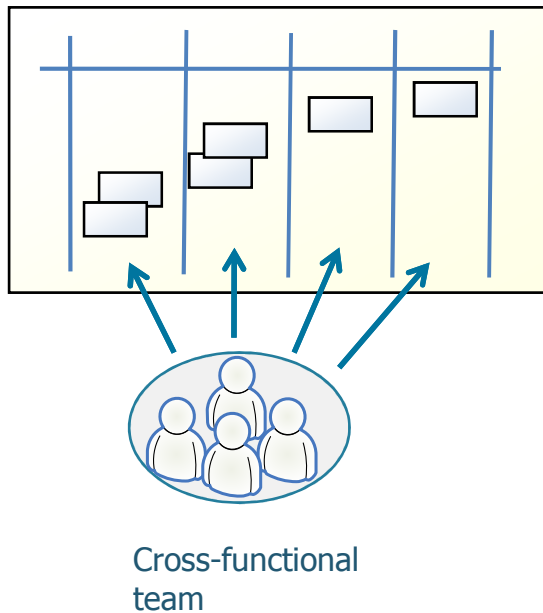
Kanban

Any day

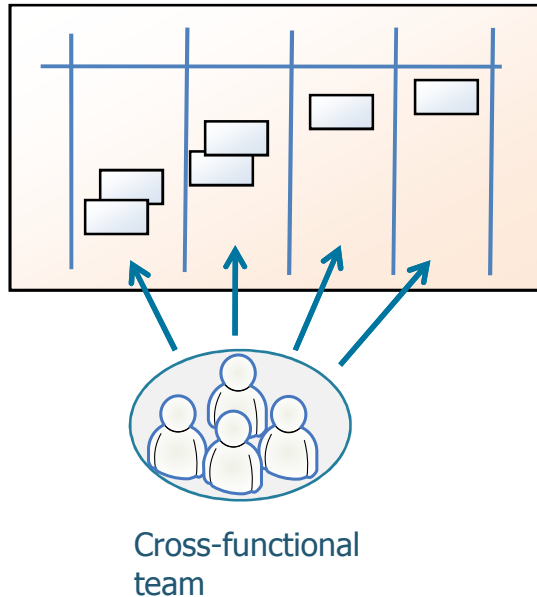


Scrum prescribes cross-functional teams

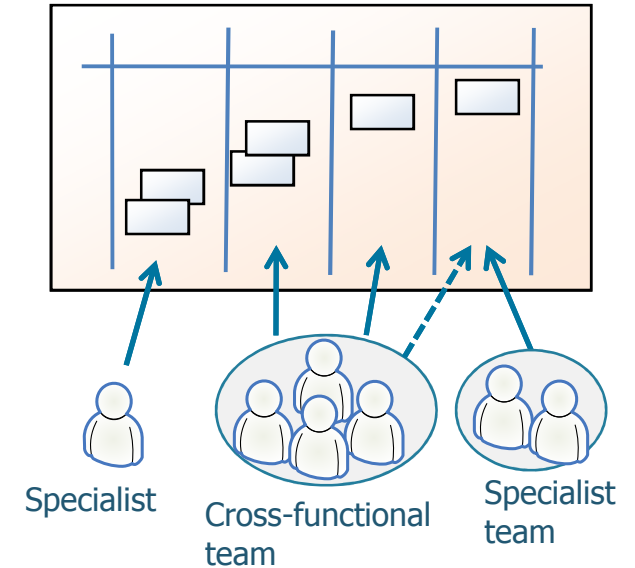
Scrum



Kanban – example 1

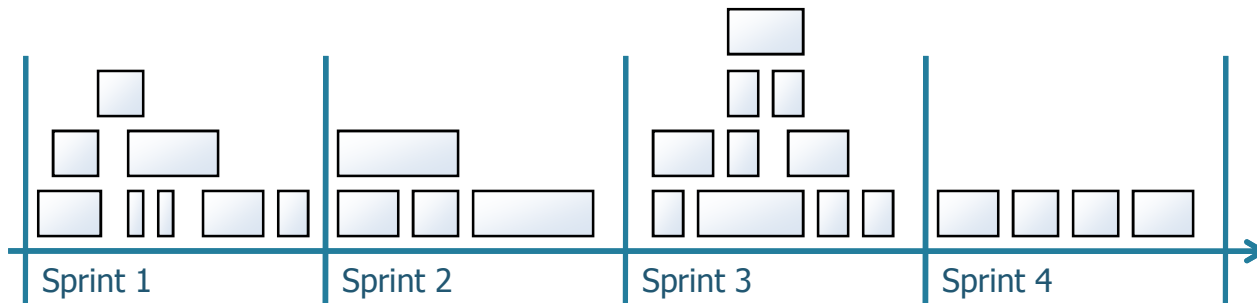


Kanban – example 2

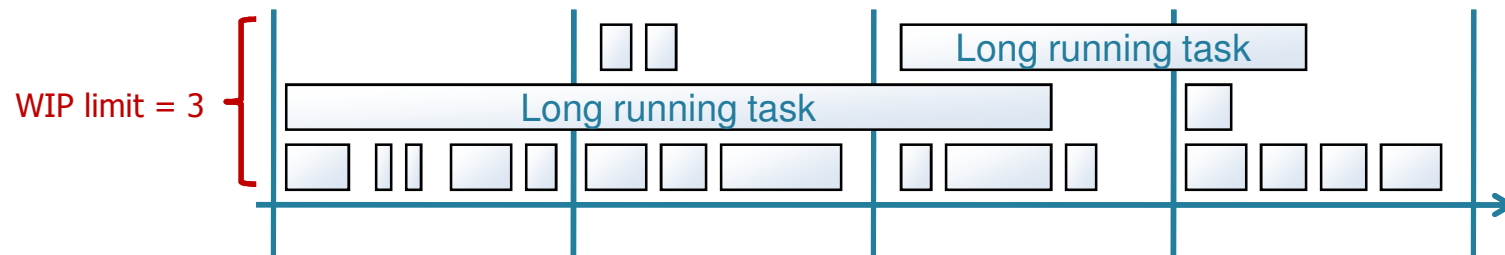


Scrum backlog items must fit in a sprint

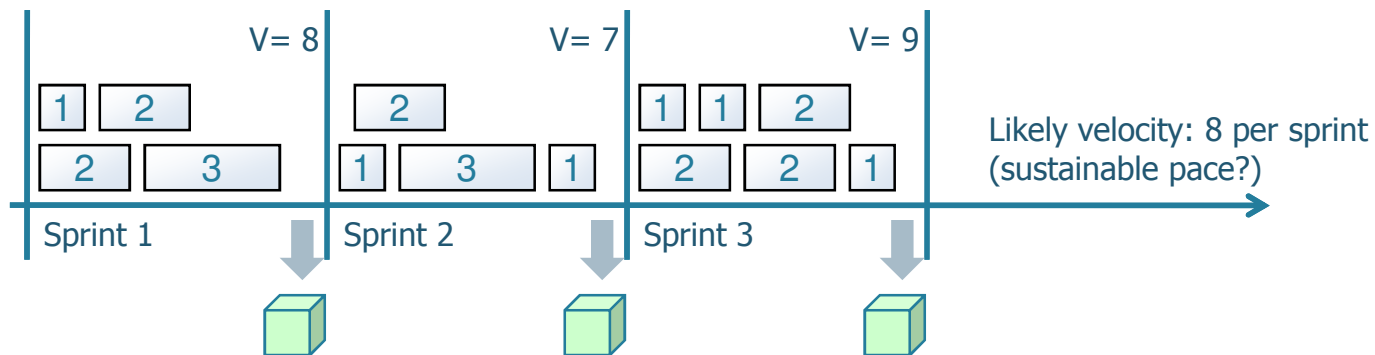
Scrum



Kanban

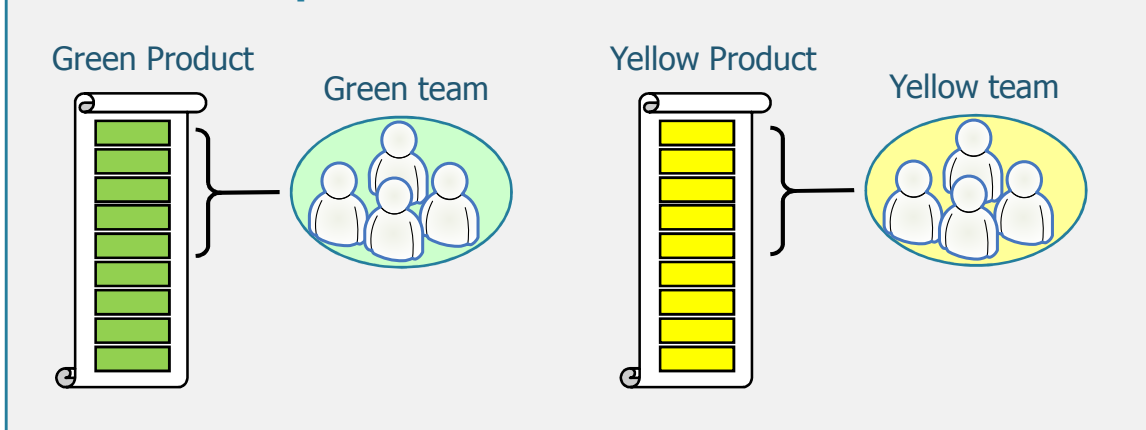


In Scrum, estimation and velocity is prescribed



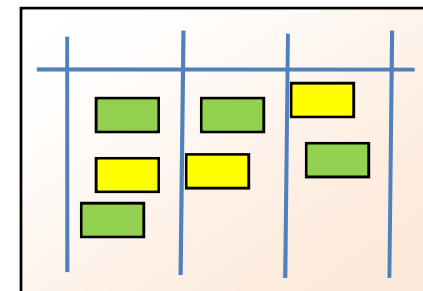
Both allow working on multiple products simultaneously

Scrum example 1

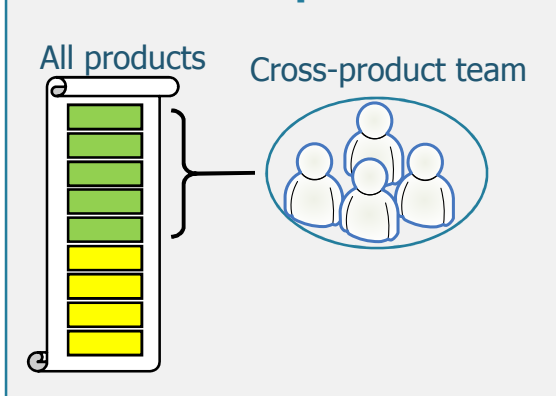


Kanban example 1

Color-coded tasks

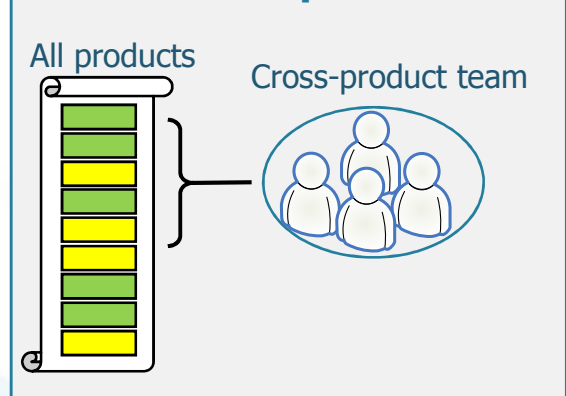


Scrum example 2



Henrik Kniberg

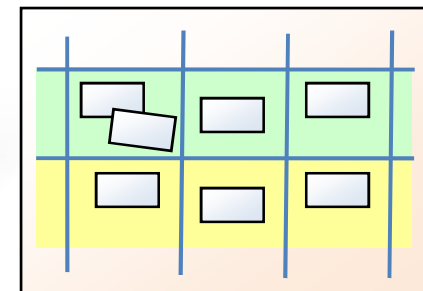
Scrum example 3



crisp

Kanban example 2

Color-coded swimlanes

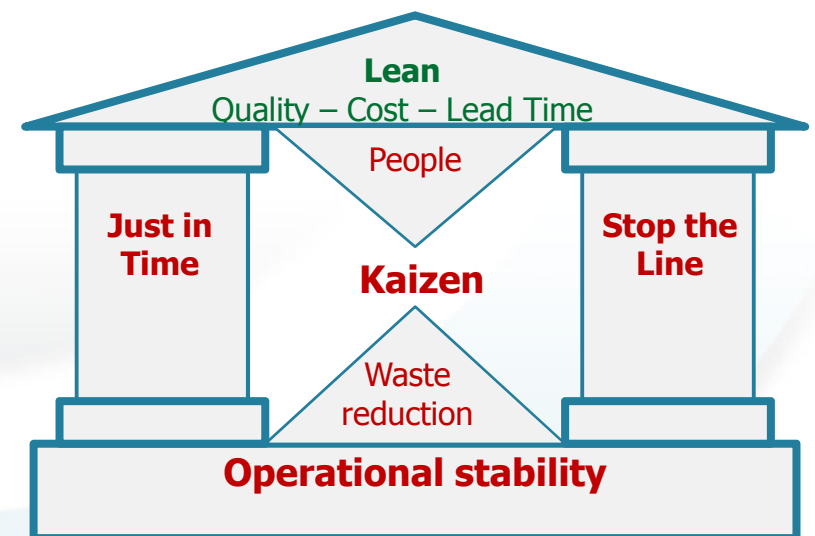


Both are Lean → and Agile ↓

1. **Individuals and Interactions** over Processes and Tools
2. **Working Software** over Comprehensive Documentation
3. **Customer Collaboration** over Contract Negotiation
4. **Responding to Change** over Following a Plan

1. Base your management decisions on a **Long-Term Philosophy**, Even at the Expense of Short-Term Financial Goals
2. Create **Continuous Process Flow** to Bring Problems to the Surface
3. Use **Pull** Systems to Avoid Overproduction
4. Level Out the Workload (**Heijunka**)
5. Build a Culture of **Stopping to Fix Problems**, to Get Quality Right the First Time
6. **Standardized Tasks** are the Foundation for Continuous Improvement and Employee Empowerment
