travelping

Shipping AAA Products using Erlang & Kanban

25/03/2011

Erlang Factory SF Bay Area 2011

What we are doing…

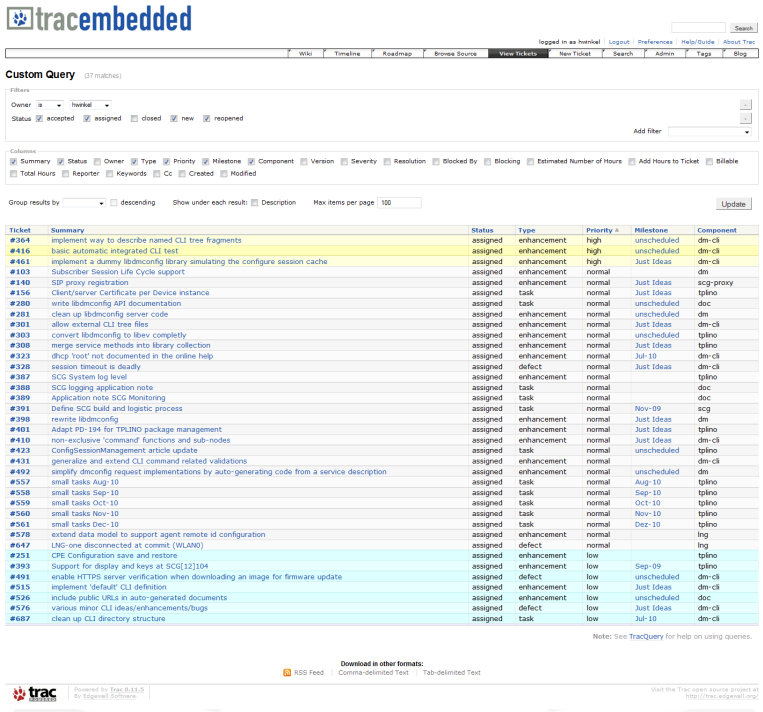
- AAA-Systems (RADIUS, DIAMETER gets traction)
- Session Control Systems (BNG, AMF, BRAS)
- TPLINO Platform as enabler
- Managed Service for AAA and Session Contr
- Whitelabel ISP Services

- High Quality Requirements !
- “AAA on Rails” another App-Server Vision …

Why we Changed SW Development

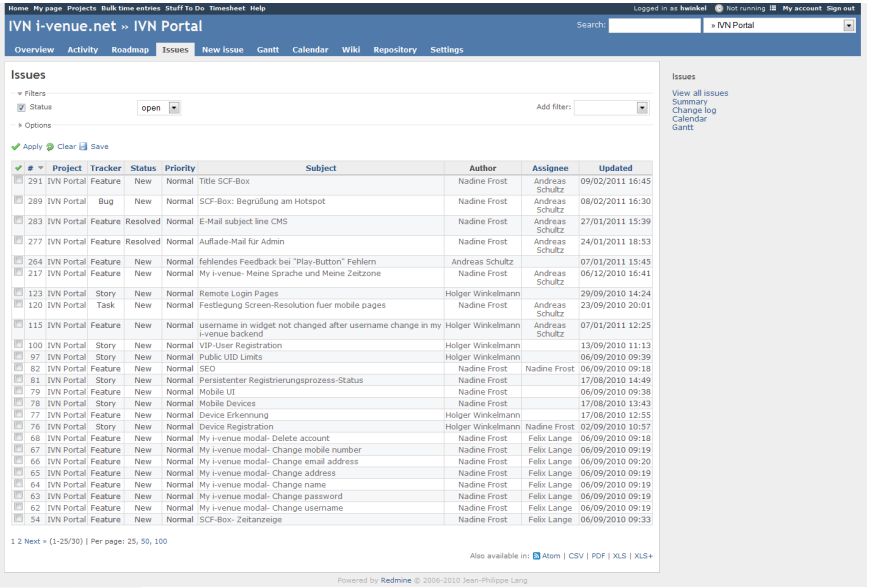
- Time to Market
- Stability issues in the past
- Fragmented technologies (PHP, C, Java, Lua …)
- Problems with SCRUM-Sprints and -Estimations
- Does SCRUM fit well to product development ???
- Failed Sprints leads to frustration ☹
- Backlog, Tickets, Issues in different tools
- We still have operational tasks „disturbing“

Existing Environment



- Redmine + GIT
  - We try to adapt a Process

- Trac + SVN
  - introduced by Dev-Team



What makes a SW Product ?

travelping

- Named, Shippable Unit
- Developer- vs. Operations-View
- Versioned, Packaged, Visible
- Installable & Un-Installable!
- Many NON-Functional Requirements

- Boring - but needed to succeed or even survive

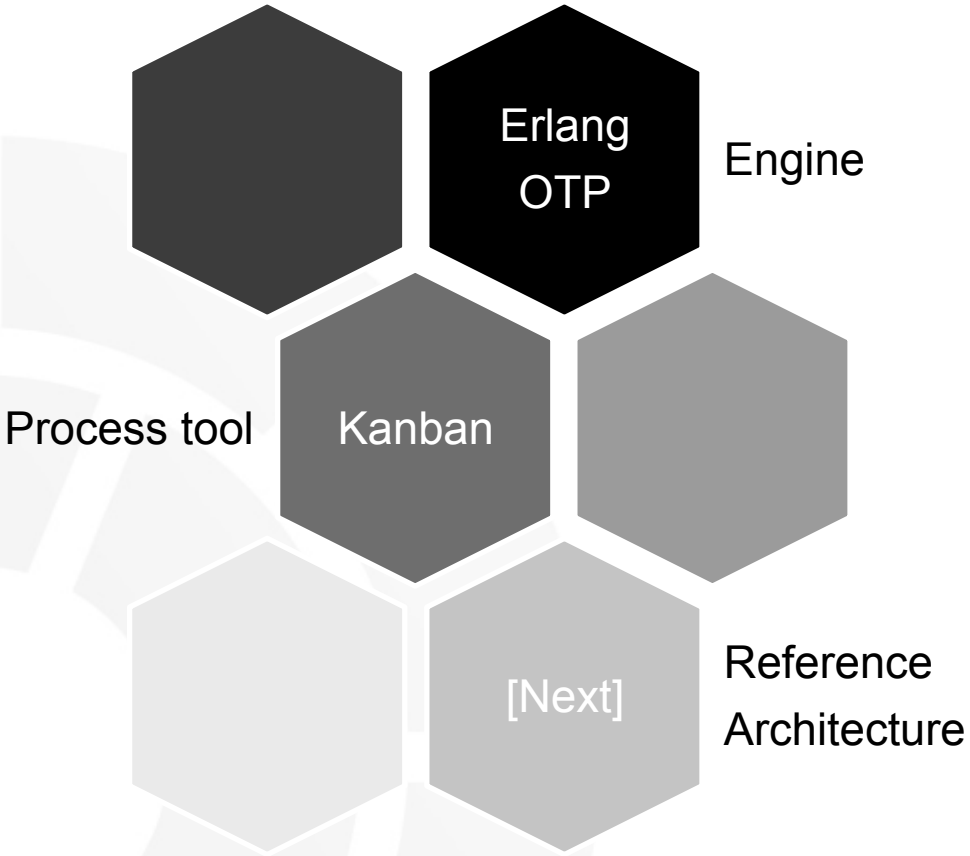
The war of A vs. B vs. C vs. …

travelping

- Read books, Blogs & watch conference talks
- SCRUM vs. Kanban vs. XP vs. RUP (waterfall)
- Scala vs. Erlang vs. Go vs. Node.js
- Linux vs. FreeBSD vs. NetBSD (for embedded)
- Eclipse vs. Emacs vs ….
- Redmine vs. Trac vs. Bugzilla
- SVN vs. GIT vs. Mercurial
- ….

Introduction of Erlang & Kanban

travelping



Erlang at Travelping

travelping

- Started already two Years ago
- To many segfaults in C based TPLINO (SCG,LNG)
- Devices with 8MB Flash / 32MB RAM today
- Gives us room for a bigger VM then Lua actually
- Erlang has it own track record
- Rethink Java for the Server
- Proof of Concept of RADIUS Server in Erlang
- One Technology for all Products
- Small but Great Erlang Community

Erlang at Travelping

travelping

- Visited two Erlang-Factories and -Universities
- Erlang gives us:
  - A proven Track Record (20 Years)
  - A platform build for our purposes (B2T)
  - A defined programming model (OTP)
  - just 3 books to read ;-)
- We build in about three weeks
  - The fastest RADIUS server we know so far
  - Without a single crash
  - Delivers a 100Mbit/s stream of RADIUS packets

So - Erlang fits well for us !..

travelping

Dot.Coms & Media

CouchDB relax

riak

Banking

RabbitMQ Open Source Enterprise Messaging

Defense

Transport

Telecoms

tplino

tposs


Decider


Expert


Developer

ERICSSON

ERLANG

Supported Packages

Erlang/OTP

Erlang Solutions Ltd

Linux

tplino

Operating Systems

FreeBSD

redhat

ubuntu

ARM

MIPS TECHNOLOGIES

PowerPC

Architectures Processors

intel

CAVIUM NETWORKS

TILERA


Developer


CPEs


Industrial


Mobile


Server


u/ATCA

Kanban at Travelping

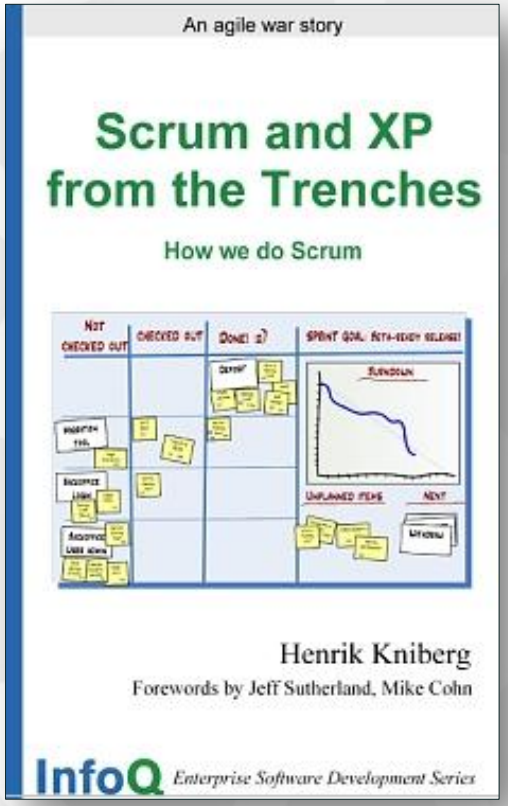
travelping

- Kanban in a Nutshell
- Why SCRUM failed (for us)
- The Development-Process
- Value Stream Mapping
- Build a Kanban-Board
- Card Anatomy
- WIP Limits und other policies

Kanban vs. Scrum

travelping

- Many Thanks to Henrik Kniberg of crisp.se !

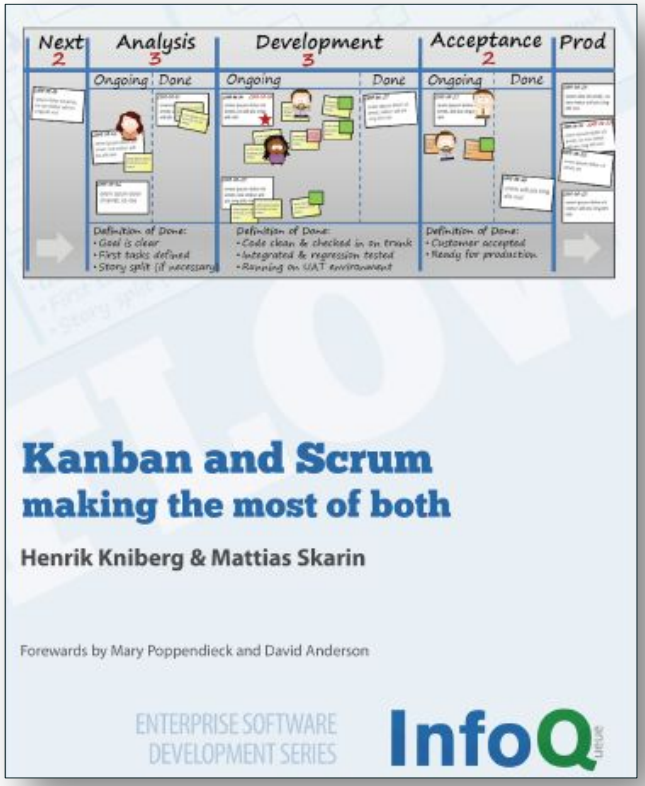


Henrik Kniberg

CERTIFIED Scrum Trainer

Agile Alliance® Board of directors

henrik.kniberg@crisp.se
+46 70 4925284



Which team needs most improvement?

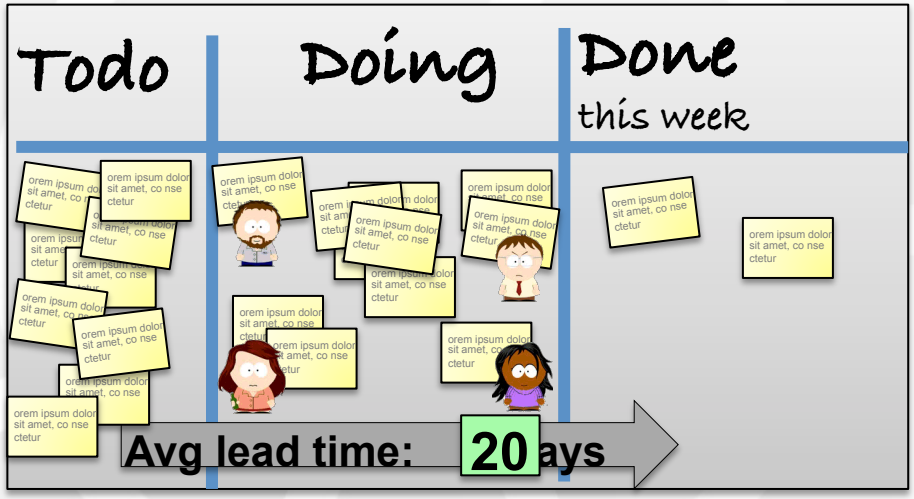
travelping

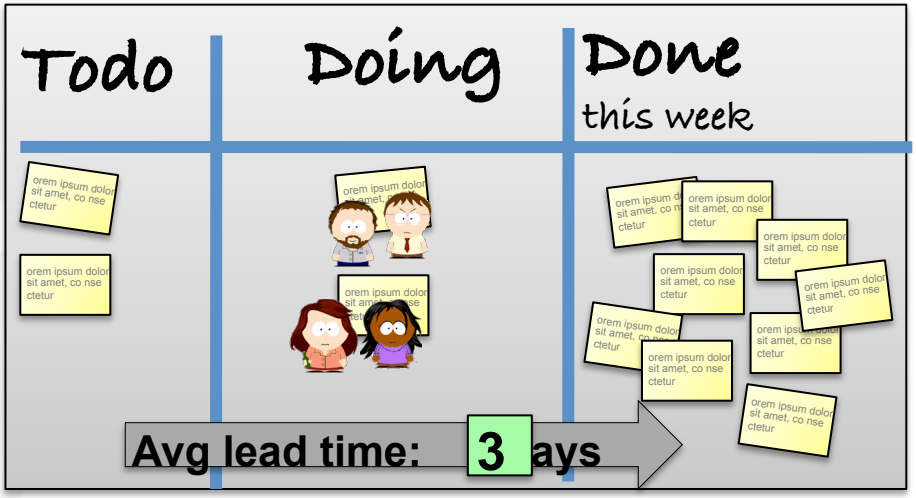
Managers who don't know how to
measure what they want
settle for
wanting what they can measure

Russel Ackoff

Team 1

Team 2

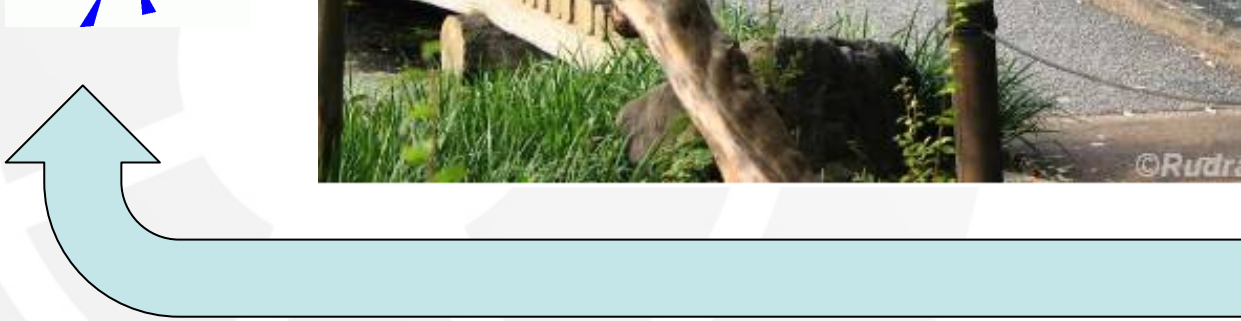




travelping

Kanban at Imperial Palace Gardens

WiP = 8



Kanban

travelping

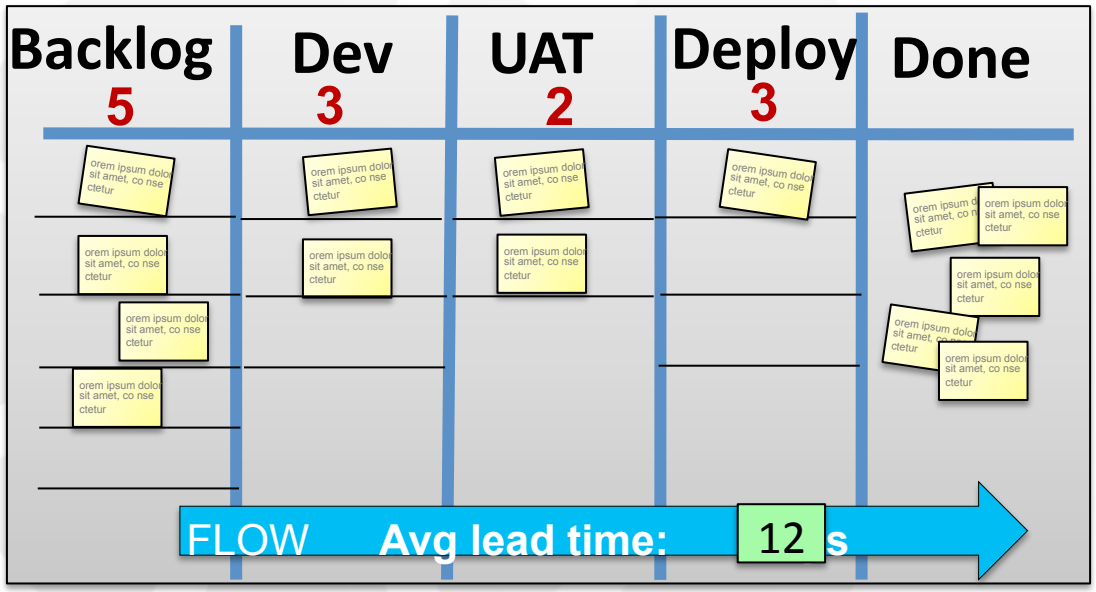
- ”Signaling” System
- Visualisation

看板 ”Visual Card”

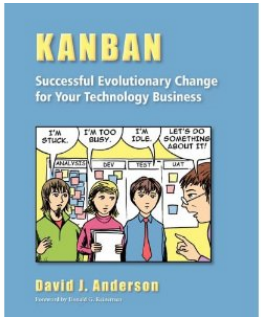
Kanban in SW development

travelping

- Visualize the workflow
- Limit WIP (work in progress)
- Measure & optimize flow
- Explicit policies (definition of Done, WIP limits, etc)



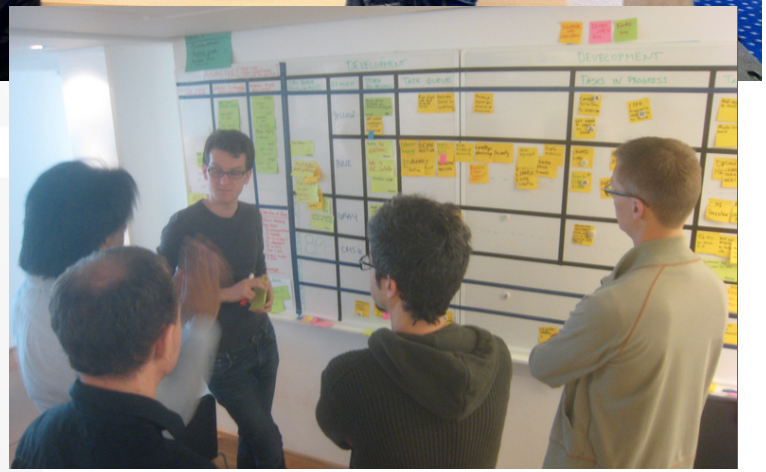
Pioneered by
David Anderson
in 2004



Kanban example

travelping

Board shared by the teams



travelping

Compare for understanding not for judgement

Tool

”anything used as a means of accomplishing a task or purpose.”
- dictionary.com

Thinking tools

a.k.a. ”mindsets” or ”philosophies”

Lean Agile Systems Thinking Theory of Constraints

travelping

Toolkits

a.k.a. ”frameworks”

RUP XP Scrum Kanban

Physical tools

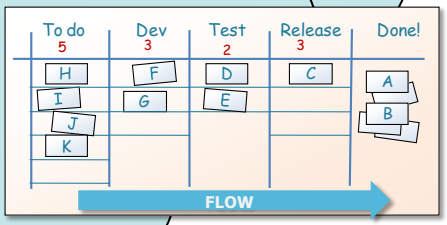
Process tools

a.k.a. ”organizational patterns”

Product Owner role

Pair programming

Visualize the workflow



travelping

Compare RUP, XP, Scrum, Kanban ...

More prescriptive

More adaptive

RUP (120+)

- Architecture Reviewer
- Business Designer
- Business-Model Reviewer
- Business-Process Analyst
- Capsule Designer
- Change Control Manager
- Code Reviewer
- Configuration Manager
- Course Developer
- Database Designer
- Deployment Manager
- Design Reviewer
- Designer
- Graphic Artist
- Implementer
- Integrator
- Process Engineer
- Project Manager
- Project Reviewer
- Requirements Reviewer
- Requirements Specifier
- Software Architect
- Stakeholder
- System Administrator
- System Analyst
- Technical Writer
- Test Analyst
- Test Designer
- Test Manager
- Tester
- Tool Specialist
- User-Interface Designer
- Architectural analysis
- Assess Viability of architectural proof-of-concept
- Capsule design
- Class design
- Construct architectural proof-of-concept
- Database design
- Describe distribution
- Describe the run-time architecture
- Design test packages and classes
- Develop design guidelines
- Develop programming guidelines
- Identify design elements
- Identify design mechanisms
- Incorporate design elements
- Prioritize use cases
- Review the architecture
- Review the design
- Structure the implementation model
- Subsystem design
- Use-case analysis
- Use-case design
- Analysis model
- Architectural proof-of-concept
- Bill of materials
- Business architecture document
- Business case
- Business glossary
- Business modeling guidelines
- Business object model
- Business rules
- Business use case
- Business use case realization
- Business use-case model
- Business vision
- Change request
- Configuration audit findings
- Configuration management plan
- Data model
- Deployment model
- Deployment plan
- Design guidelines
- Design model
- Development case
- Development-organization assessment
- End-user support mateirla
- Glossary
- Implementation model
- Installation artifacts
- Integration build plan
- Issues list
- Iteration assessment
- Iteration plan
- Manual styleguide
- Programming guidelines
- Quality assurance plan
- Reference architecture
- Release notes
- Requirements attributes
- Requirements management plan
- Review record
- Risk list
- Risk management plan
- Software architecture document
- Software development plan
- Software requirements specification
- Stakeholder requests
- Status assessment
- Supplementary business specification
- Supplementary specification
- Target organization assessment
- Test automation architecture
- Test cases
- Test environment configuration
- Test evaluation summary
- Test guidelines
- Test ideas list
- Test interface specification
- Test plan
- Test suite
- Tool guidelines
- Training materials
- Use case model
- Use case package
- Use-case modeling guidelines
- Use-case realization
- Use-case storyboard
- User-interface guidelines
- User-interface prototype
- Vision
- Work order
- Workload analysis model

XP (13)

- Whole team
- Coding standard
- TDD
- Collective ownership
- Customer tests
- Pair programming
- Refactoring
- Planning game
- Continuous integration
- Simple design
- Sustainable pace
- Metaphor
- Small releases

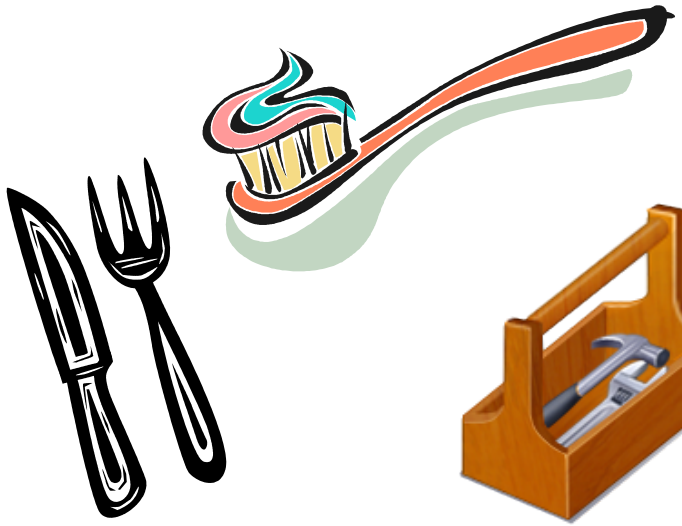
Scrum (9)

- Scrum Master
- Product Owner
- Team
- Sprint planning meeting
- Daily Scrum
- Sprint review
- Product backlogt
- Sprint backlog
- BUrndown chart

Kanban (3)

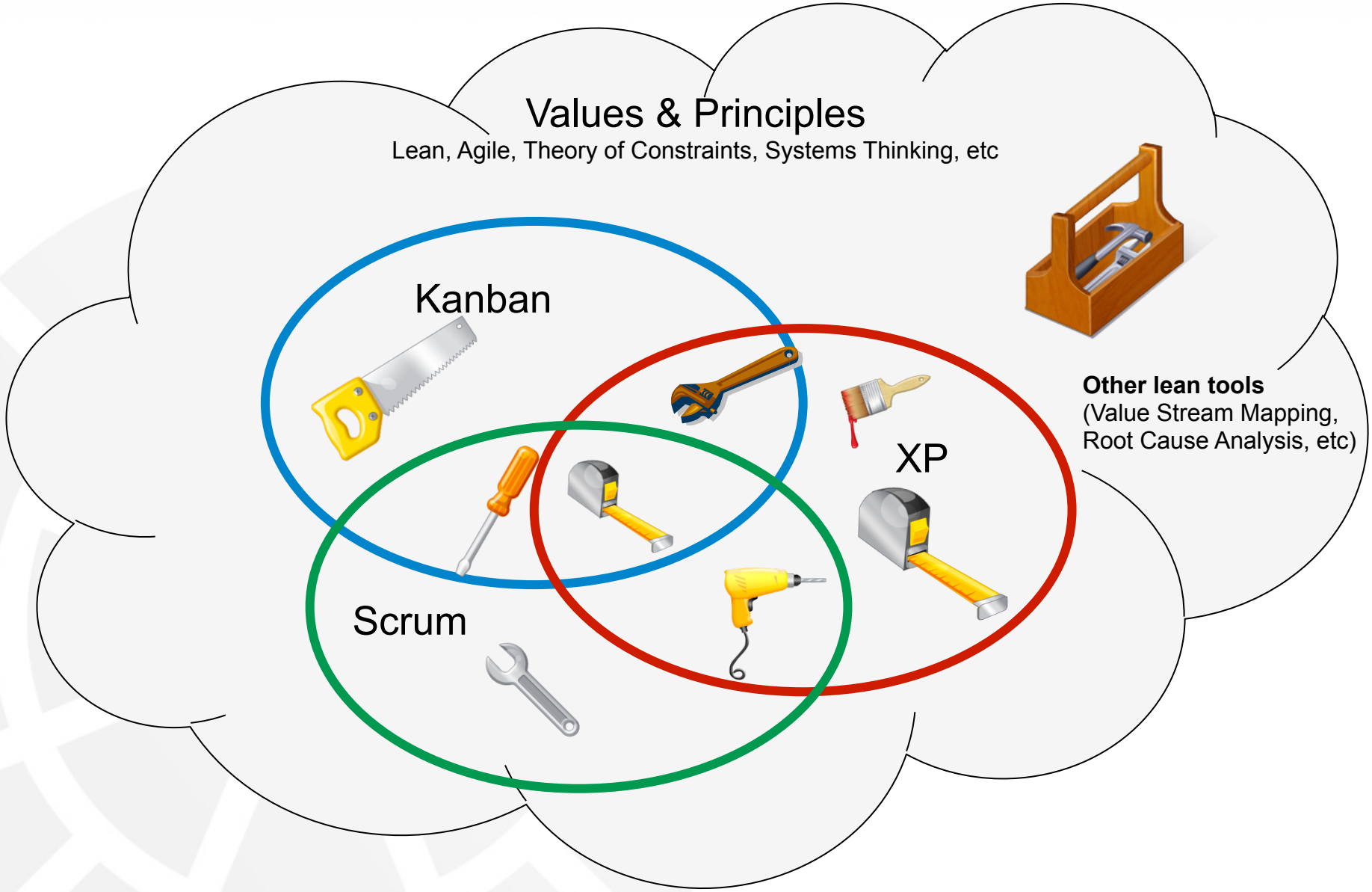
- Visualize the workflow
- Limit WIP
- Measure and optimize lead time

Do Whatever (0)



Lean & Agile process toolkits

travelping



Any tool can be misused...

travelping

Don’t blame the tool!...

travelping

Why limit WIP?

(work in progress)

Multitasking Game

travelping

Setup

4 – 6 customers

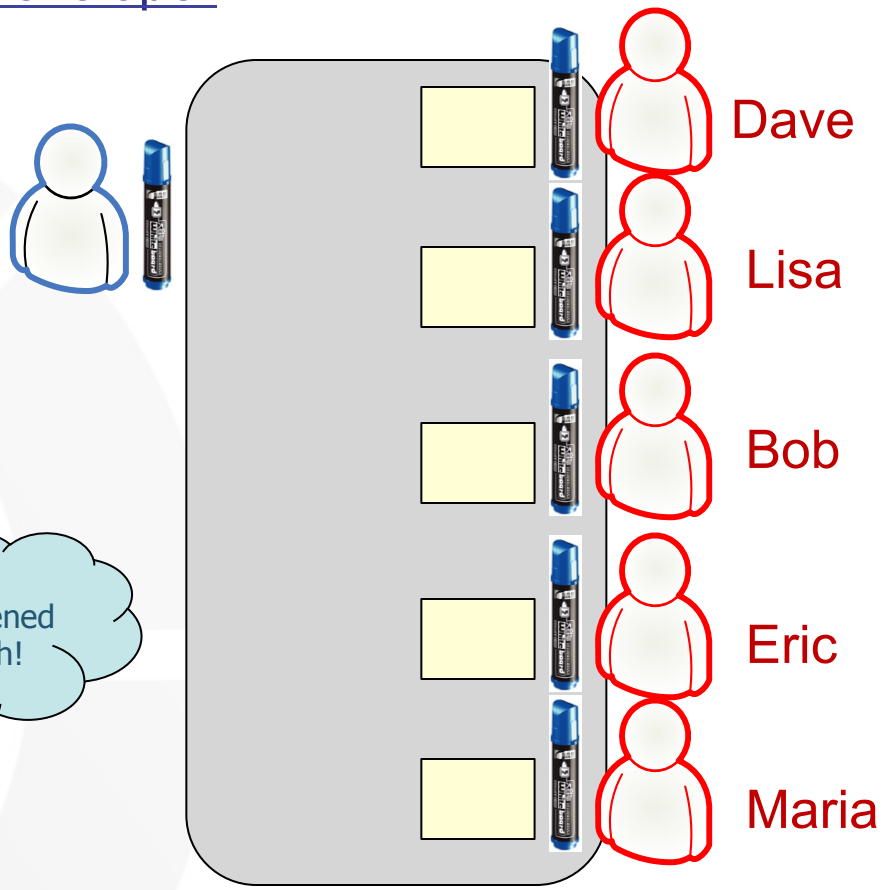
1 Developer

10 minutes movie

Henrik Kniberg
Kanban and Scrum Talk
at Oredev 11.11. 2010
(Start at Minute 13:00)

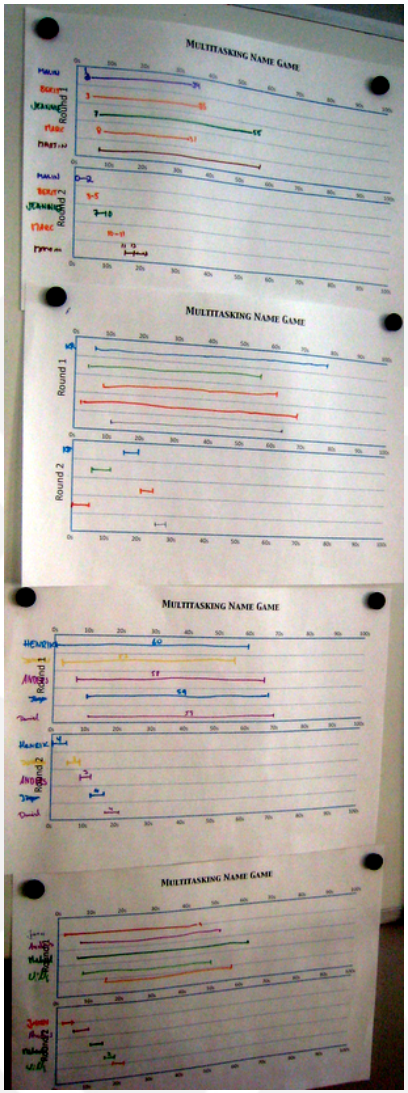
http://vimeo.com/16918747

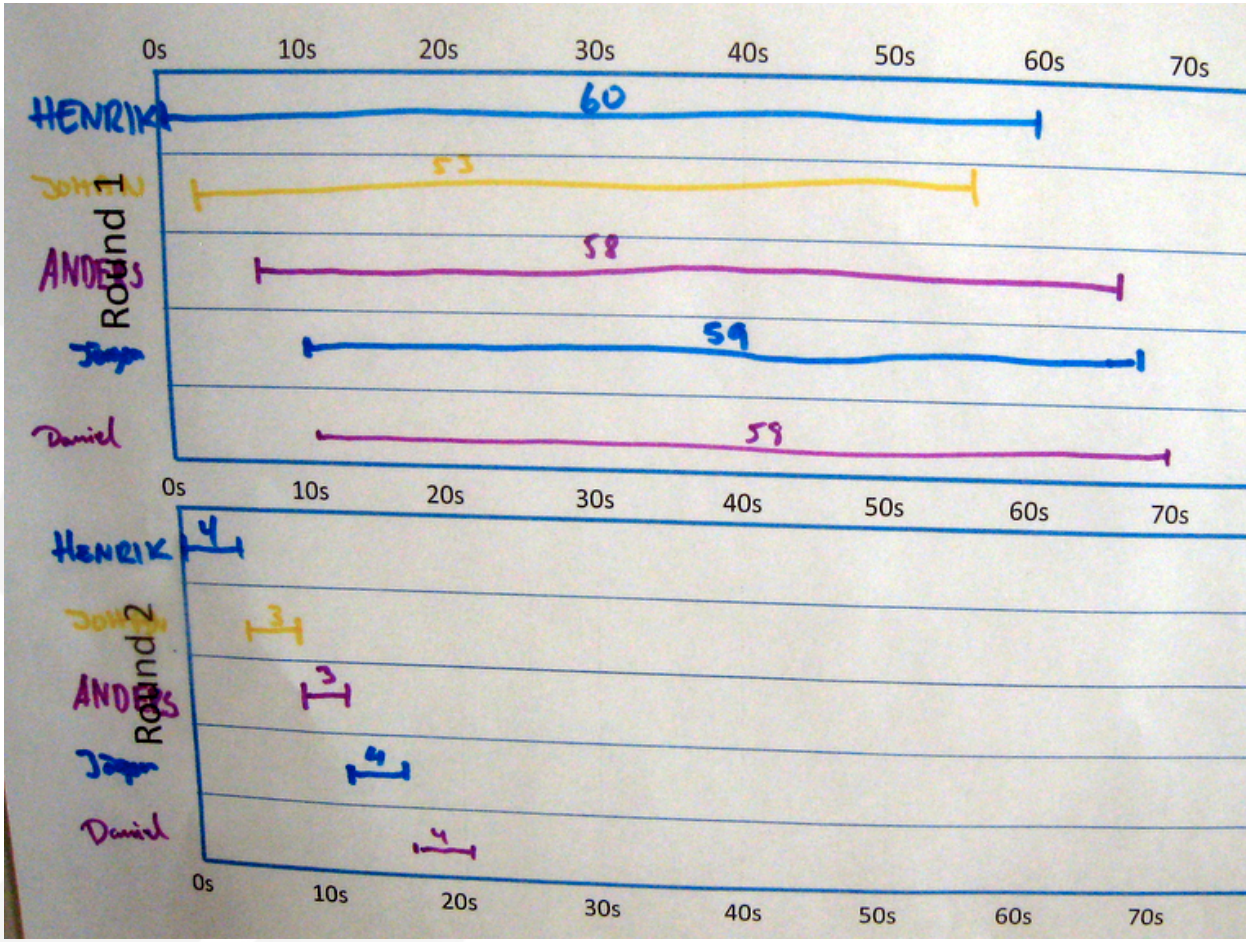
Highly recommened Video to watch!