7. Use **Visual Controls** So No Problems are Hidden
8. Use Only **Reliable, Thoroughly Tested Technology** That Serves Your People and Processes
9. **Grow Leaders** Who Thoroughly Understand the Work, Live the Philosophy, and Teach It to Others
10. **Develop Exceptional People and Teams** Who Follow Your Company's Philosophy
11. **Respect Your Extended Network** of Partners and Suppliers by Challenging Them and Helping Them Improve
12. Go and See for Yourself to Thoroughly Understand the Situation (**Genchi Genbutsu**)
13. **Make Decisions Slowly** by Consensus, Thoroughly Considering All Options; Implement Decisions Rapidly
14. **Become a Learning Organization** Through Relentless Reflection (**Hansei**) and Continuous Improvement (**Kaizen**)

Henrik Kniberg



Minor difference:

Scrum prescribes a prioritized product backlog



Scrum:

- **Product backlog must exist**
- **Changes to product backlog take effect next sprint (not current sprint)**
- **Product backlog must be sorted by business value**

Kanban:

- **Product backlog is optional**
- **Changes to product backlog take effect as soon as capacity becomes available**
- **Any prioritization scheme can be used. For example:**
 - Take any item
 - Always take the top item
 - Always take the oldest item
 - 20% on maintenance items, 80% on new features
 - Split capacity evenly between product A and product B
 - Always take red items first

.. but many teams combine these approaches

Minor difference:

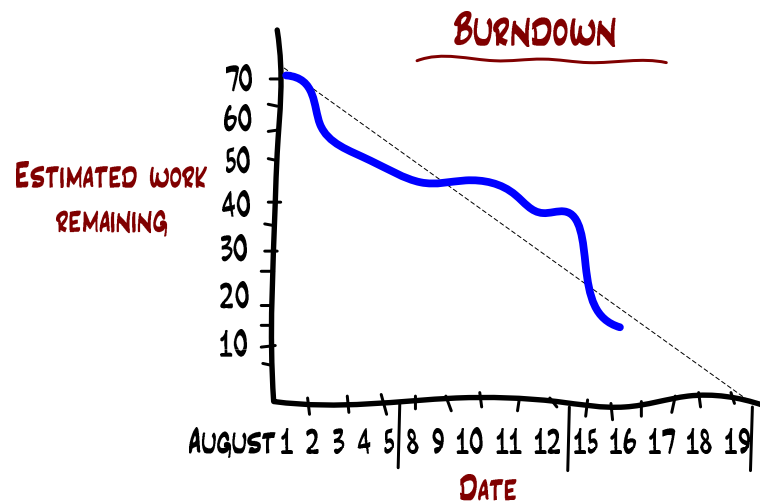
Scrum prescribes daily meetings



... but many Kanban teams do that anyway.