Typical results of the Game

travelping

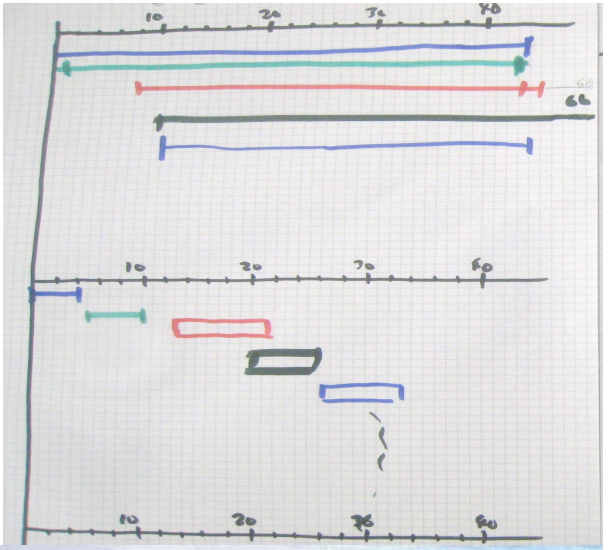


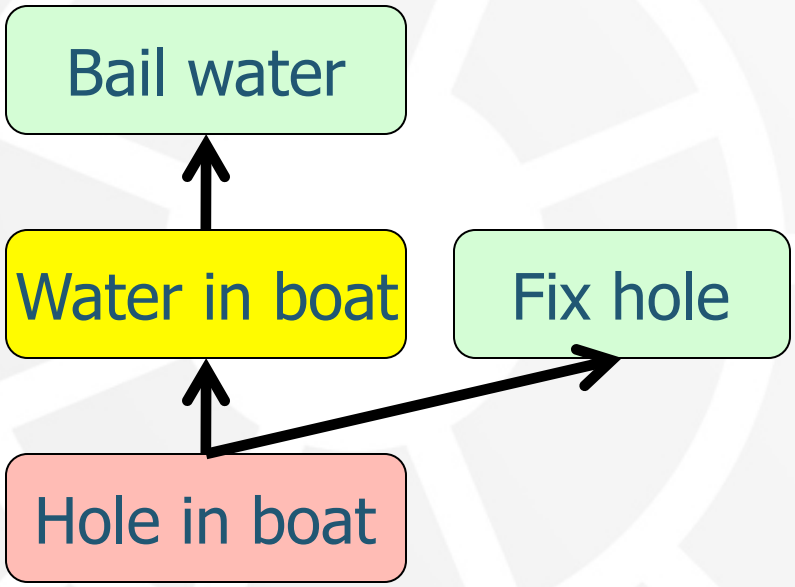


Causes of unnecessary multitasking

travelping

- Focusing on starting stuff rather than finishing
- Not limiting WIP
- Focusing on keep people busy (fear of slack)
- Accepting the “reason” & solving the symptom instead of the problem



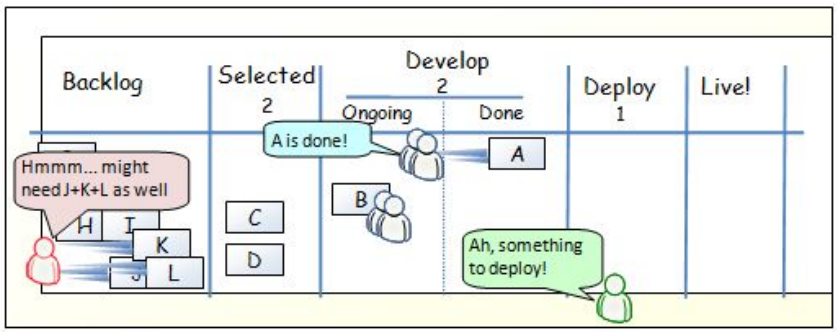


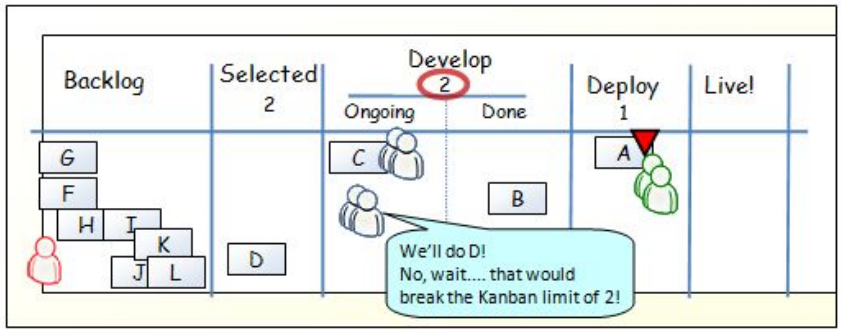
travelping

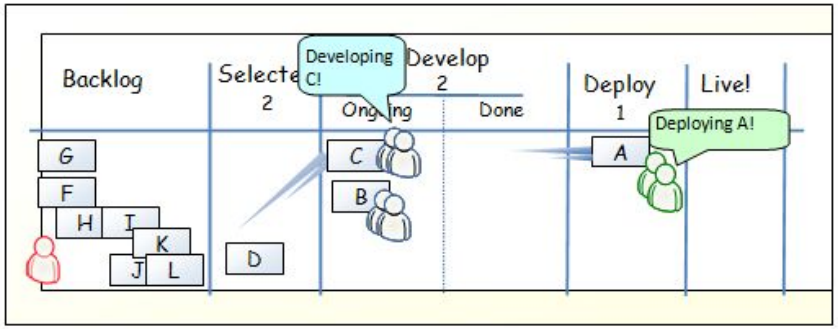
One day in Kanban land

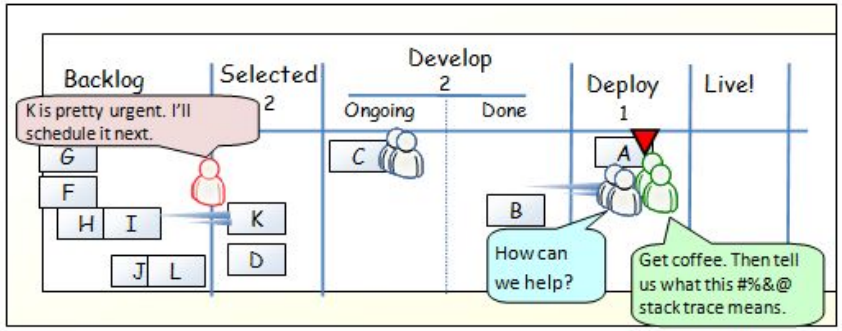
”One day in Kanban land”

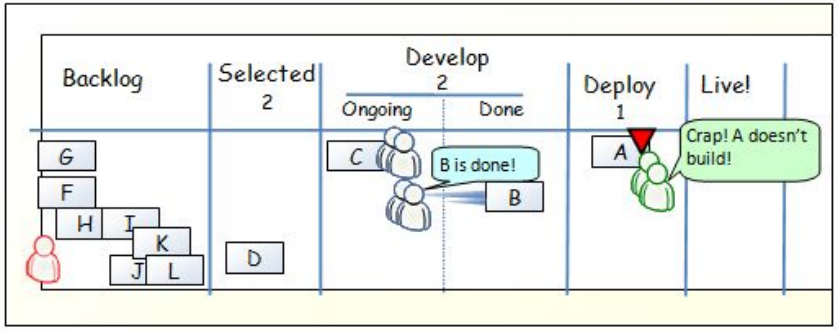
http://blog.crisp.se/henrikkniberg/tags/kanban/

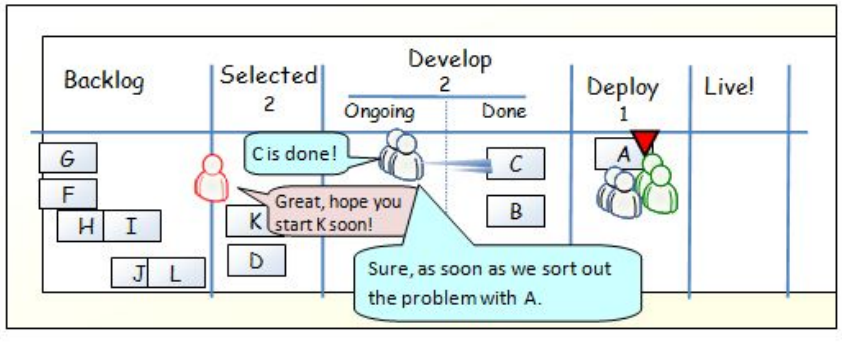




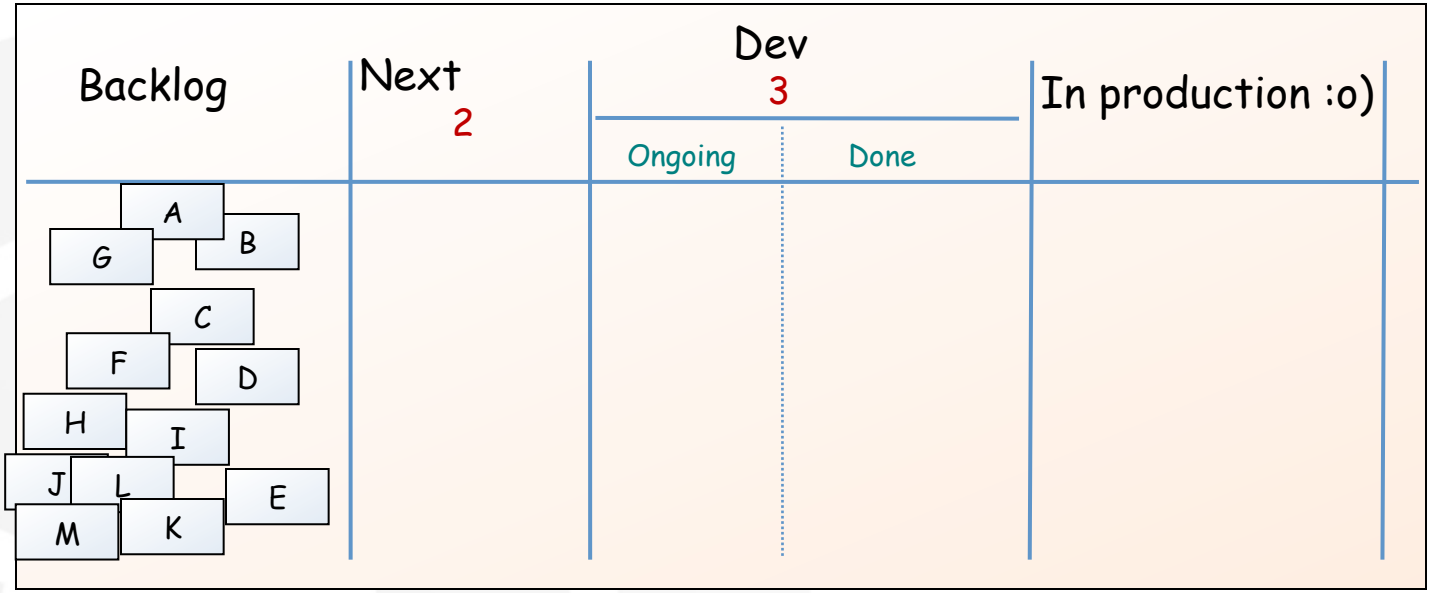




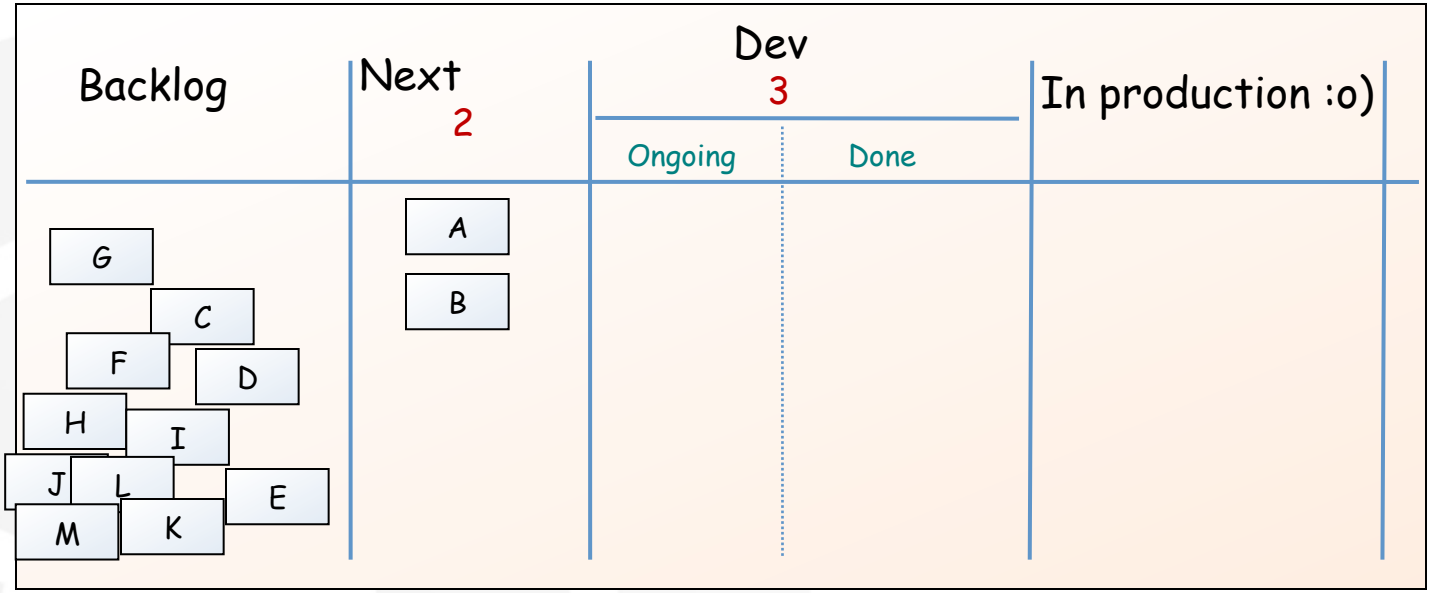




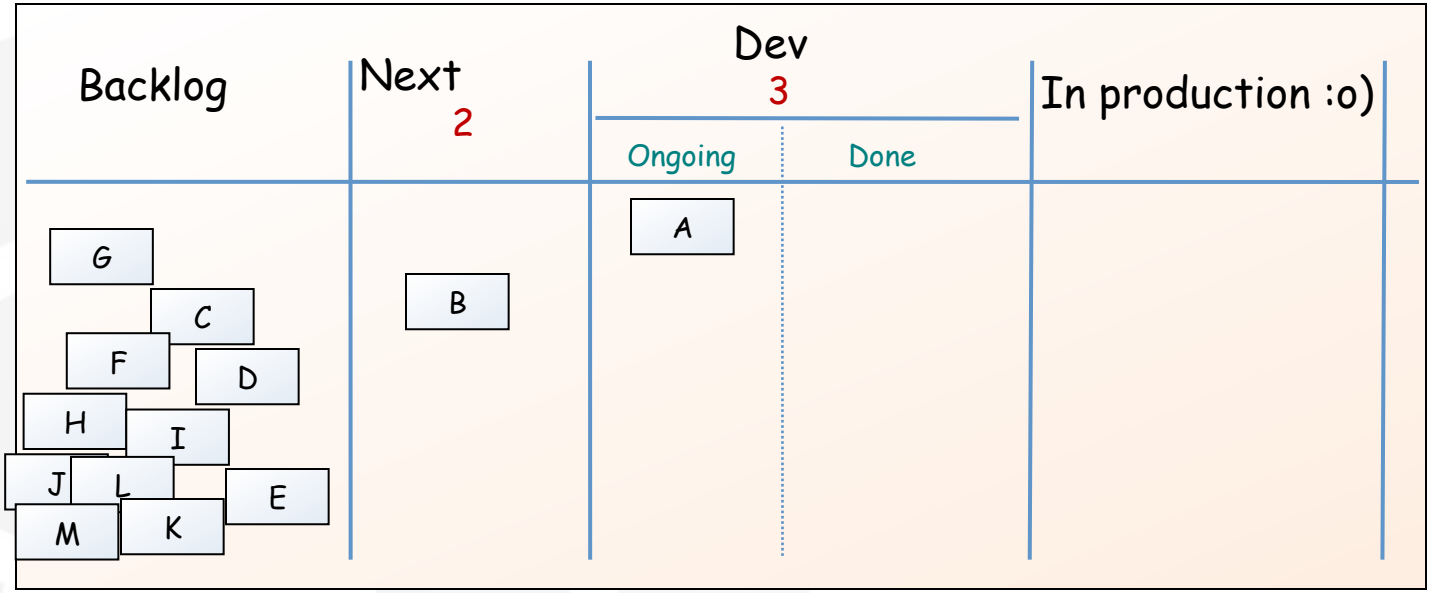
Scenario 1 – one piece flow



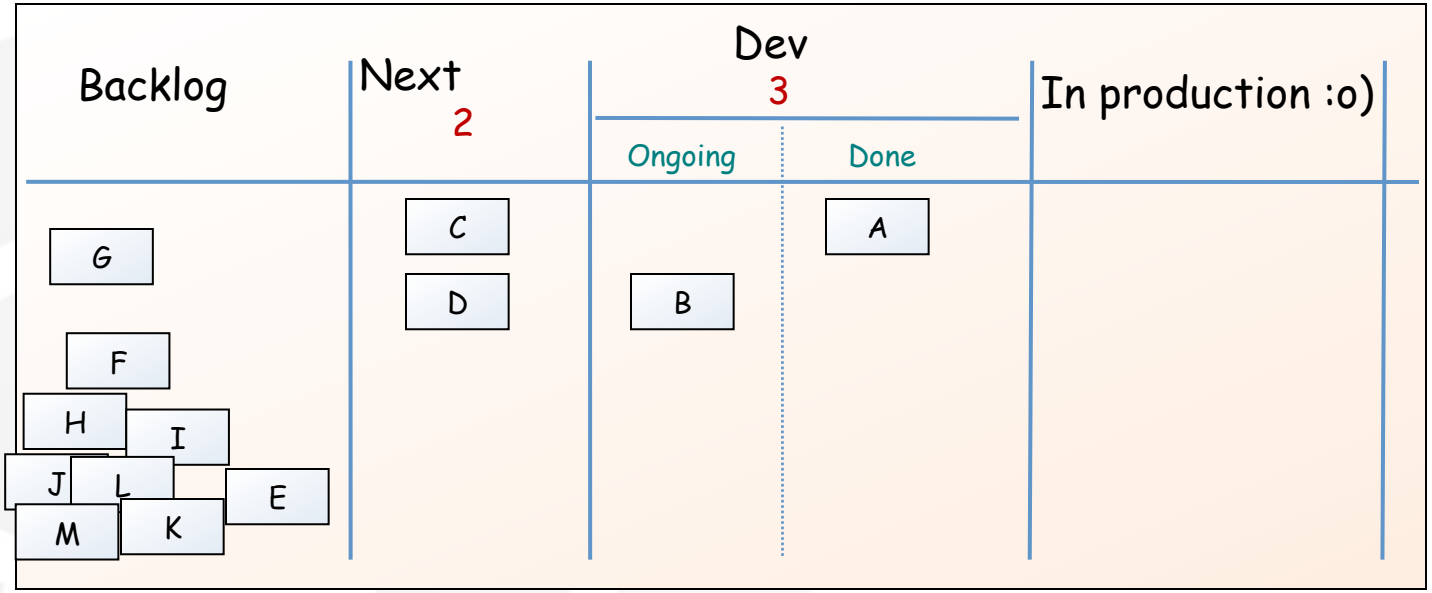
Scenario 1 – one piece flow



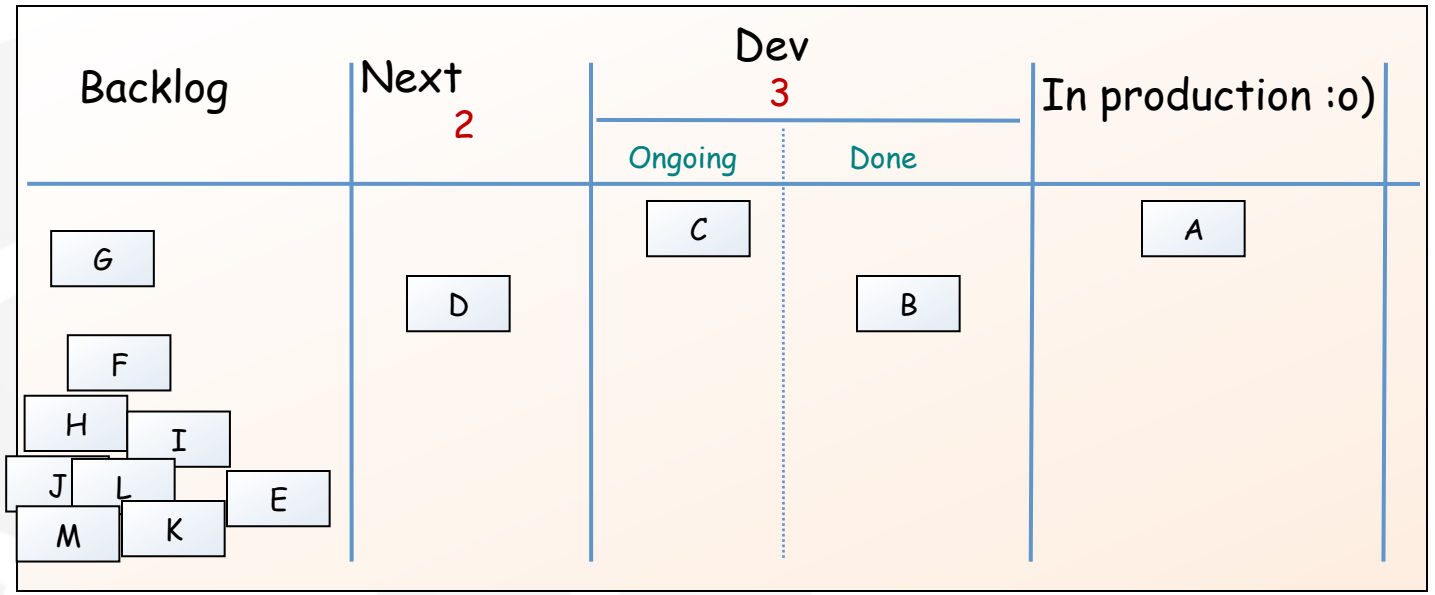
Scenario 1 – one piece flow



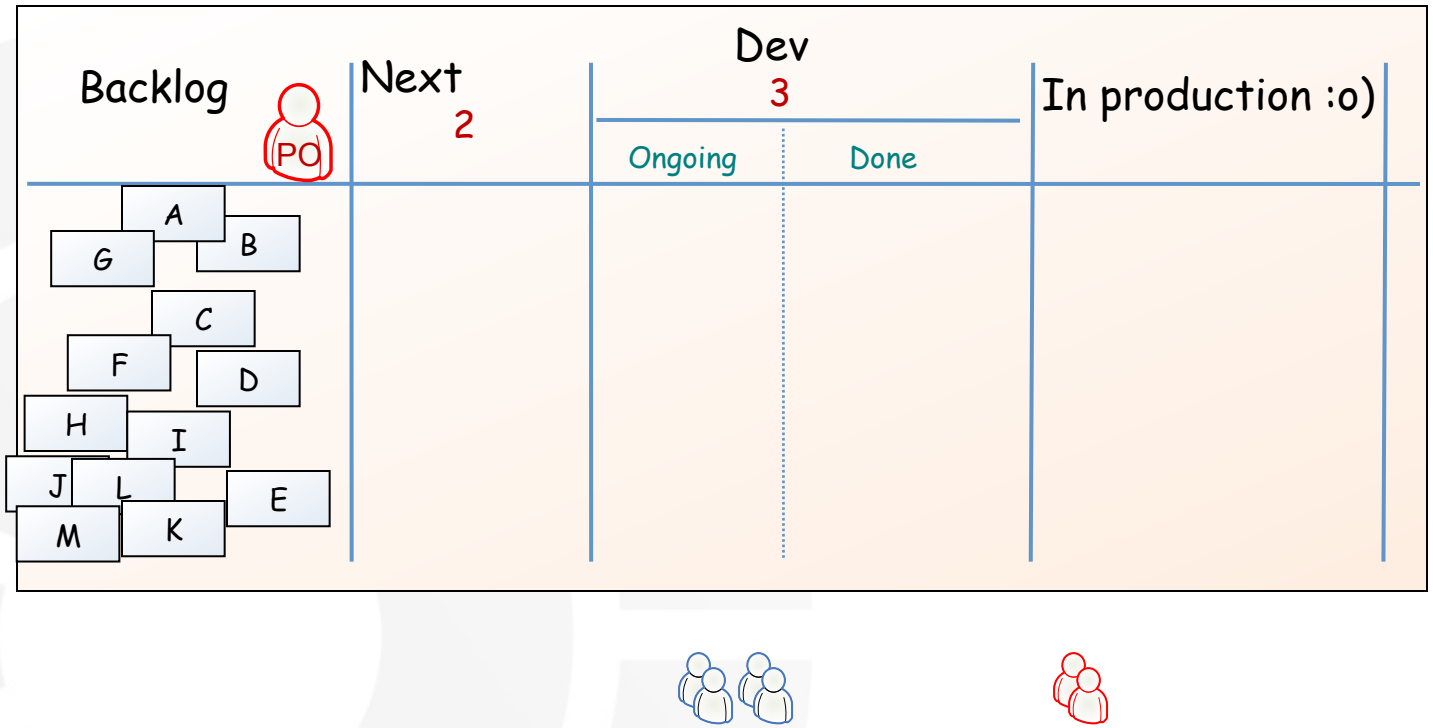
Scenario 1 – one piece flow



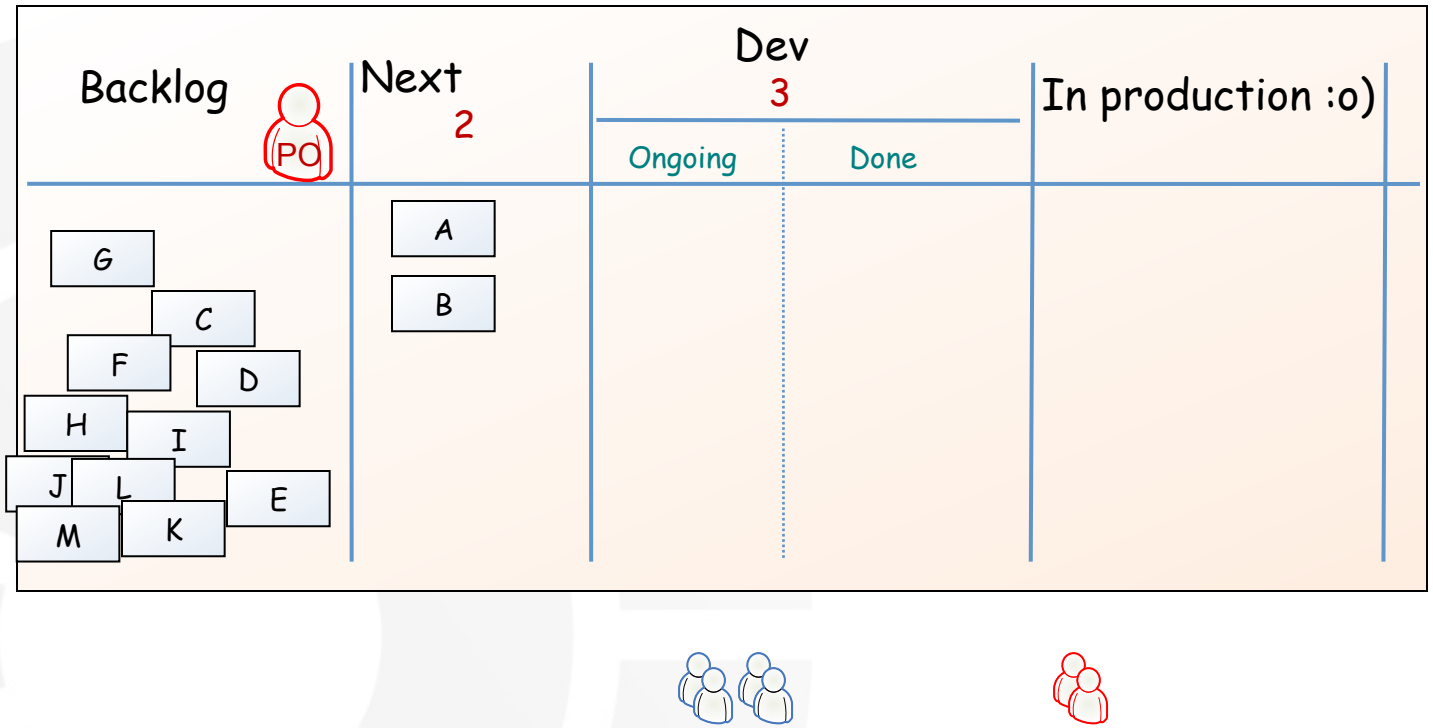
Scenario 1 – one piece flow.