Minor difference:

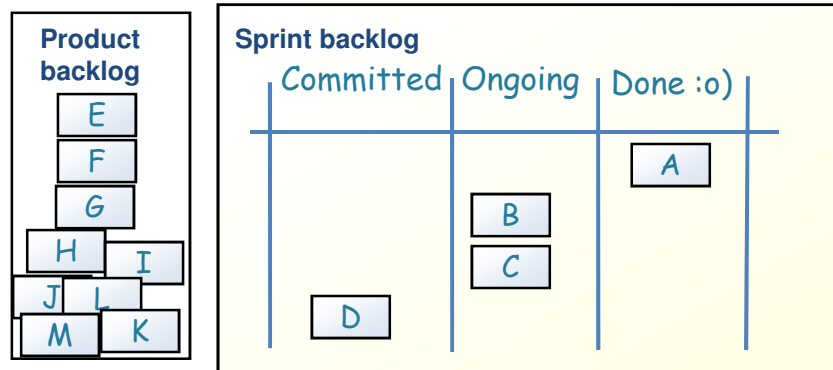
In Scrum, burndown charts are prescribed



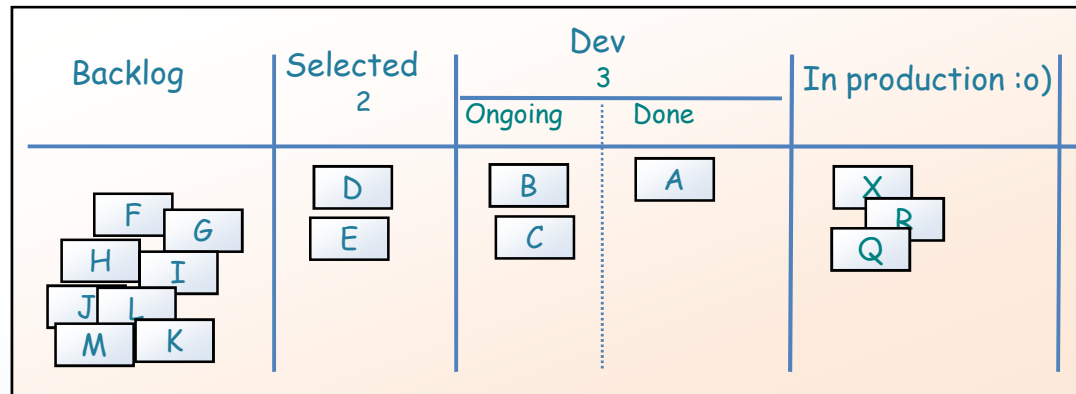
No specific types of diagrams prescribed in Kanban. Teams use whatever they need.