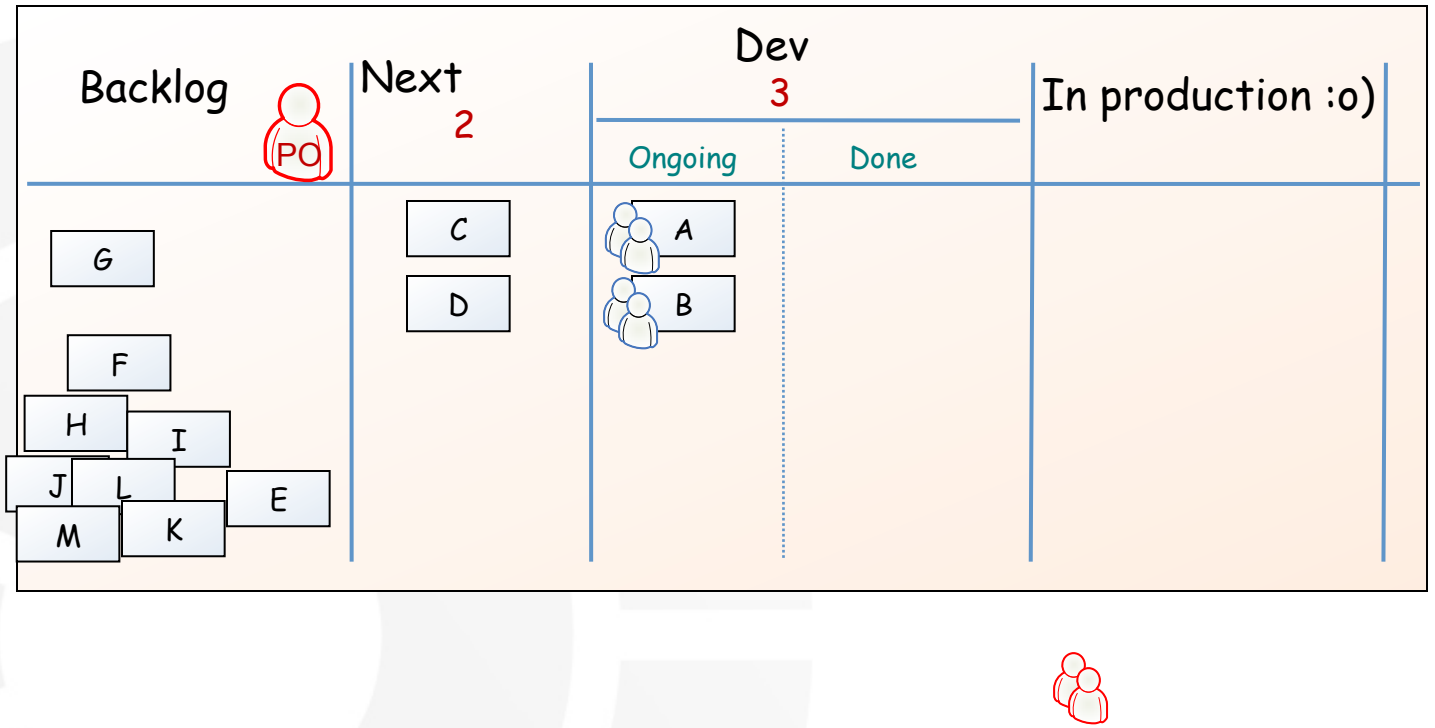
Scenario 2 – Deployment problem



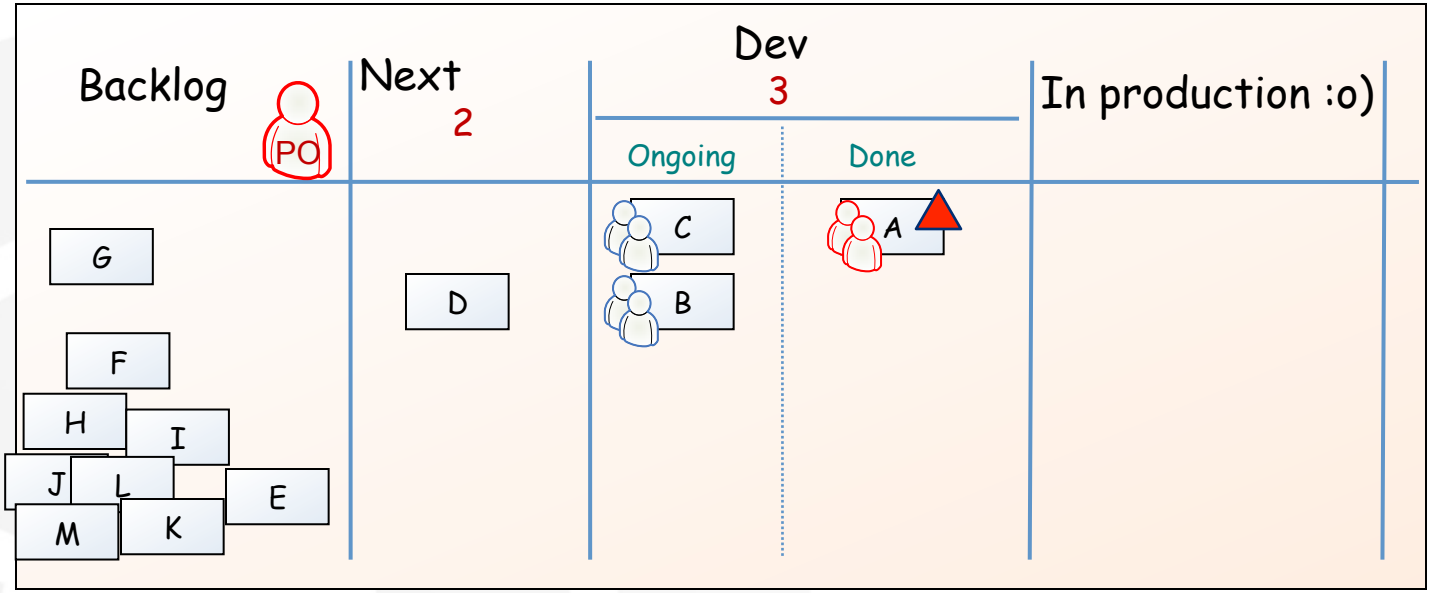
Scenario 2 – Deployment problem



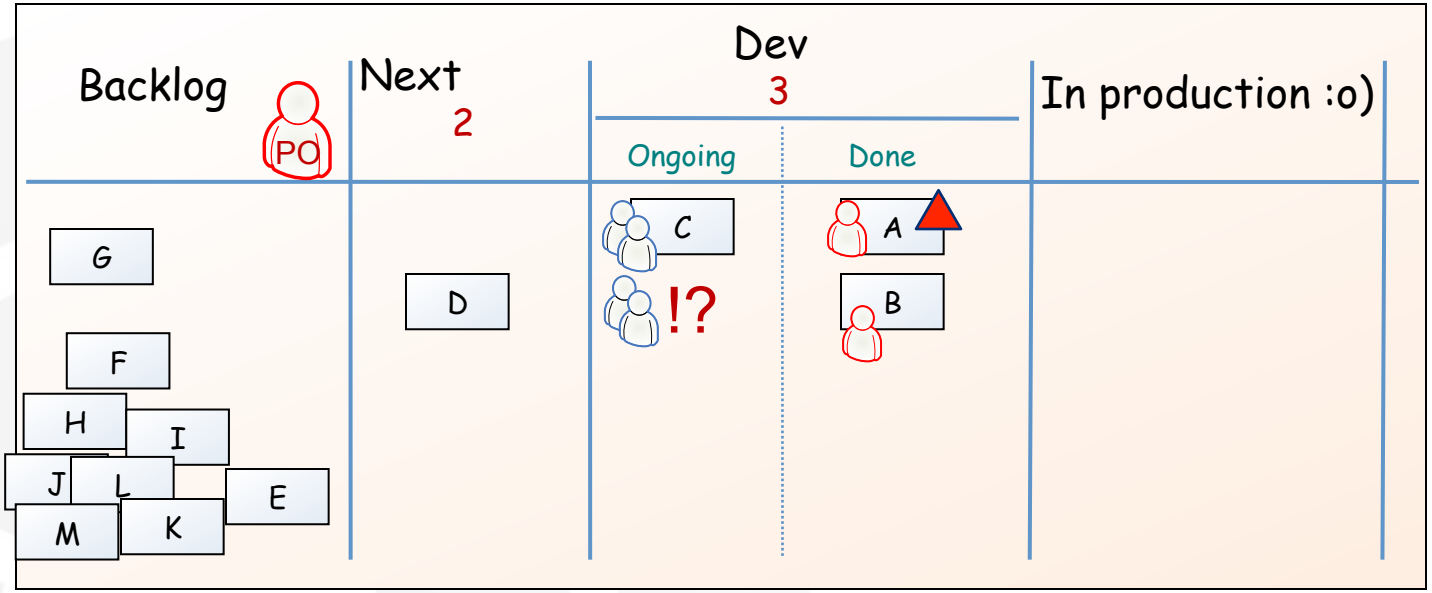
Scenario 2 – Deployment problem



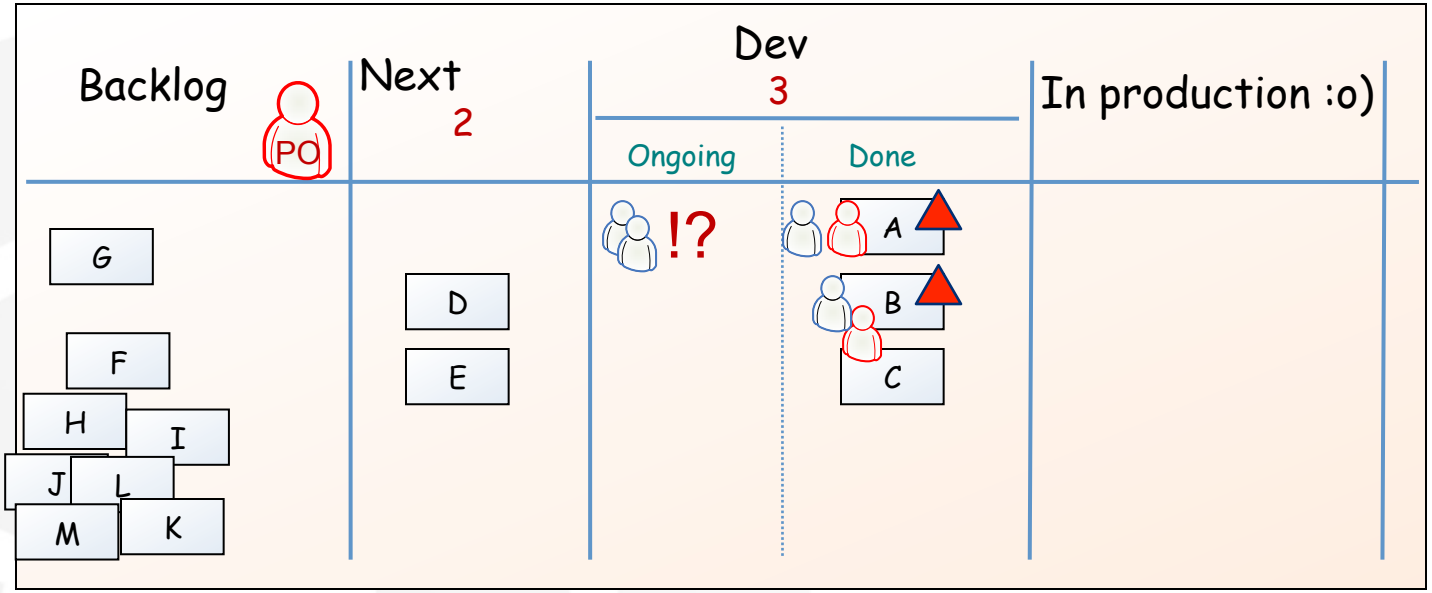
Scenario 2 – Deployment problem



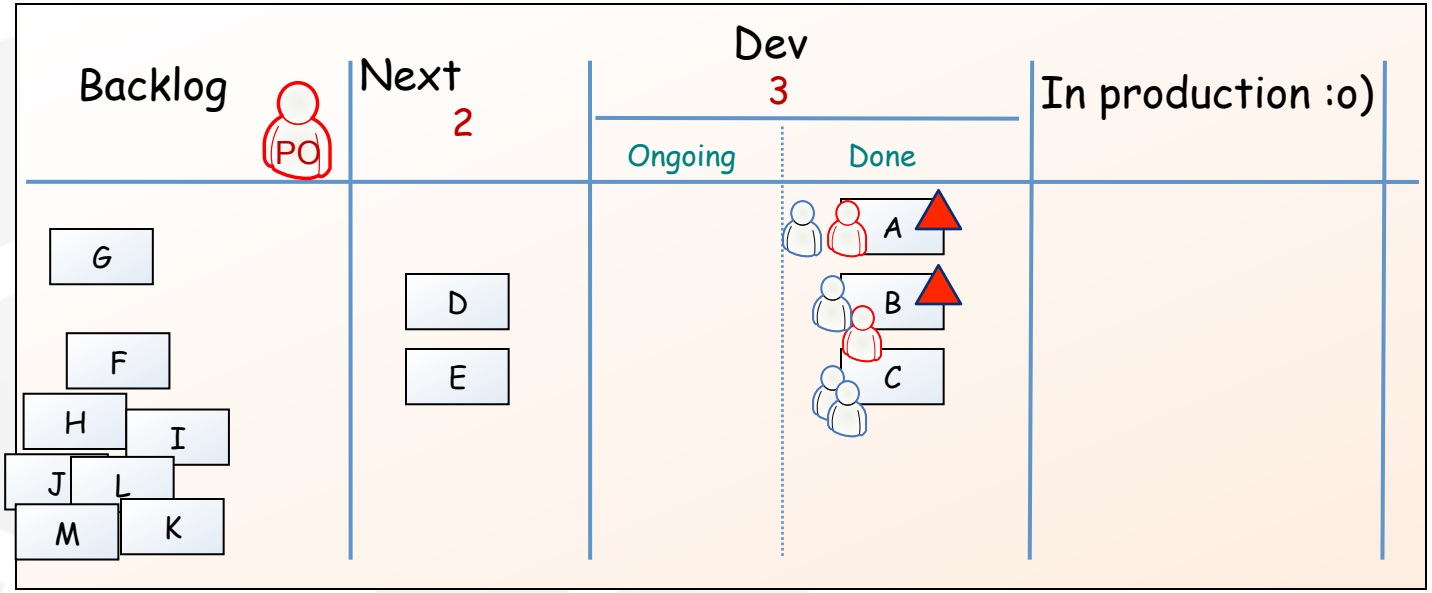
Scenario 2 – Deployment problem



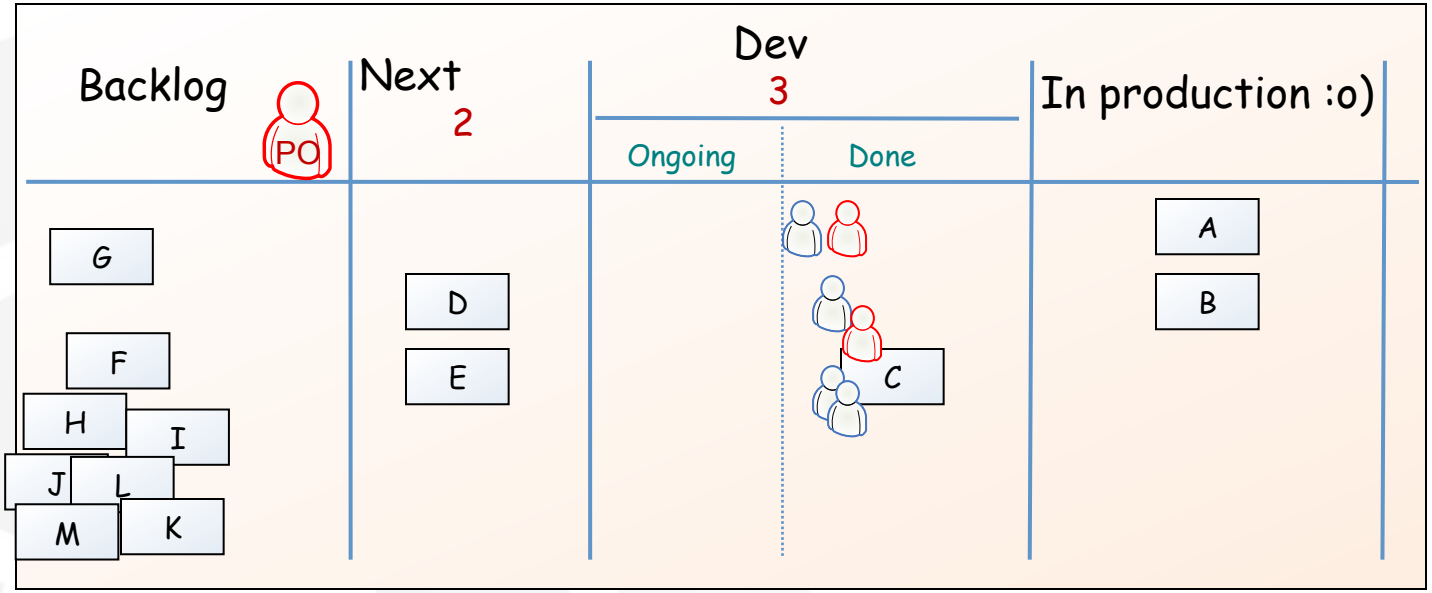
Scenario 2 – Deployment problem



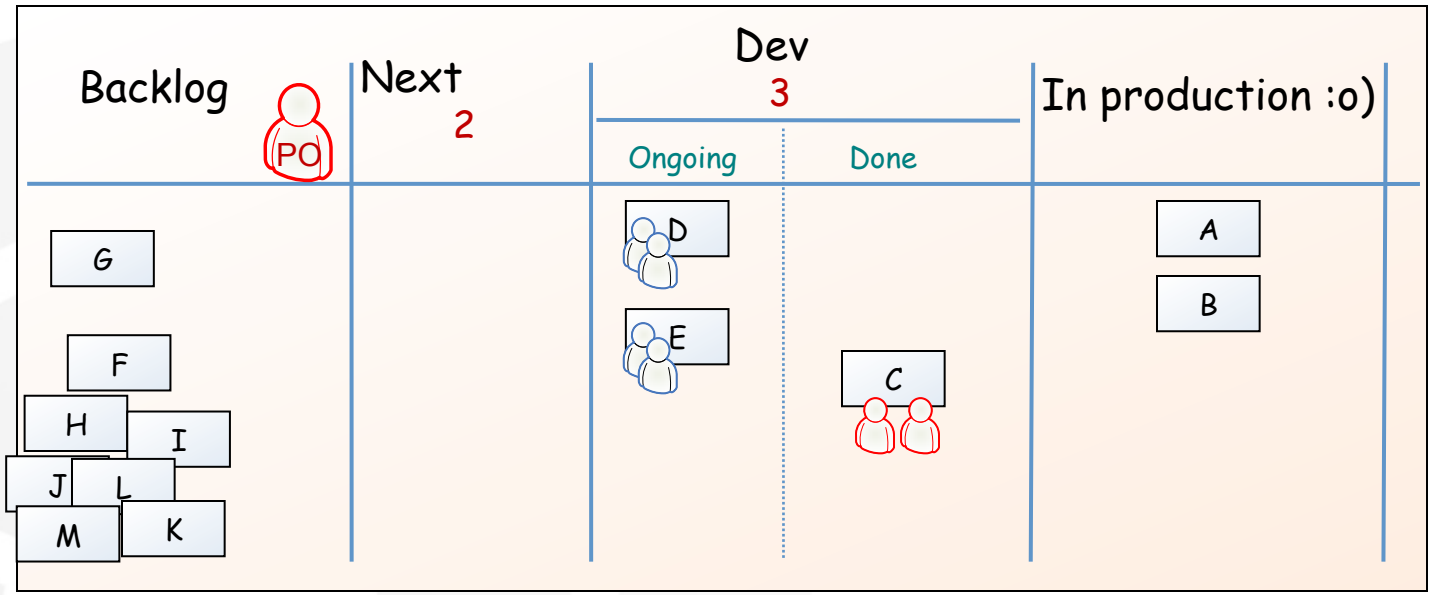
Scenario 2 – Deployment problem



Scenario 2 – Deployment problem



Scenario 2 – Deployment problem

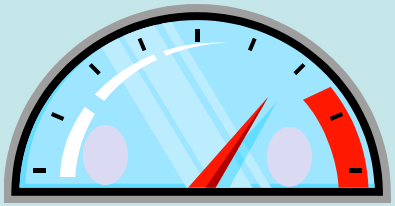


travelping

Evolve your process

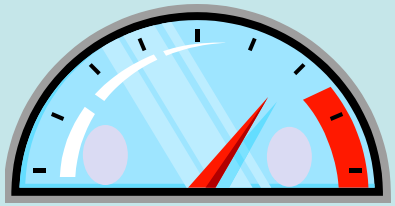
Kanban & Scrum are empirical

travelping

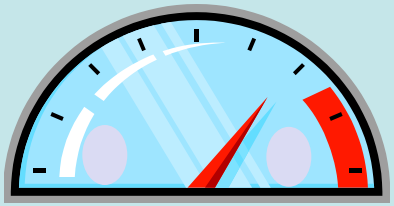


Capacity
(aka velocity)

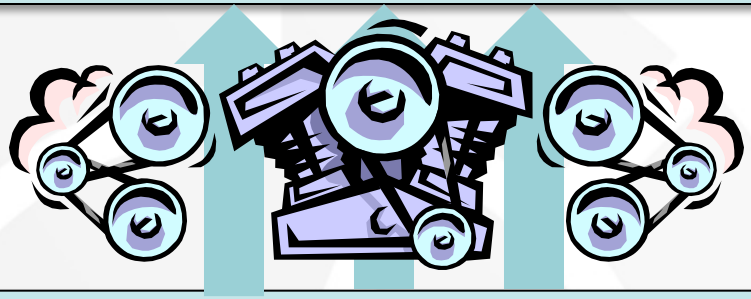
Lead time
(aka cycle time)



Quality
(defect rate, etc)

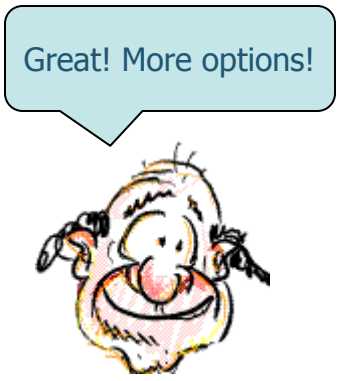


Predictability
(SLA fulfillment, etc)



Many small teams	Few large teams
Low WIP limits	High WIP limits
Few workflow states	Many workflow states
No iterations	Long iterations
Little planning	Lots of planning
.... etc ...	.... etc ...

Kanban is more configurable



Henrik Kniberg

Kanban kick-start example

version 1.2
2009-11-16

Next 2

2009-09-03
ipsum dolor sit amet, co nse otetur adi pis cing elit nisl

Analysis 3

Ongoing

2009-09-02
orem ipsum dolor sit amet, nse otetur adi pis elit nisl

orem ipsum dolor sit amet, co nse otetur

2009-09-02
orem ipsum dolor sit amet, co nse

Done

2009-09-01
orem ipsum dolor sit amet, co nse otetur

Definition of Done:
- Goal is clear
- First tasks defined
- Story split (if necessary)

Development 3

Ongoing

2009-08-30 2009-09-08
orem ipsum dolor sit amet, co adi pis cing elit nisl

2009-08-29
orem ipsum dolor sit amet, nse otetur adi pis cing elit

orem ipsum dolor sit amet, co nse otetur

Done

2009-08-27
orem ipsum dolor sit amet, otetur adi pis cing elit nisl

Definition of Done:
- Code clean & checked in on trunk
- Integrated & regression tested
- Running on UAT environment

Acceptance 2

Ongoing

2009-08-27
orem ipsum dolor sit amet, adi pis cing elit nisl

Done

2009-08-26
orem adi pis cing elit nisl

Definition of Done:
- Customer accepted
- Ready for production

Prod

2009-08-20
orem olor sit amet, co nse otetur adi pis cing elit nisl

2009-08-20 2009-08-25
orem ipsum dolor sit amet, co nse otetur adi pis cing elit nisl

2009-08-22
orem ipsum dolor sit amet, co

2009-08-25
orem ipsum dolor sit otetur adi pis cing elit nisl

Feature / story

Date when added to board

Hard deadline (if applicable)

2009-08-20 2009-09-30

(description)

★ = priority

★★ = panic

Who is analyzing / testing right now

Task / defect

(description) =task (description) =defect

= completed

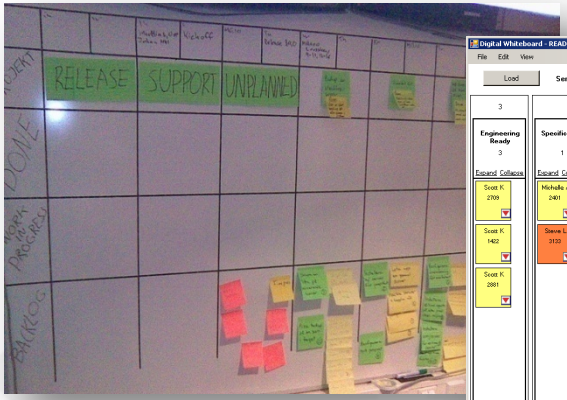
why = blocked

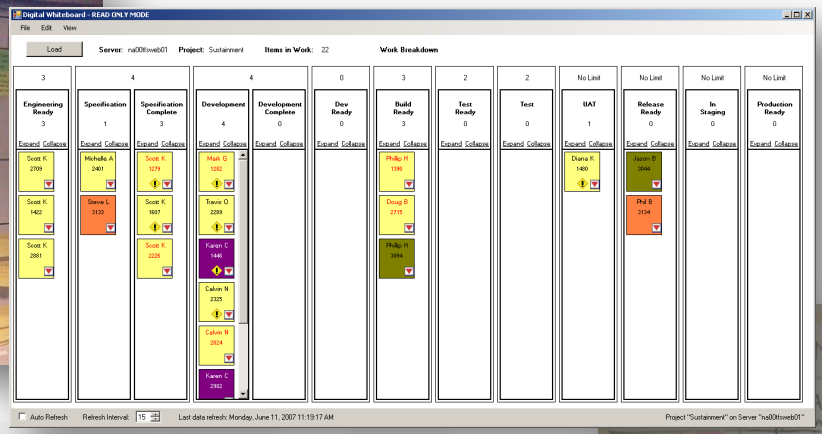
= who is doing this right now

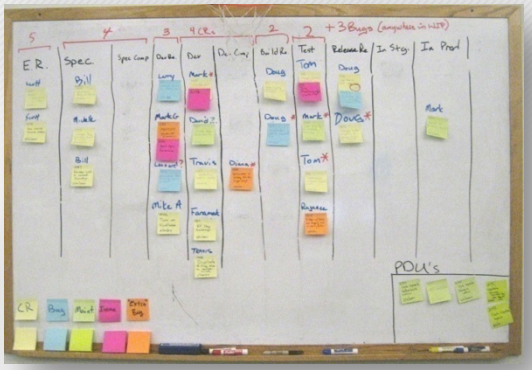
What to pull first

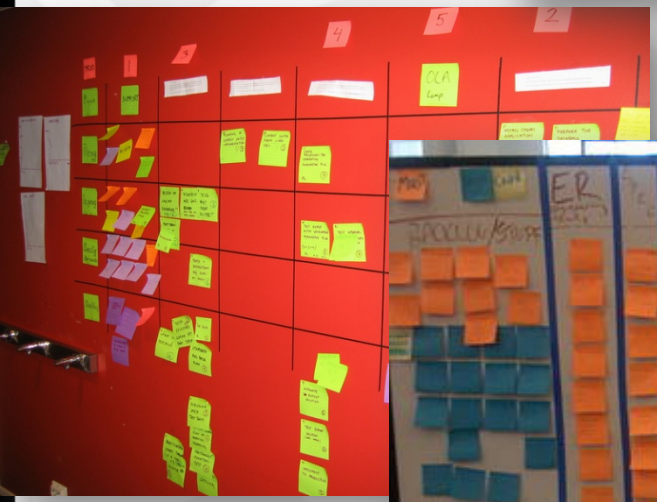
- **Panic**features ★★
  (should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
- **Priority** features ★
- Hard **deadline** features
  (only if deadline is at risk)
- **Oldest** features

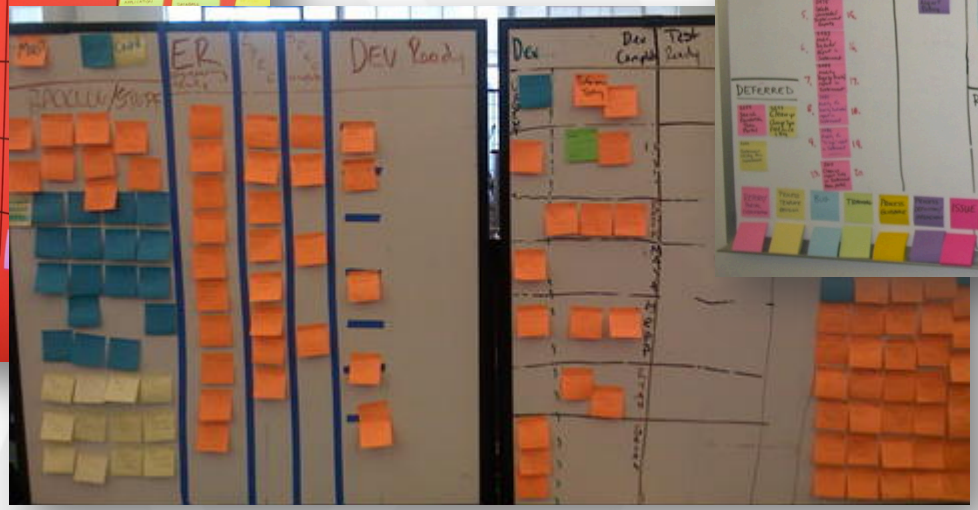
Evolve your own unique board!











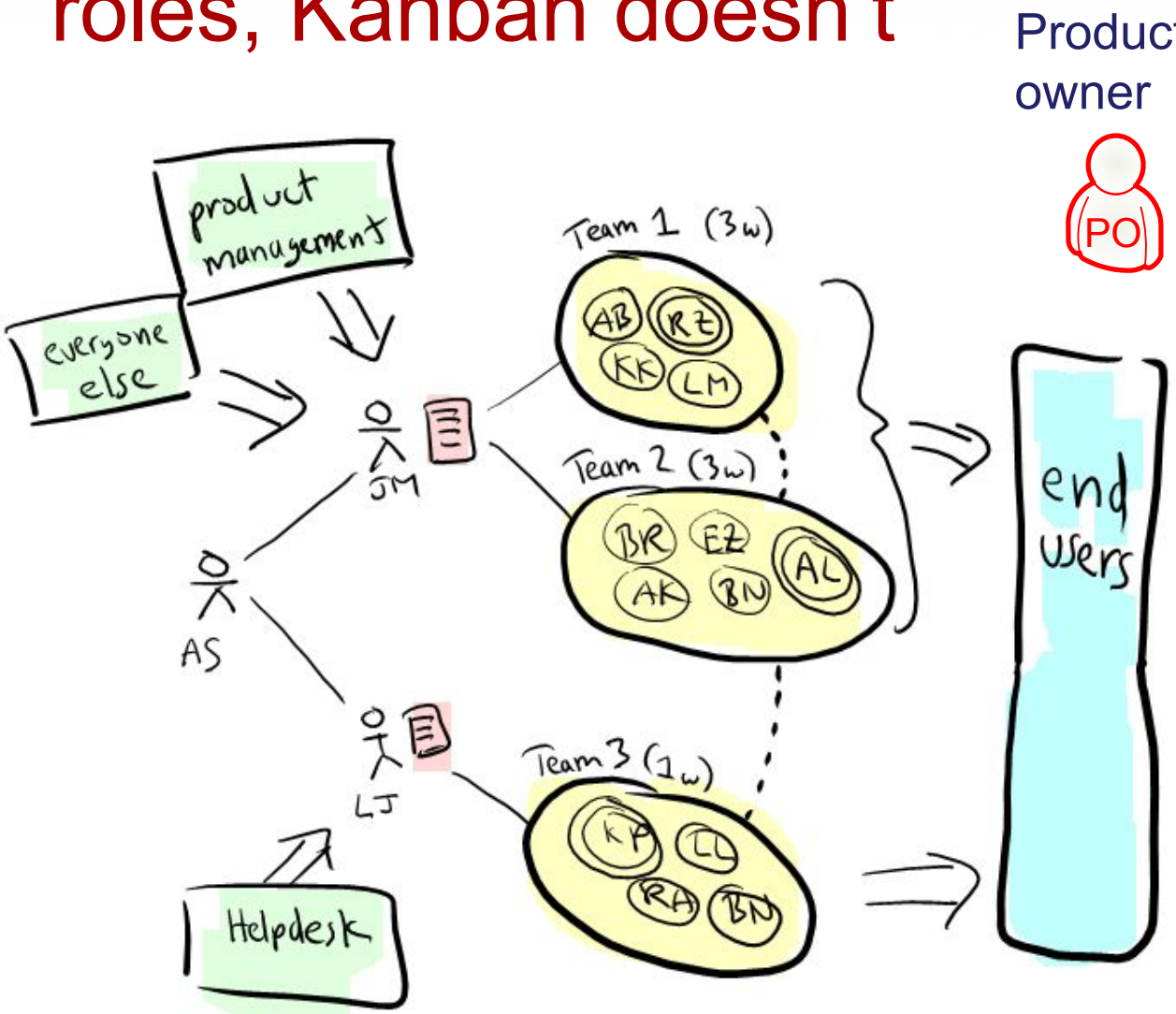
Some of these photos courtesy of David Anderson, Mattias Skarin, and various other people

travelping

Kanban compared to Scrum

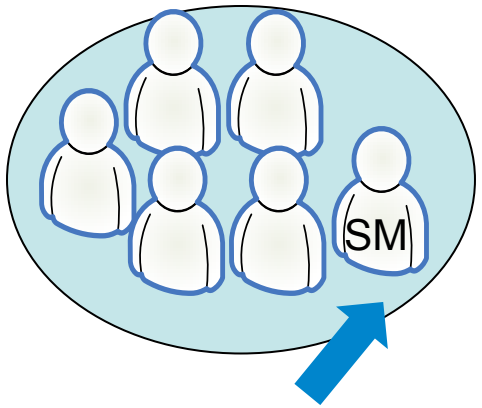
Scrum prescribes roles, Kanban doesn't

travelping



Product owner

Team



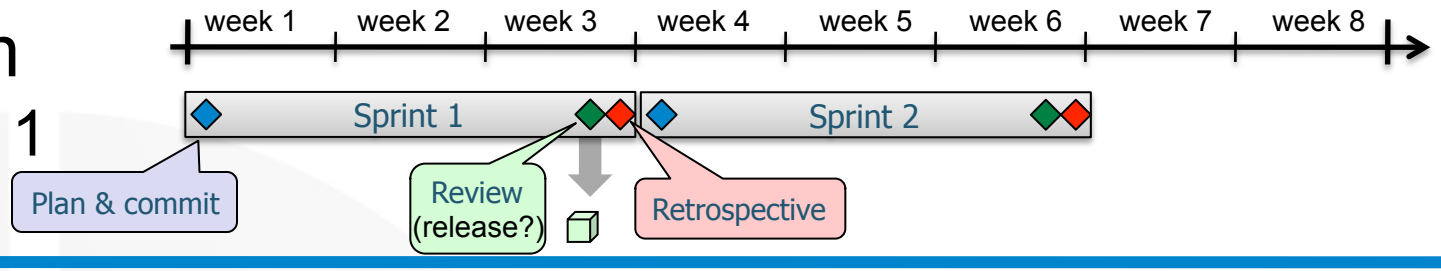
Scrum Master

Scrum prescribes timeboxed iterations

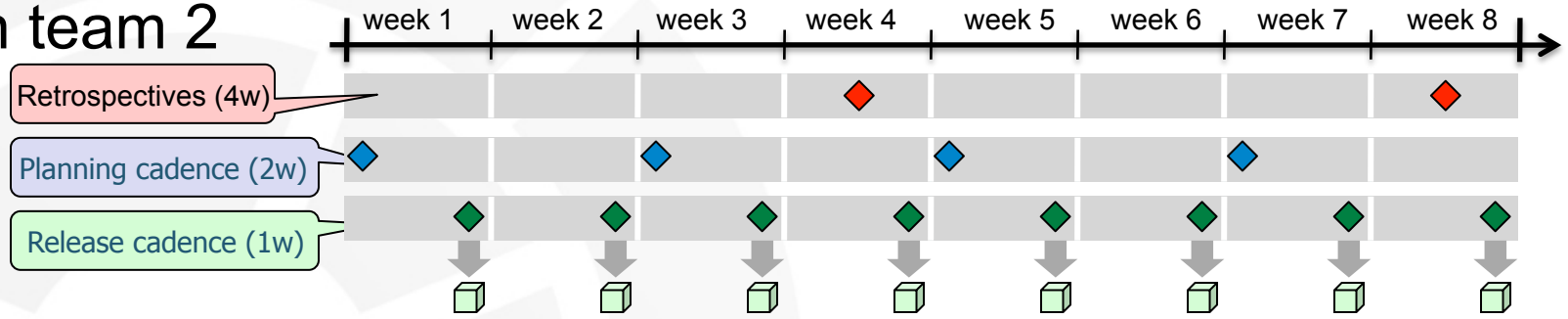
travelping

Scrum team

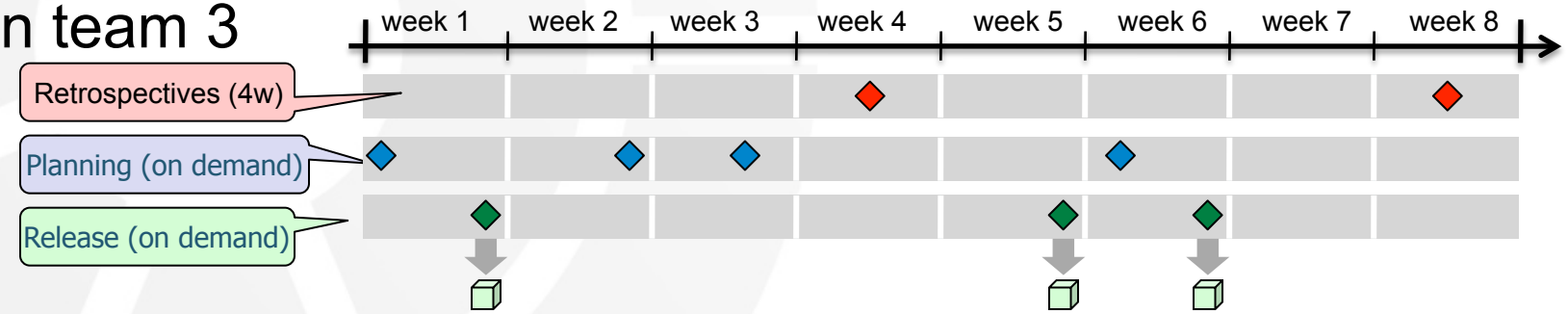
Kanban team 1



Kanban team 2



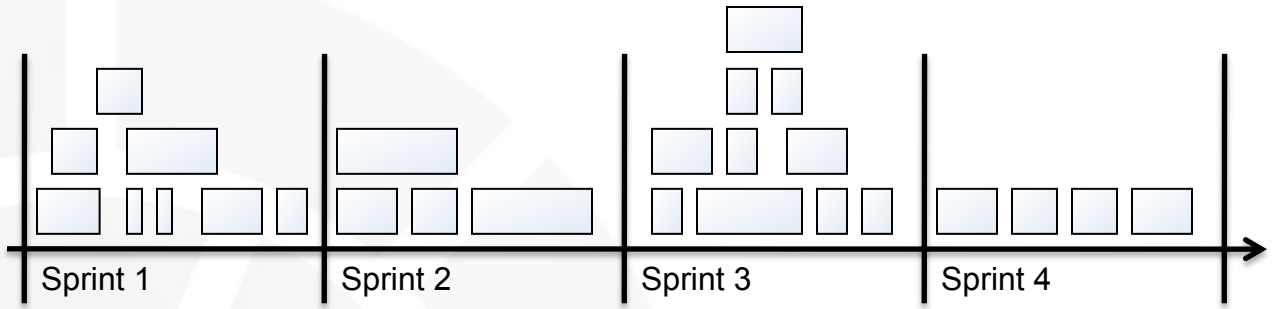
Kanban team 3



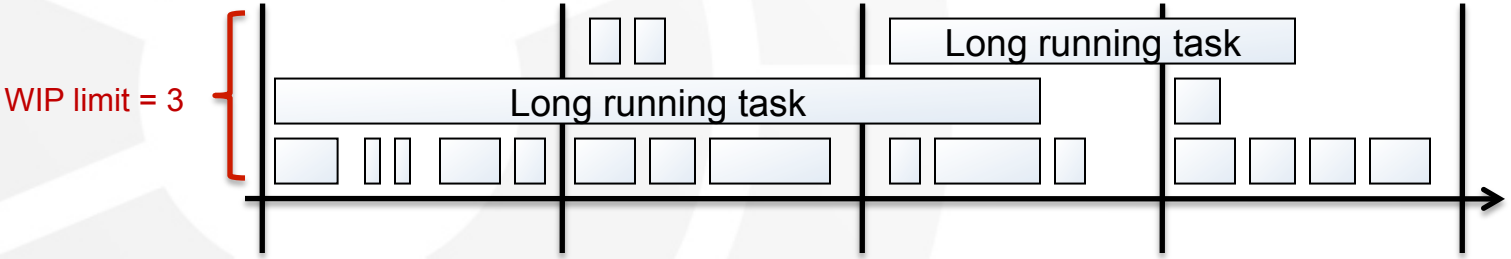
Scrum backlog items must fit in a sprint

travelping

Scrum



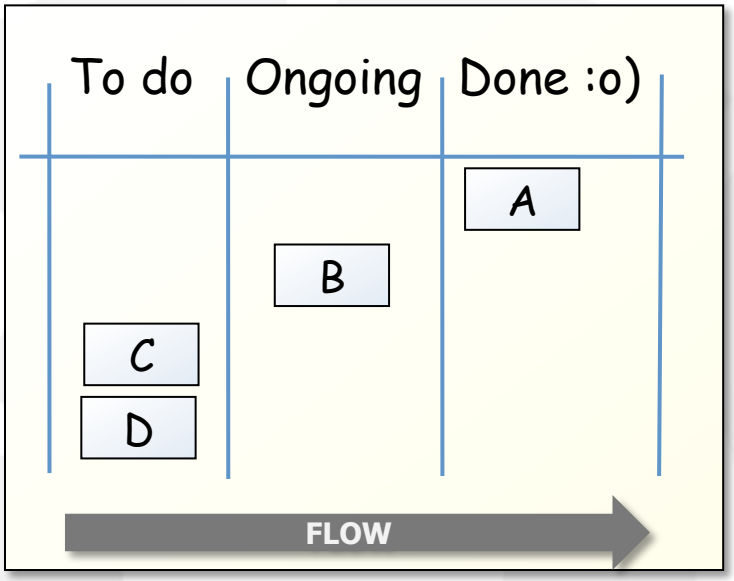
Kanban



Both limit WIP but in different ways

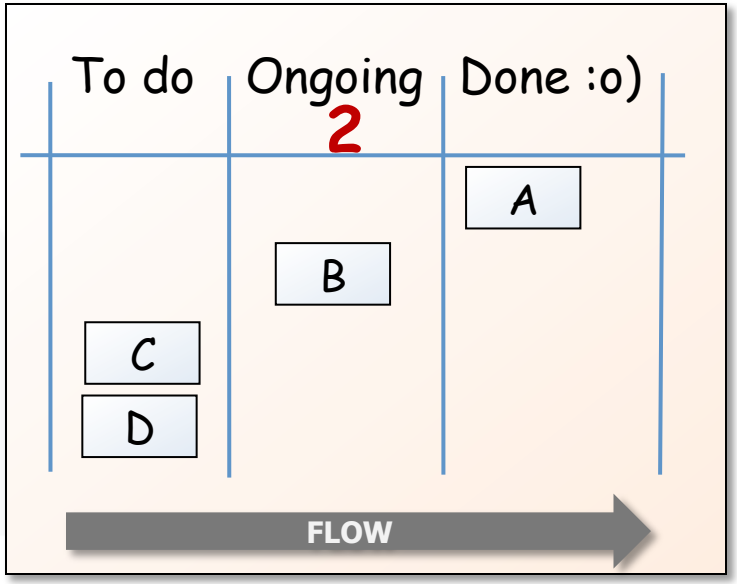
travelping

Scrum board



WIP limited per unit of time (iteration)