Example: Scrum board vs Kanban board

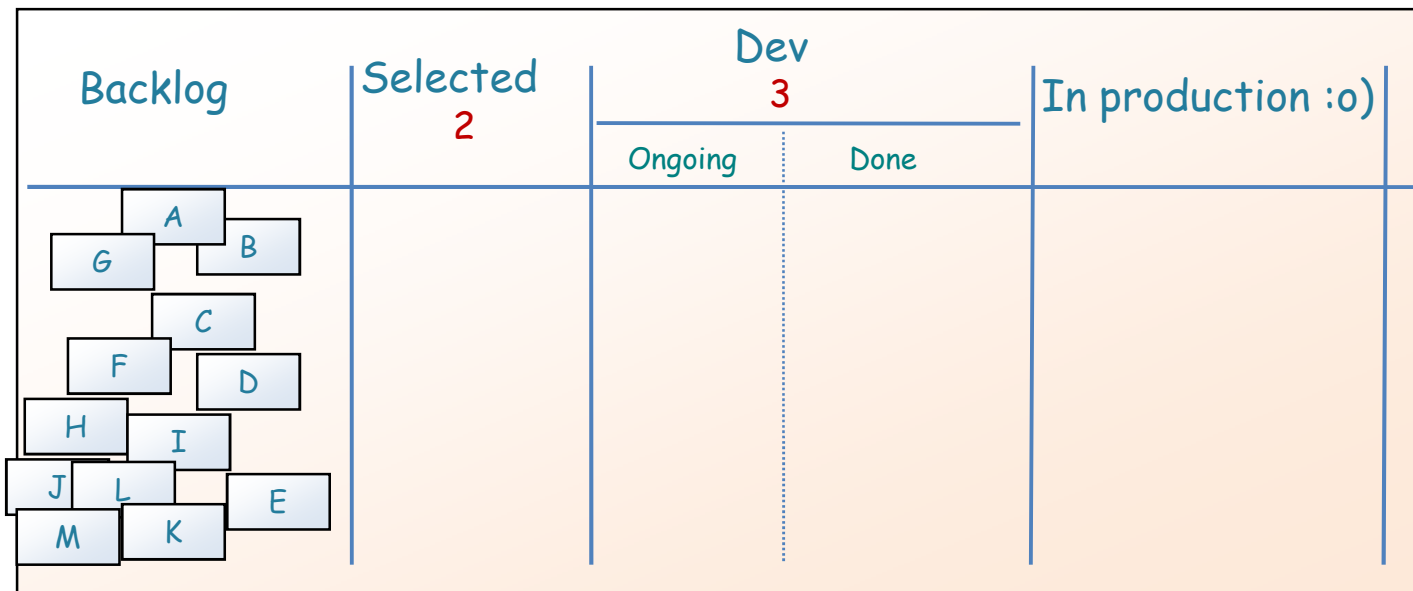
Scrum



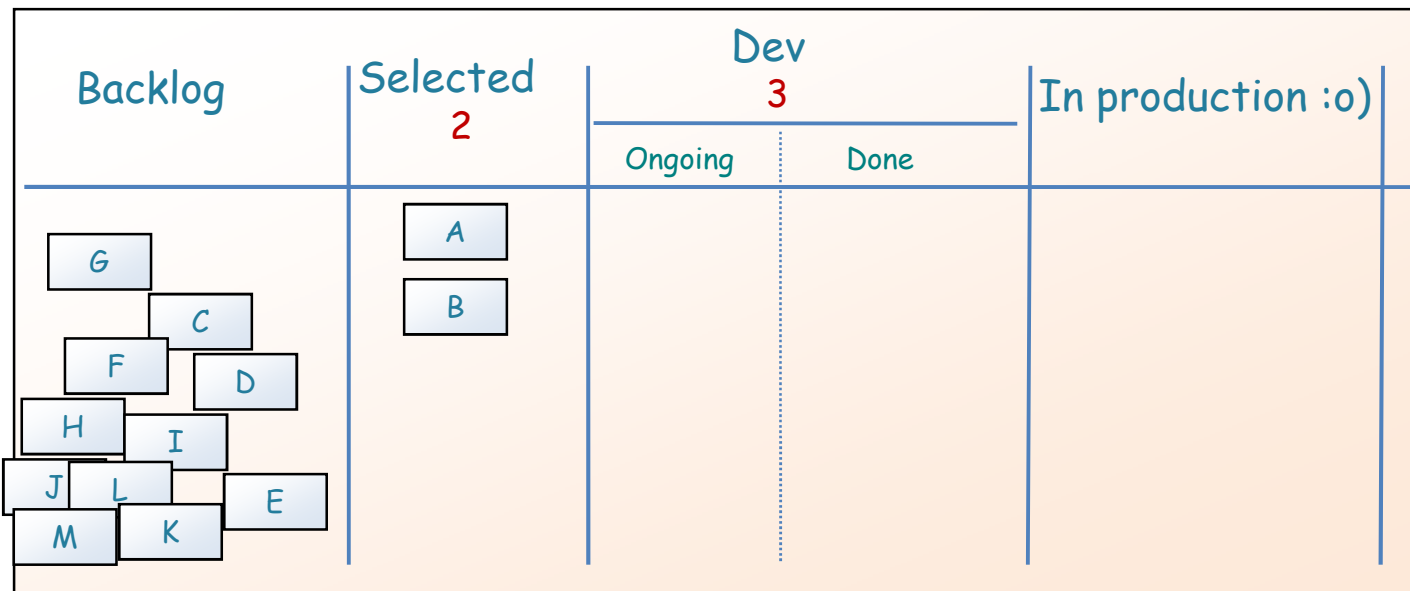
Kanban



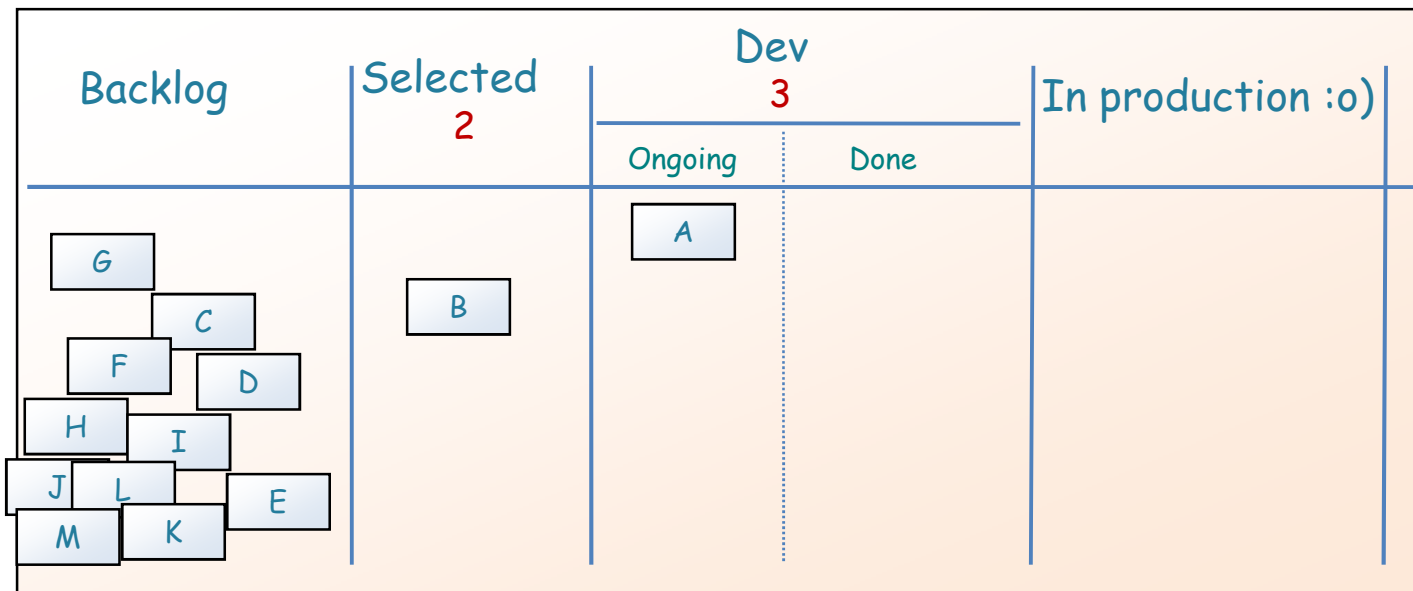
Scenario 1 – one piece flow



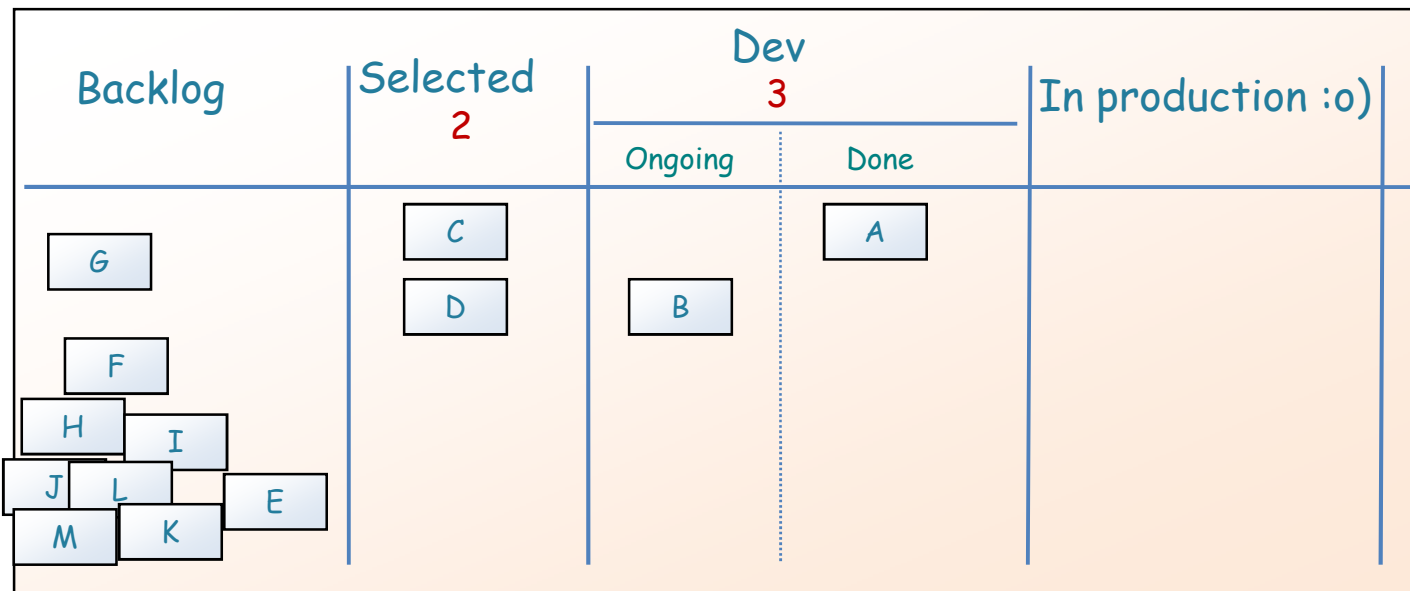
Scenario 1 – one piece flow



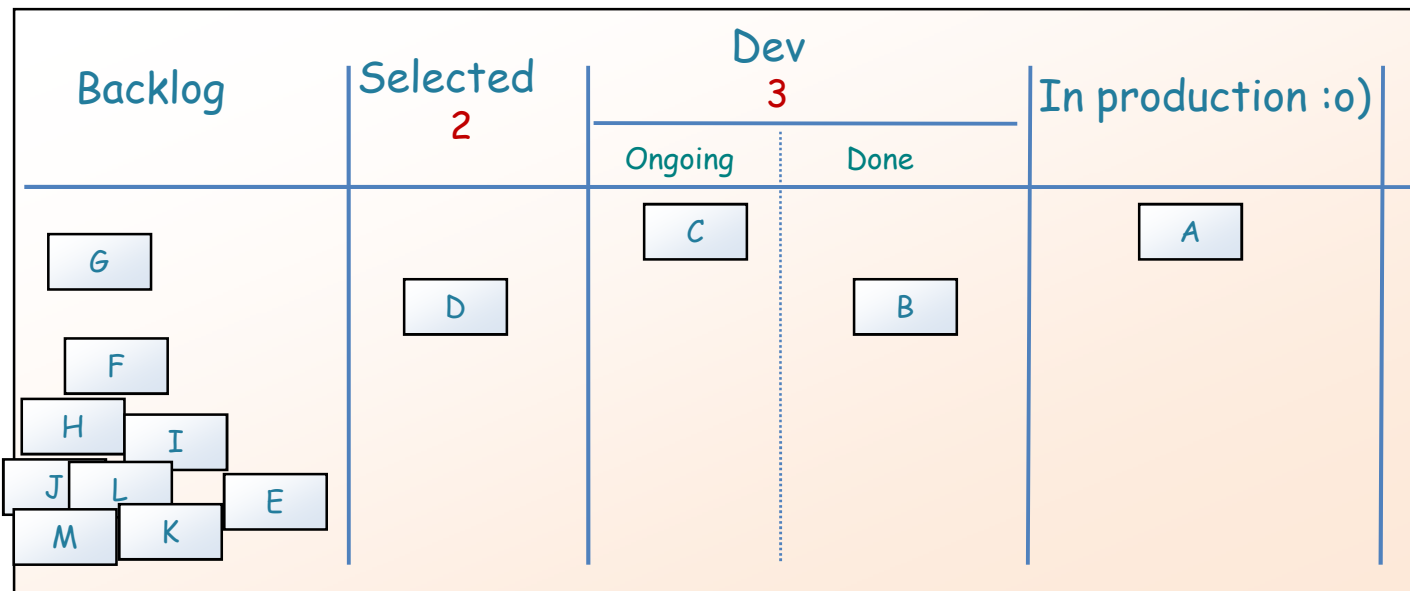
Scenario 1 – one piece flow



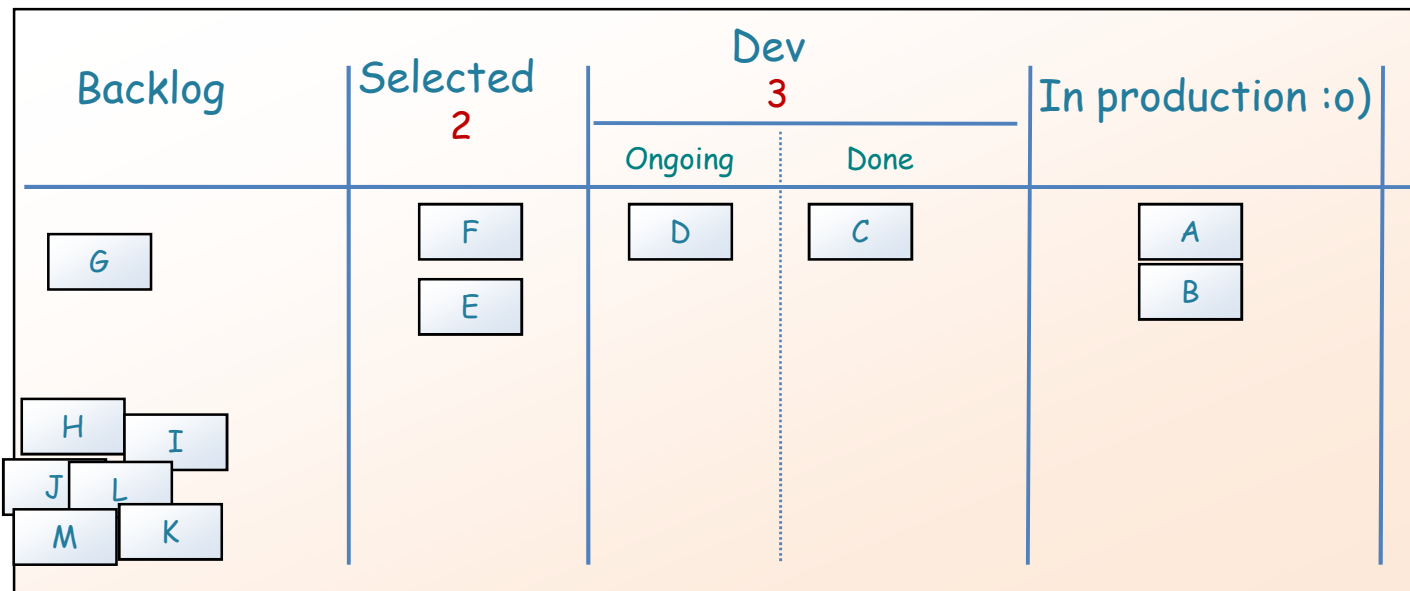
Scenario 1 – one piece flow



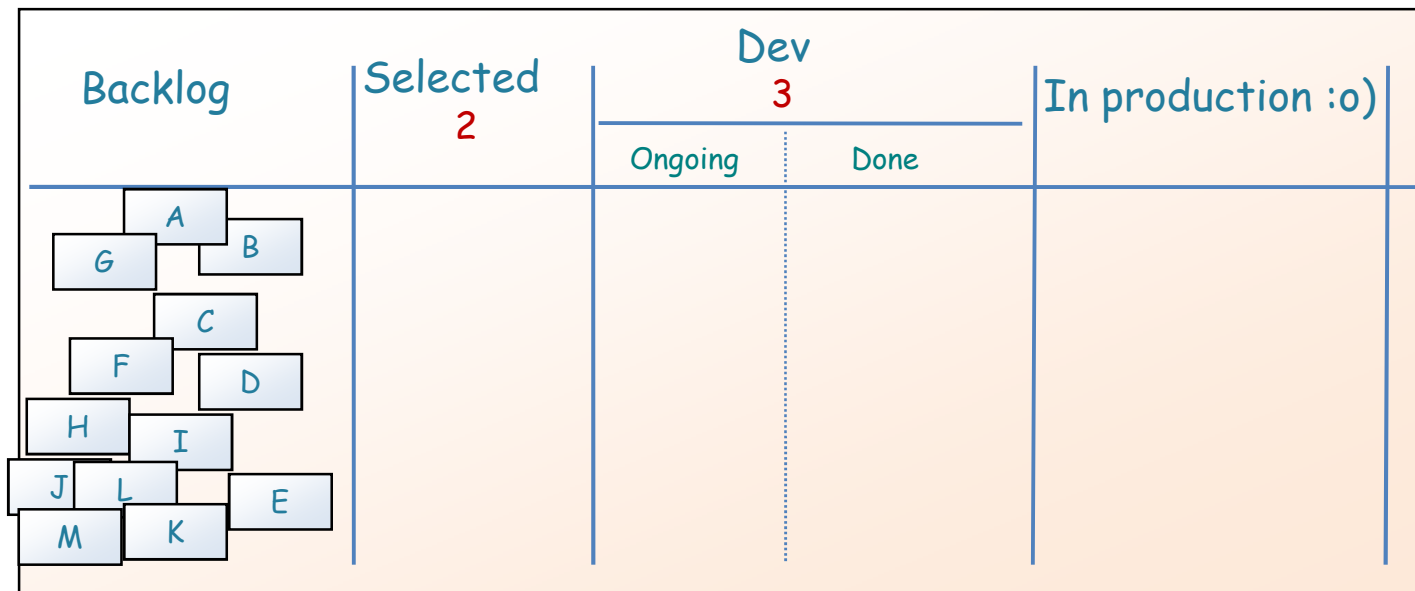
Scenario 1 – one piece flow



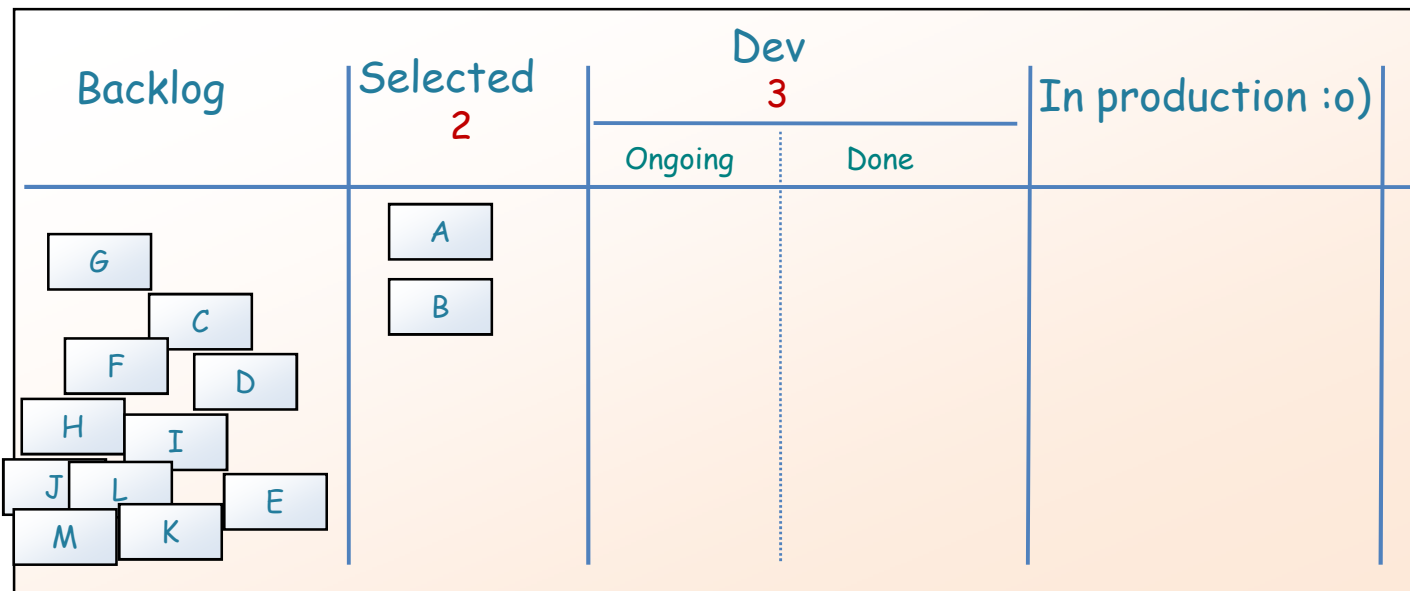
Scenario 1 – one piece flow



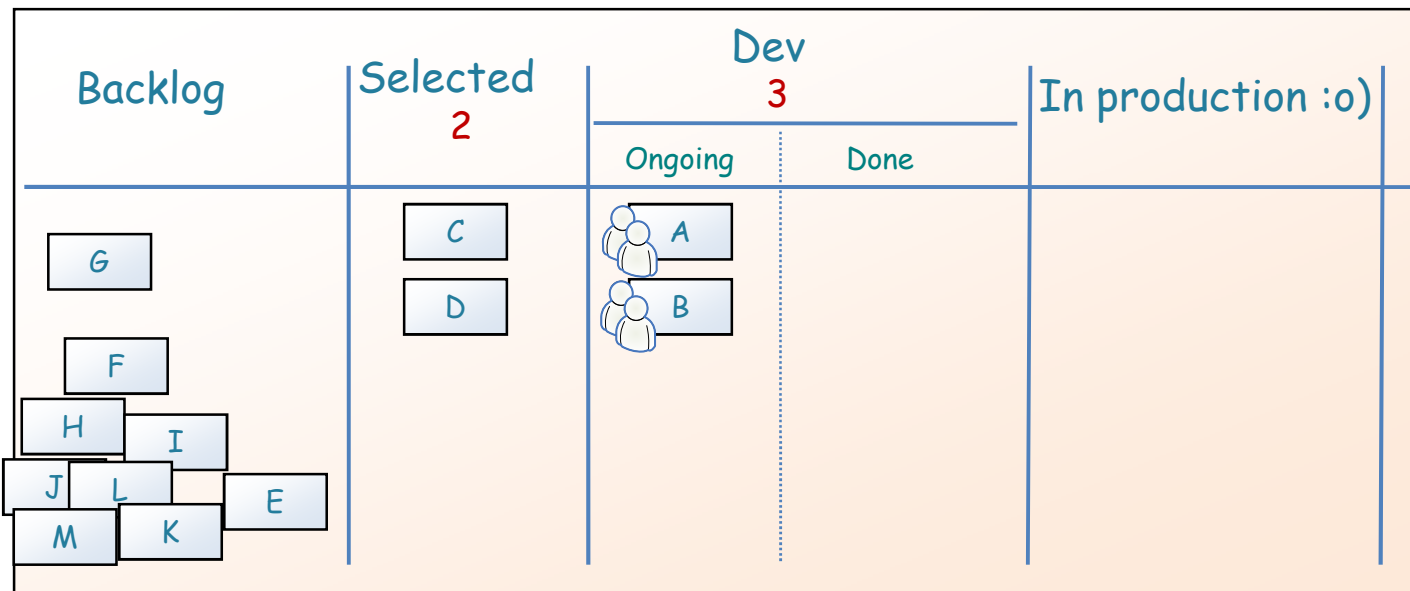
Scenario 2 – Deployment problem



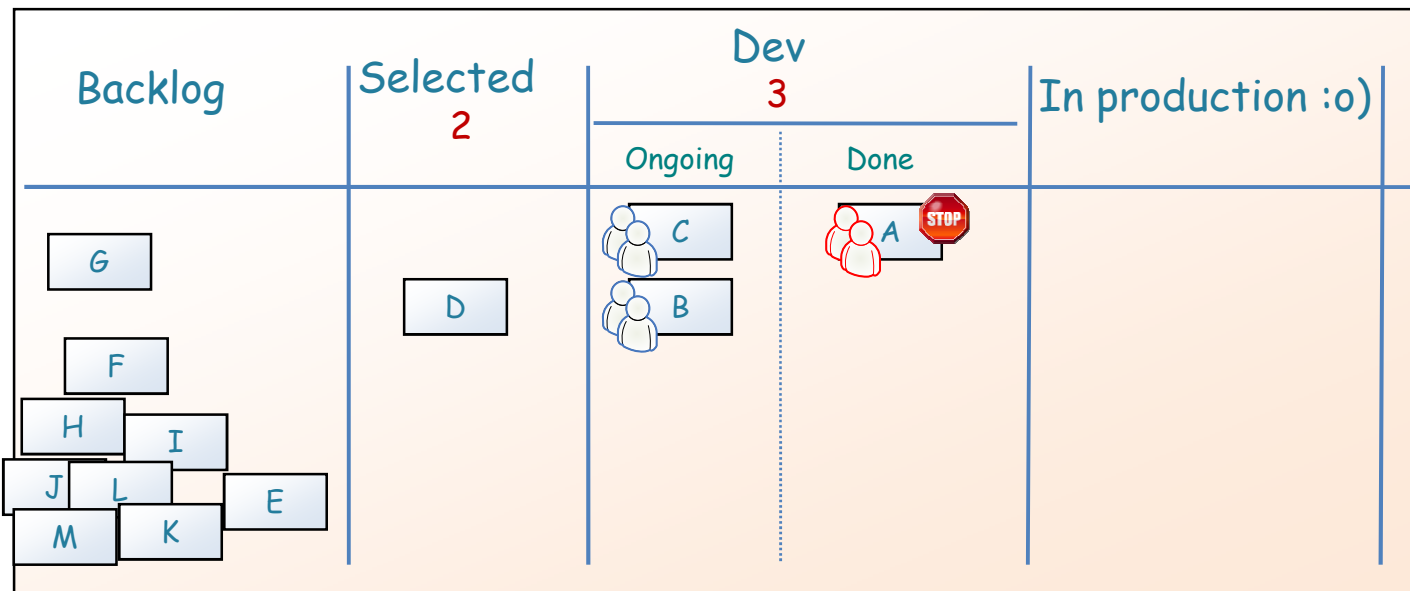
Scenario 2 – Deployment problem



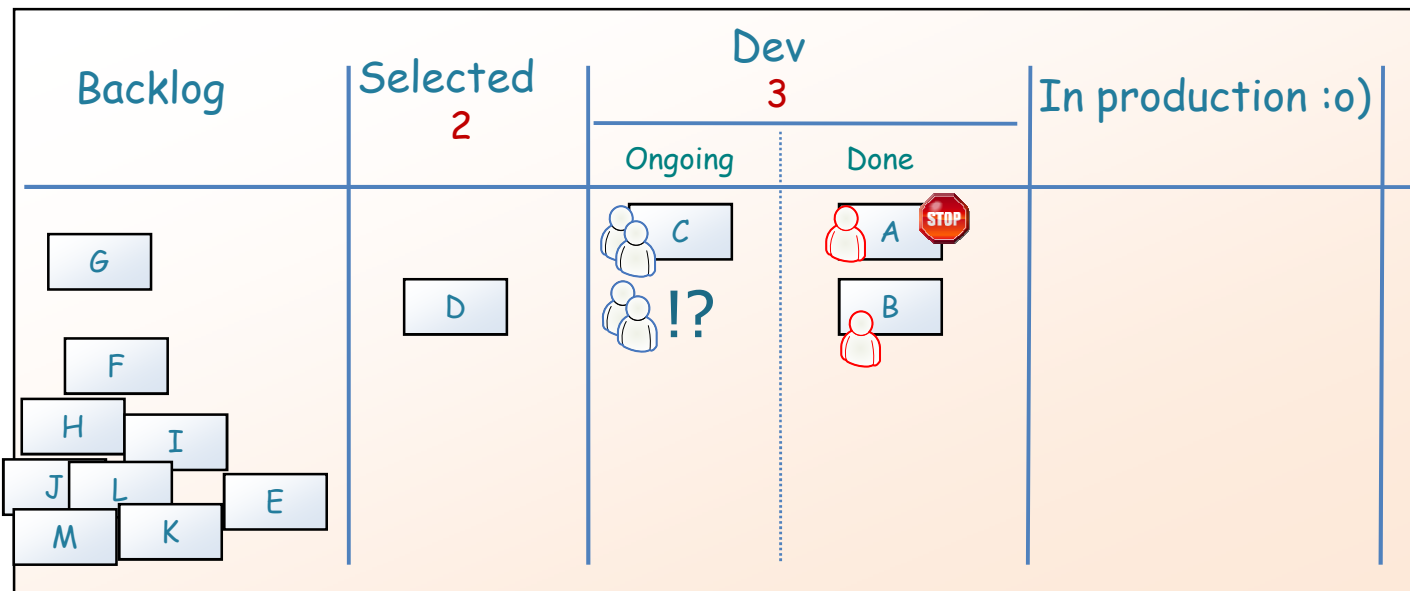
Scenario 2 – Deployment problem



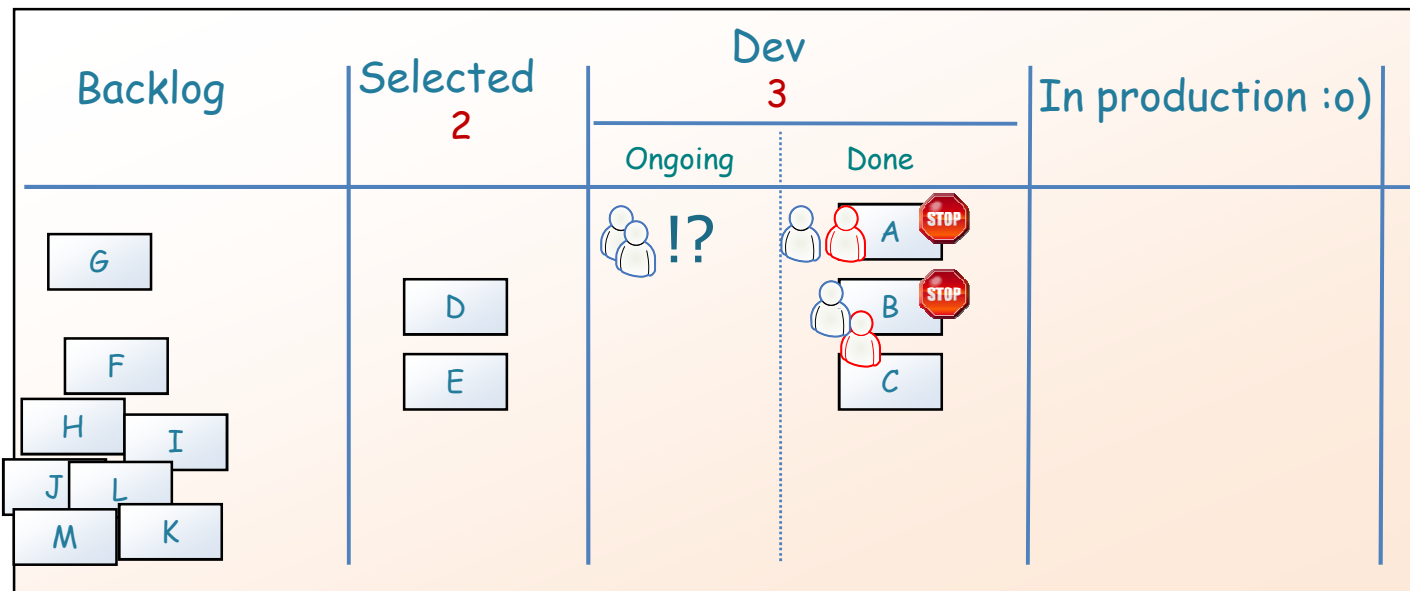
Scenario 2 – Deployment problem



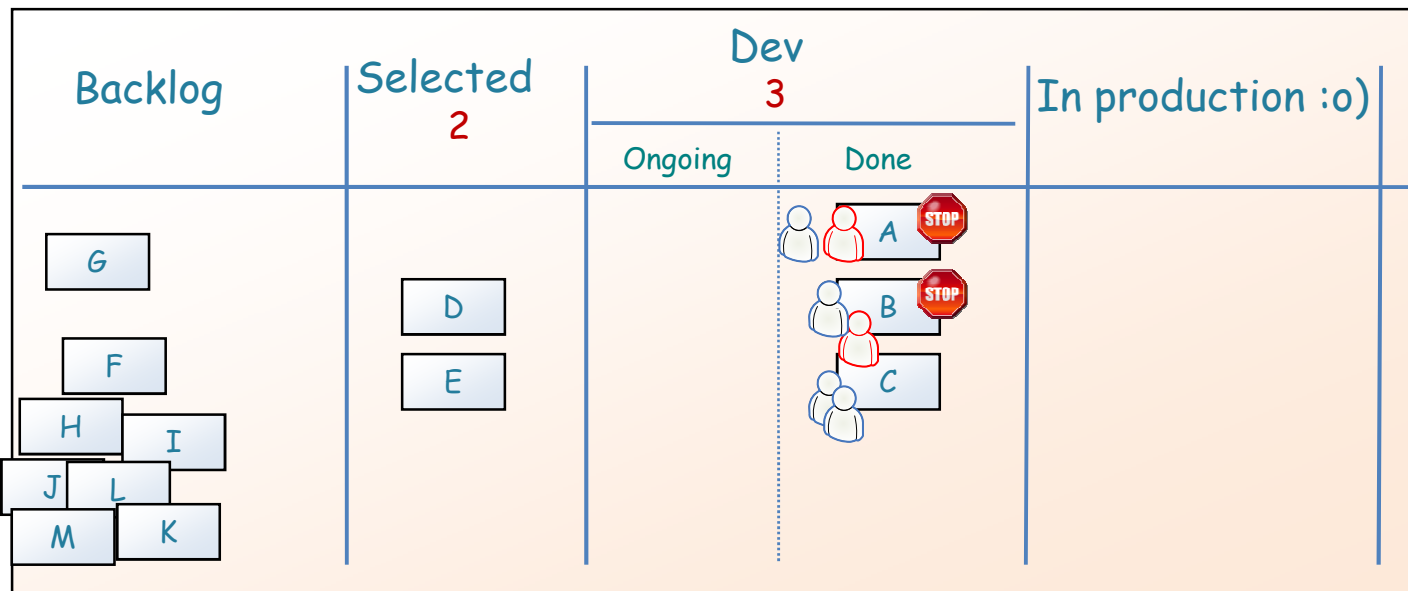
Scenario 2 – Deployment problem



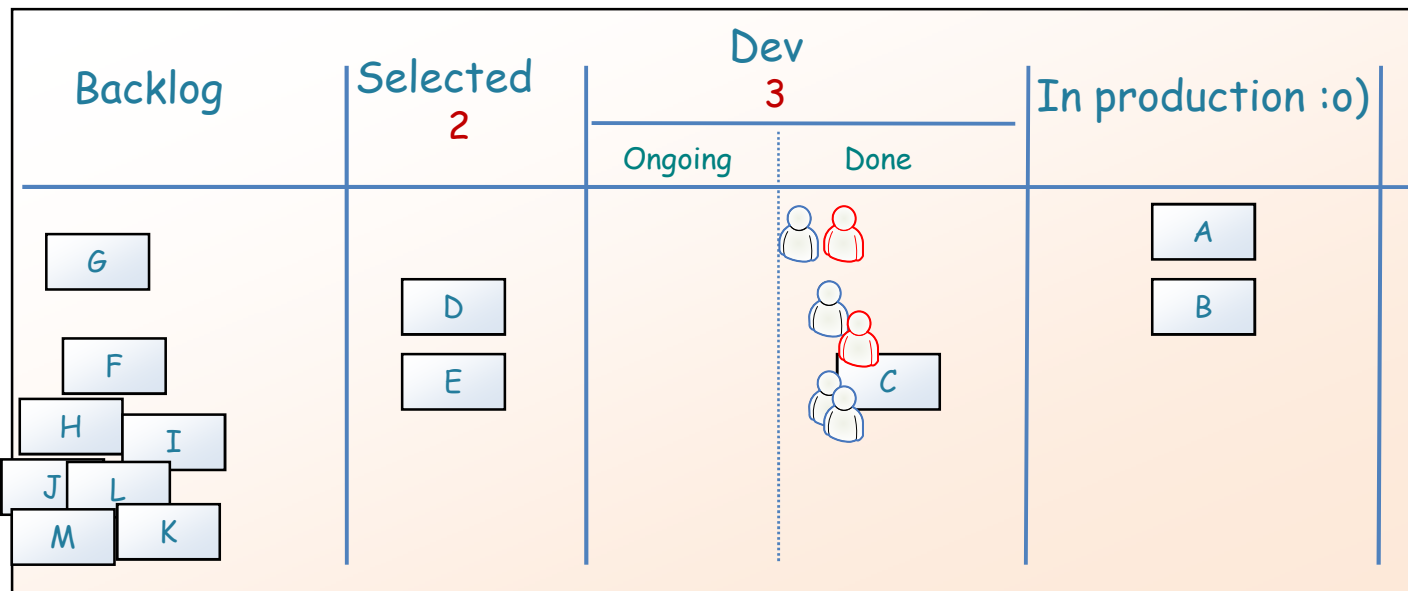
Scenario 2 – Deployment problem



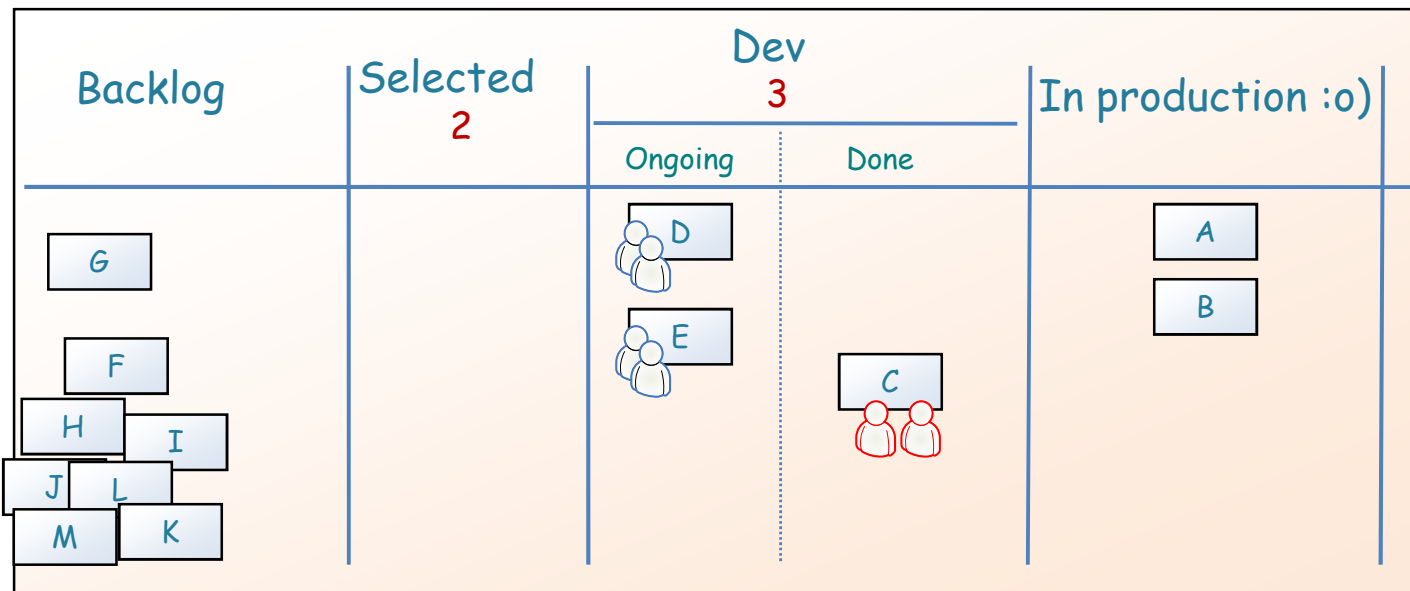
Scenario 2 – Deployment problem



Scenario 2 – Deployment problem