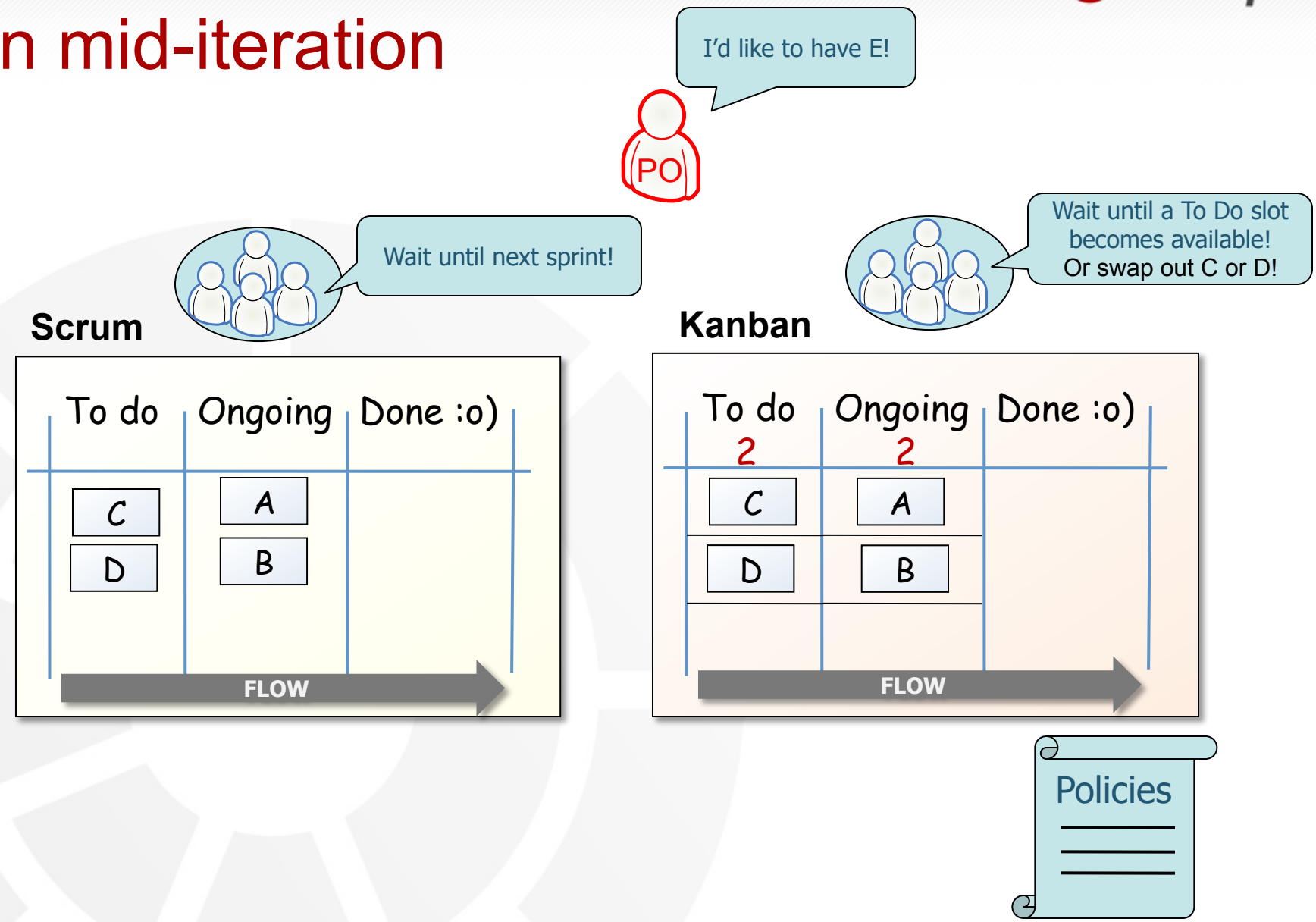
Kanban board



WIP limited per workflow state

Scrum discourages change in mid-iteration

travelping

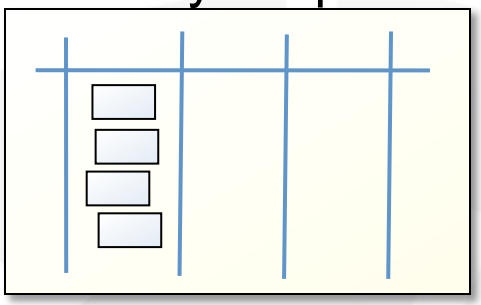


Scrum board is reset between each iteration

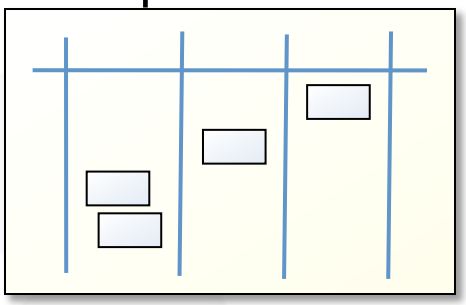
travelping

Scrum

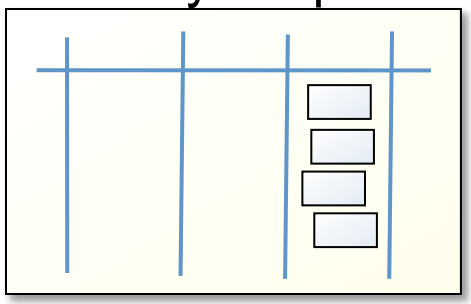
First day of sprint



Mid-sprint

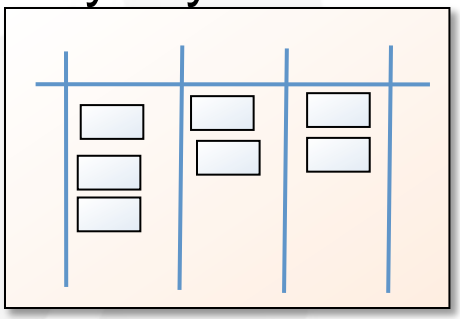


Last day of sprint



Kanban

Any day



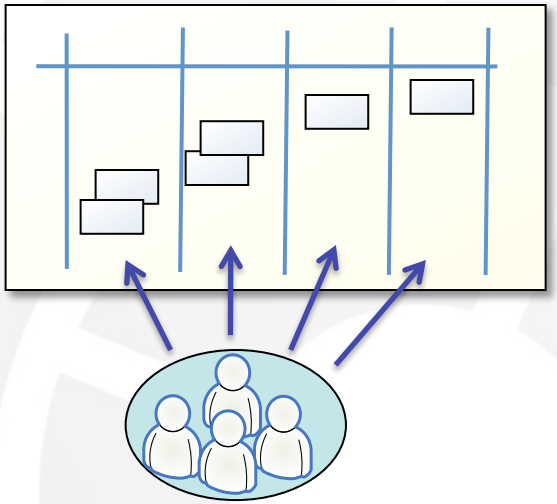
Scrum prescribes cross-functional teams

Kanban supports both specialists & generalists

travelping

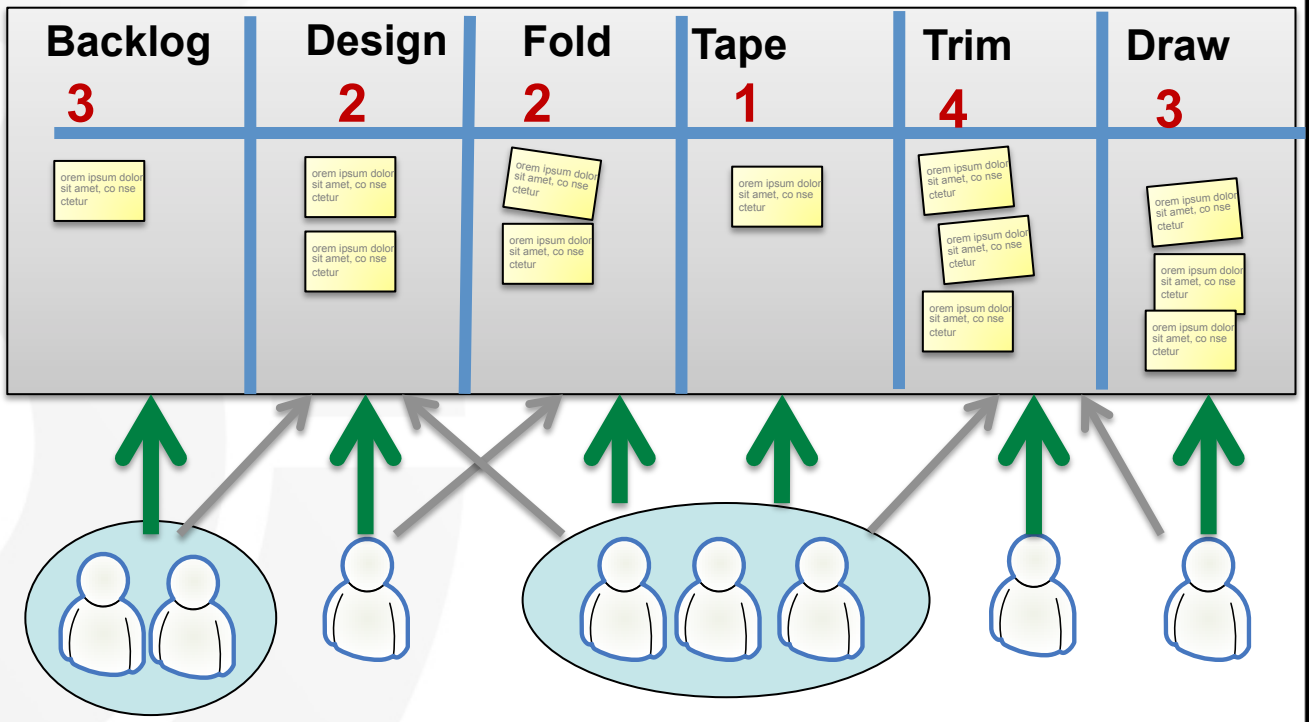
Scrum team

Kanban team 1



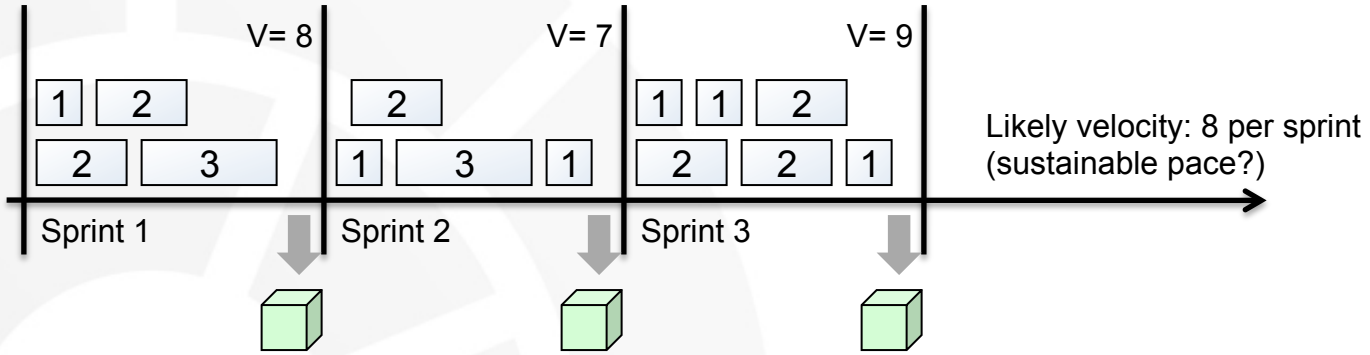
Cross-functional team

Kanban team 2



Scrum prescribes estimation and velocity

travelping

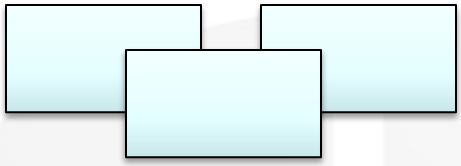


Estimation is flexible in both Kanban & Scrum

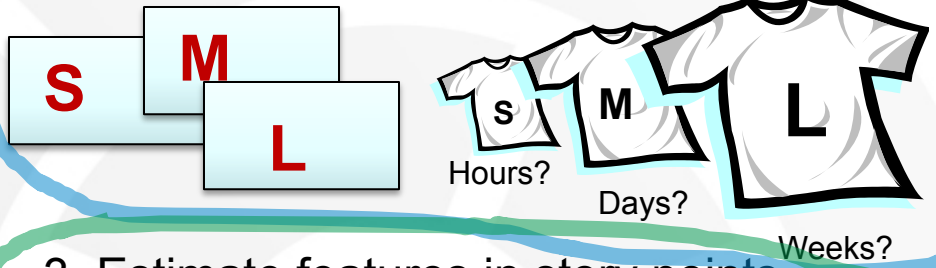
travelping

Features

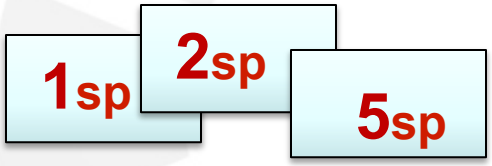
1. Don't estimate features. Just count them.



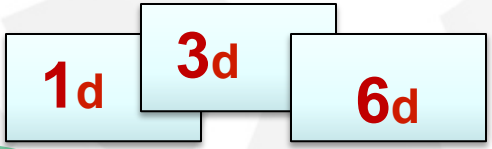
2. Estimate features in t-shirt size



3. Estimate features in story points



4. Estimate features in ideal man-days

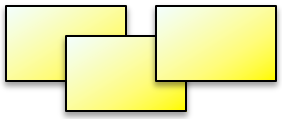


Tasks

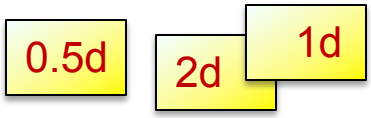
1. Skip tasks

"Typical" Kanban

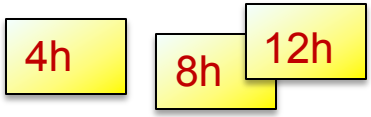
2. Don't estimate tasks. Just count them.



3. Estimate tasks in days



4. Estimate tasks in hours

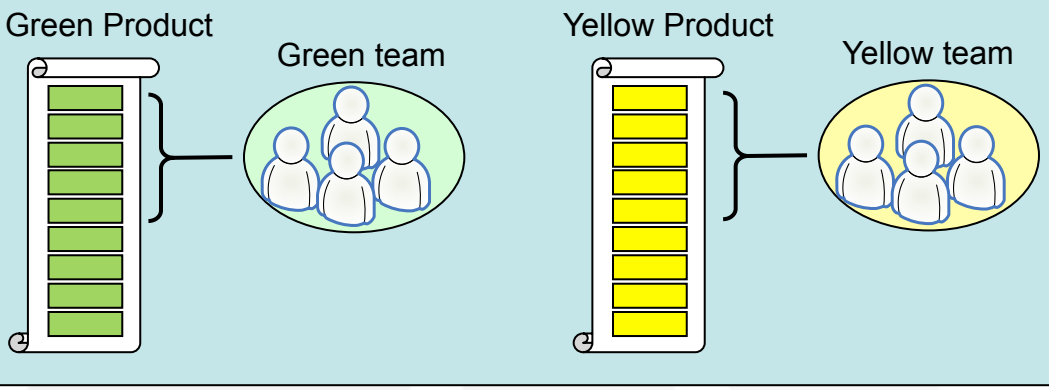


"Typical" Scrum

Both allow working on multiple products simultaneously (if you must…)

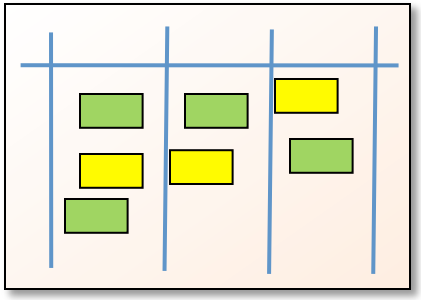
travelping

Scrum example 1

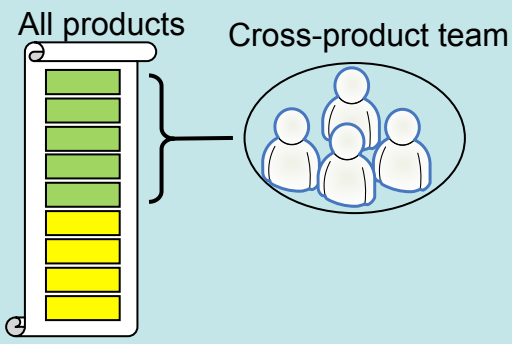


Kanban example 1

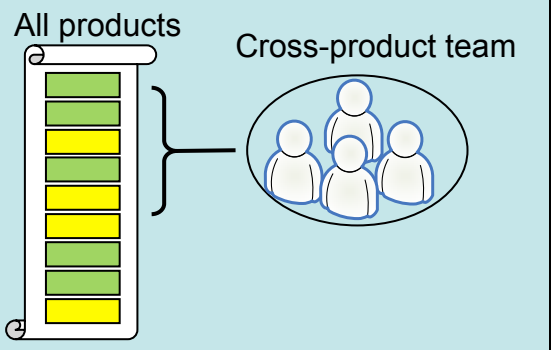
Color-coded tasks



Scrum example 2

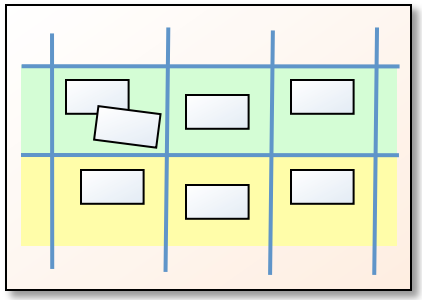


Scrum example 3



Kanban example 2

Color-coded swimlanes



travelping

Minor differences

travelping

Minor difference:

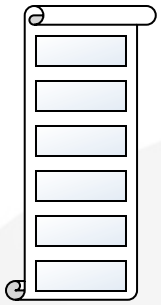
Scrum prescribes daily meetings

... but many Kanban teams do that anyway.

Minor difference:

Scrum prescribes a prioritized product backlog

travelping

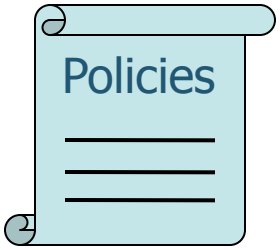


Scrum:

- 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 maintainance items, 80% on new features**
  - **Split capacity evenly between product A and product B**
  - **Always take red items first**



travelping

Example...

Your turn ...

travelping

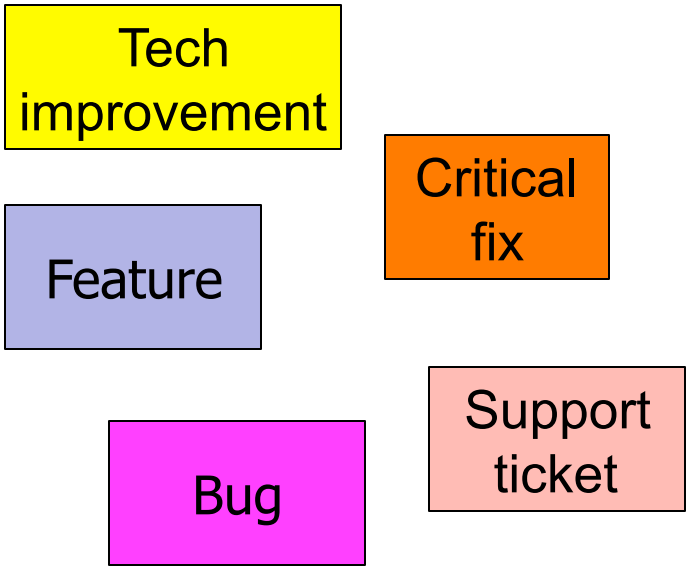
Identify a value stream that you want to improve

- Who is the customer?
- What is the input?
- What is the output?
- What is the approximate cycle time today?

Your turn

travelping

Identify item types



Sample distinguishers:

- Different urgency/size?
- Different lifecycle?
- Different cost of delay?