Scenario 2 – Deployment problem



Kanban vs Scrum

Summary

www.crisp.se/henrik.kniberg/kanban-vs-scrum.pdf

Similarities

- Both are Lean and Agile
- Both based on pull scheduling
- Both limit WIP
- Both use transparency to drive process improvement
- Both focus on delivering releasable software early and often
- Both are based on self-organizing teams
- Both require breaking the work into pieces
- In both cases the release plan is continuously optimized based on empirical data (velocity / lead time)

Differences

Scrum	Kanban
Timeboxed iterations prescribed.	Timeboxed iterations optional.
Team commits to a specific amount of work for this iteration.	Commitment optional.
Uses Velocity as default metric for planning and process improvement.	Uses Lead time as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. Specialist teams allowed.
Items broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed	No particular type of diagram is prescribed
WIP limited indirectly (per sprint)	WIP limited directly (per workflow state)
Estimation prescribed	Estimation optional
Cannot add items to ongoing iteration.	Can add new items whenever capacity is available
A sprint backlog is owned by one specific team	A kanban board may be shared by multiple teams or individuals
Prescribes 3 roles (PO/SM/Team)	Doesn't prescribe any roles
A Scrum board is reset between each sprint	A kanban board is persistent
Prescribes a prioritized product backlog	Prioritization is optional.

Henrik Kniberg

Most importantly: Start with retrospectives!



- Evolve the right process for your context.
- Don't worry about getting it right from the start.
- Expand your toolkit.
- Experiment!

Henrik Kniberg