Kanban kick-start example

Henrik Kniberg

www.crisp.se/kanban/example

version 1.2
2009-11-16

Next 2

2009-09-03
ipsum dolor sit amet, co nse otetur adi pis cing elit nisl

Analysis 3

Ongoing

2009-09-02
orem ipsum dolor sit amet, nse otetur adi pis elit nisl

orem ipsum dolor sit amet, co nse otetur

2009-09-02
orem ipsum dolor sit amet, co nse

Done

2009-09-01
orem ipsum dolor sit amet, co nse otetur

Definition of Done:
- Goal is clear
- First tasks defined
- Story split (if necessary)

Development 3

Ongoing

2009-08-30 2009-09-08
orem ipsum dolor sit amet, co adi pis cing elit nisl

2009-08-29
orem ipsum dolor sit amet, nse otetur adi pis cing elit

orem ipsum dolor sit amet, co nse otetur

Done

2009-08-27
orem ipsum dolor sit amet, otetur adi pis cing elit nisl

Definition of Done:
- Code clean & checked in on trunk
- Integrated & regression tested
- Running on UAT environment

Acceptance 2

Ongoing

2009-08-27
orem ipsum dolor sit amet, adi pis cing elit nisl

Done

2009-08-26
orem adi pis cing elit nisl

Definition of Done:
- Customer accepted
- Ready for production

Prod

2009-08-20
orem olor sit amet, co nse otetur adi pis cing elit nisl

2009-08-20 2009-08-25
orem ipsum dolor sit amet, co nse otetur adi pis cing elit nisl

2009-08-22
orem ipsum dolor sit amet, co

2009-08-25
orem ipsum dolor sit otetur adi pis cing elit nisl

Feature / story

Date when added to board

Hard deadline (if applicable)

2009-08-20 2009-09-30

(description)

★ = priority

★★ = panic

Who is analyzing / testing right now

Task / defect

(description) =task (description) =defect

= completed

why = blocked

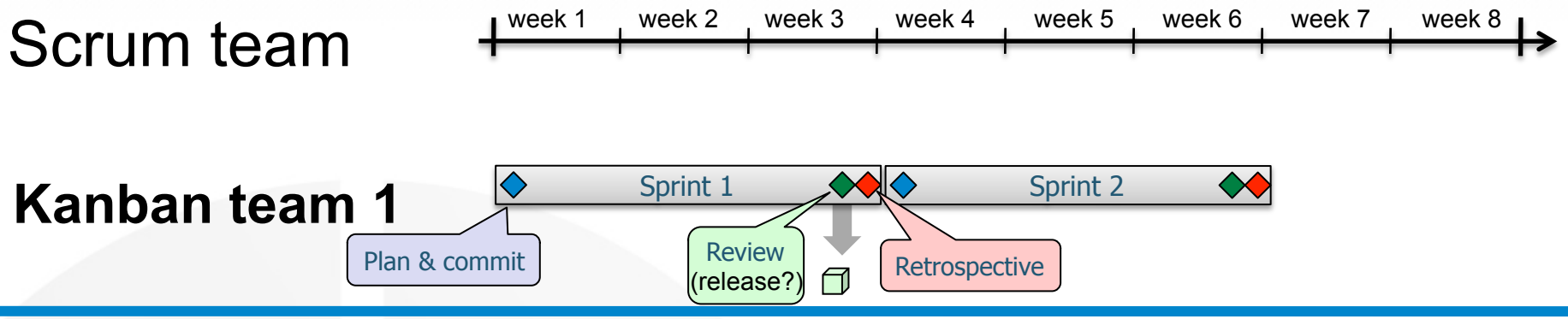
= who is doing this right now

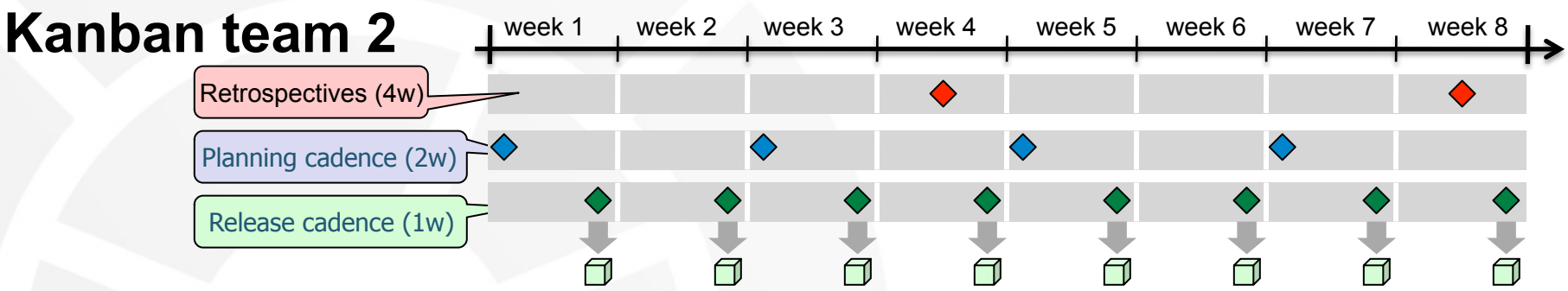
What to pull first

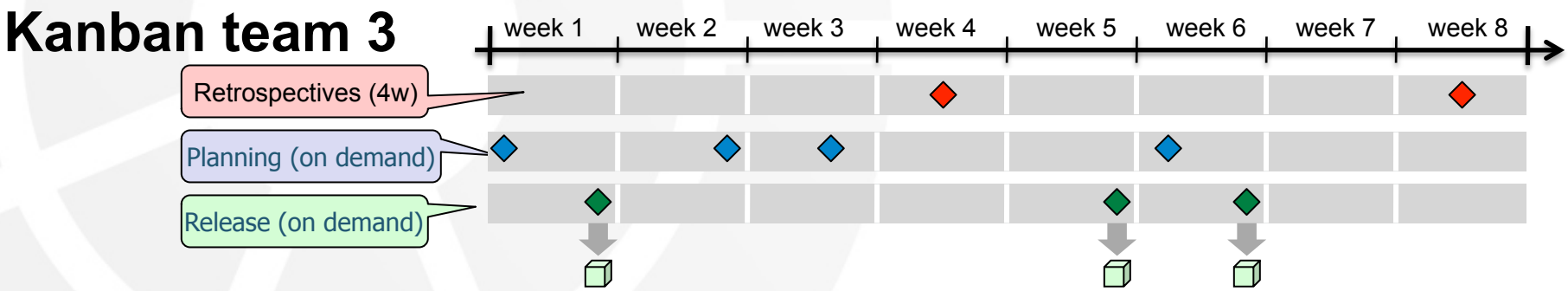
- **Panic**features ★★
  (should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
- **Priority** features ★
- Hard **deadline** features
  (only if deadline is at risk)
- **Oldest** features

Input & output coordination

travelping







Your turn

travelping

Input coordination

How & when do items enter your Kanban system?

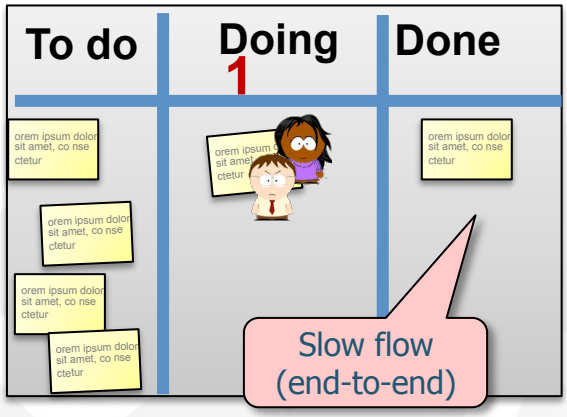
Output coordination

How & when do items exit your Kanban system?

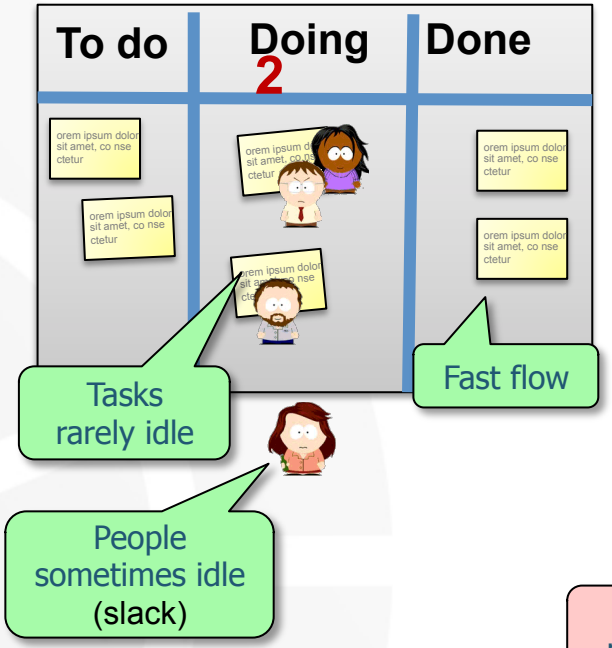
Optimizing the WIP limit

travelping

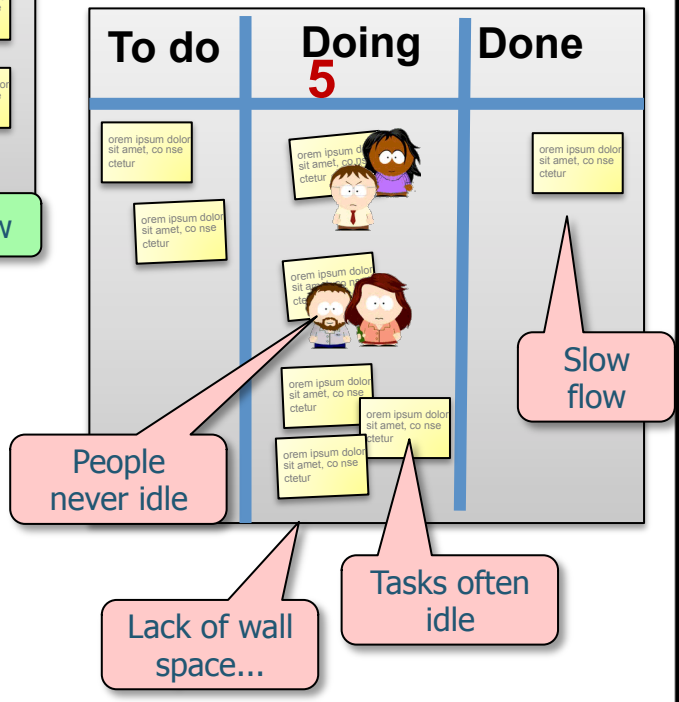
Too low WIP limit



Just Right WIP limit



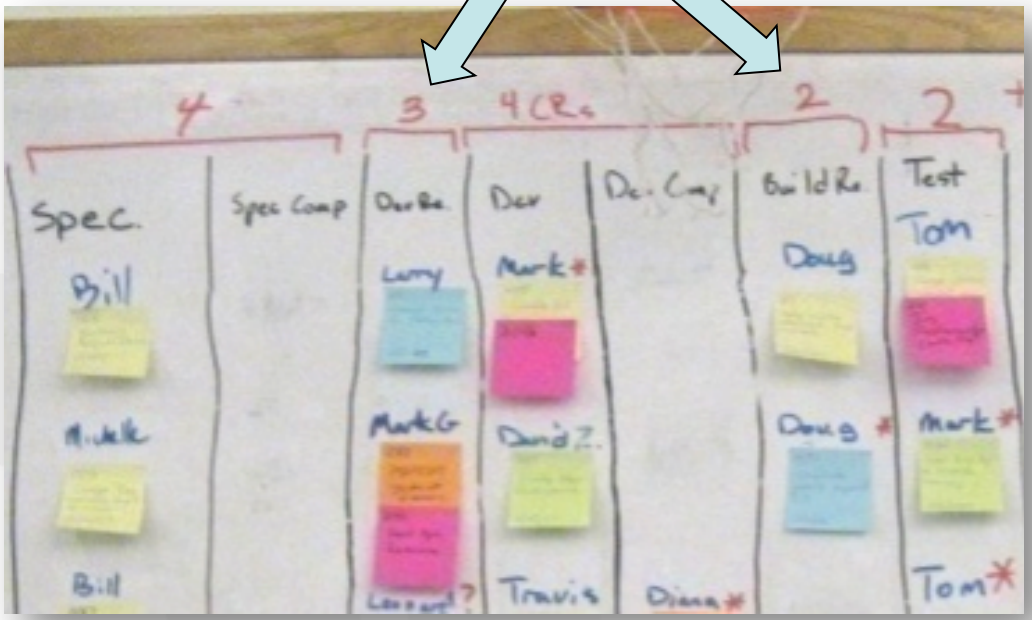
Too high WIP limit



Your turn

travelping

Set WIP limits



travelping

Final points

Kanban & Scrum

Comparison summary

travelping

Similarities

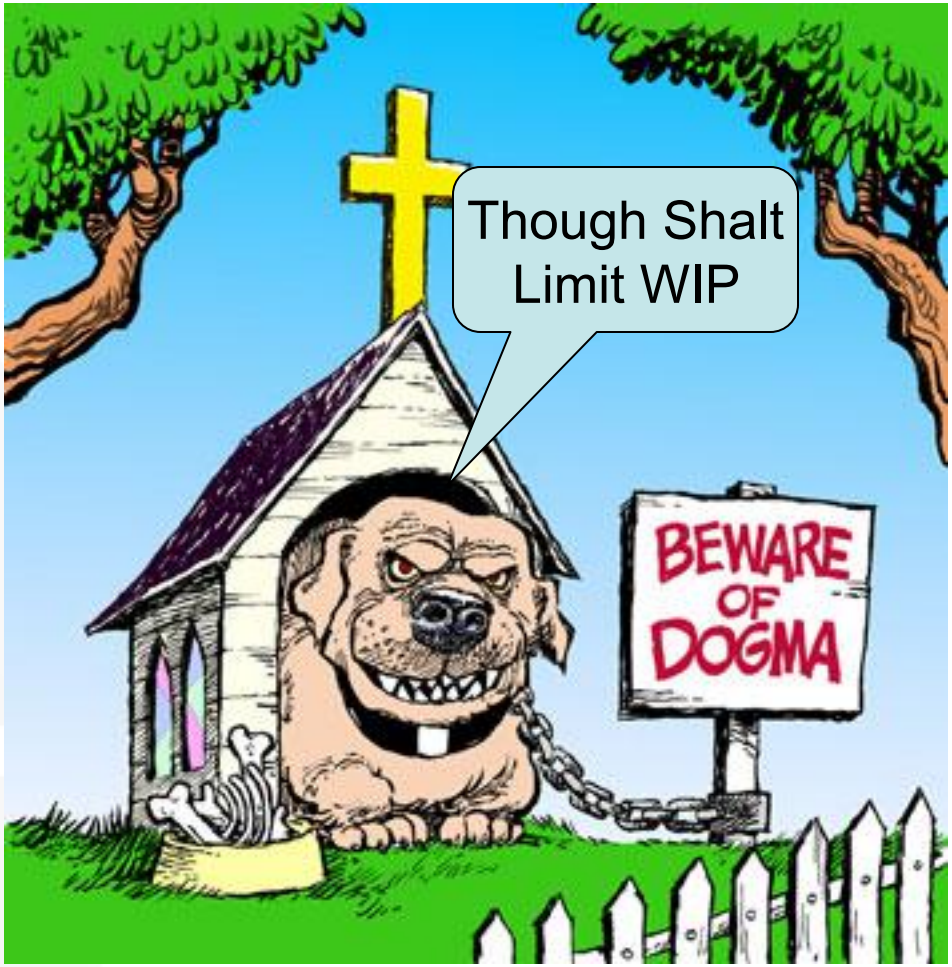
- Both are Lean and Agile
- Both based on pull scheduling
- Both limit WIP
- Both use transparency to drive process improvement
- Both focus on delivering releasable software early and often
- Both are based on self-organizing teams
- Both require breaking the work into pieces
- In both cases the release 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**.

Don’t be dogmatic

travelping

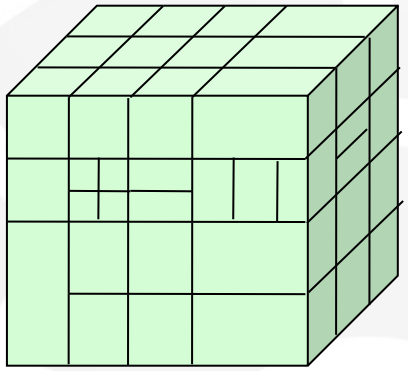


travelping

Essential skills needed

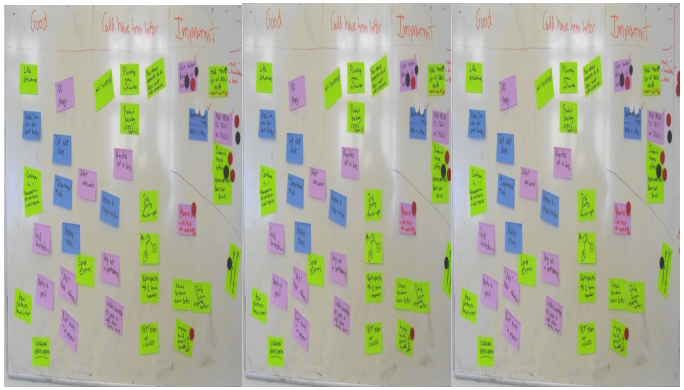
regardless of process

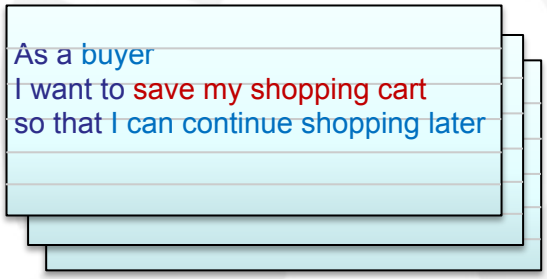
Splitting the system into useful pieces

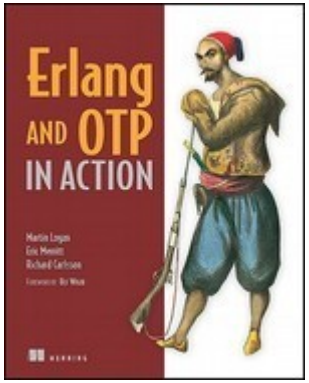


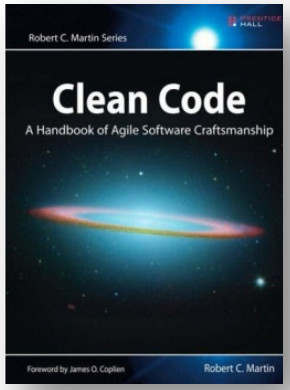
Software craftsmanship

Retrospectives









After three month . . .

travelping

- Fixed about 50 long standing Issues in 2 weeks
- Meetings are less less boring
- We know who does What, When and Why
- Developer Backlog is short and clean
- Bugs gets fixed fast – as they are visable

- Introduction without any resistence !