in real life

Agile and Lean Software Development

Mikael Svahnberg[1]

September 26, 2017

[1] Mikael.Svahnberg@bth.se

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

DevOps

Summary

Further Reading

Early Days

- Traditional Plan-driven development.
  - Long Iterations (2 year cycles)
  - Separation of Roles
    - Management
    - Requirements
    - Development
    - Testing
    - Deployment
    - Maintenance
  - Clearly demarcated process steps, tied to the Roles.
    - c.f. “Conways law”

Follow the Plan!

Agile

Agile Values

- Individuals and Interactions more than processes and tools
- Working Software more than comprehensive documentation
- Customer Collaboration more than contract negotiation
- Responding to Change more than following a plan

Discussion

- What does this mean?
- Documentation?
- Code?
- Planning?

Scrum

- One of the predominant agile frameworks.
- `PA2555 Agile and Lean Software Development`
- Not the only Agile Method
  - RAD
  - DAD
  - DSDM
  - Chrystal
  - FDD
  - …

Lean

- Often bundled with Agile, but not the same thing
    - Somewhat keyed towards the same goals, though.
- TL;DR:
    - Don’t do unnecessary stuff
    - Don’t do stuff until you know you need it
- Value Stream Mapping
- Kanban

Value Stream Mapping

- Analyse a process flow and identify *wastes*
- Originally from manufacturing industry, some translation into software engineering is needed:

Lean Manufacturing	Software Development
Inventory	W1: Partially Performed Work
Overproduction	W2: Extra Features
Extra Processing	W3: Extra Processes
Transportation	W4: Handovers
Motion	W5: Motion/Task Switching
Waiting	W6: Delays
Defects	W7: Defects

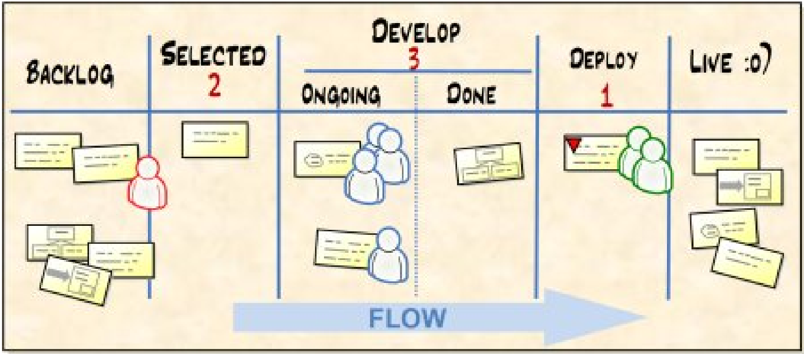
Discussion

- What would Agile do to address each of these?
- In some cases, Agile introduces contradictions within these wastes; it optimises one at the expense of another.
  - Discuss where this might occur.

Kanban

- No fixed-length iterations. Measure lead time and velocity instead.
- Less up-front planning (?)
- Rapidly shifting priorities.
- Three pillars:
    - Visualise Workflow
    - Limit Work in Progess (WIP)
    - Measure Lead Time

Kanban


BACKLOG
SELECTED
2
DEVELOP
3
ONGOING
DONE
DEPLOY
1
LIVE :o)
FLOW

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

DevOps

Summary

Further Reading

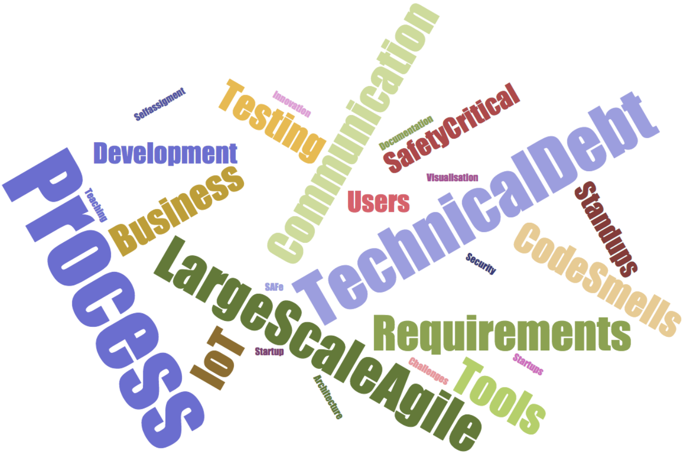
Research Trends; Perspectives

(from SEI's BAPO model for Product-Line Engineering)

- Business
- Architecture
- Processes
- Organisation

XP2017 + Workshops

(Technical Debt, IoT, Large-Scale Agile, Safety Critical)



Industry Trends

- Have left individual practices
  - e.g. Test first, pair programming, etc.
- Focus now on whole methods:
  - XP, Scrum, Lean Software Development
- From Team level / Team Performance to Organisational Understanding
- From 30 day sprints to Continuous Deployment
- From Development to Value Delivery

Research Trends: Sustainable Agile Development

- Scalable
- Test Automation
- Manage Technical Debt

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

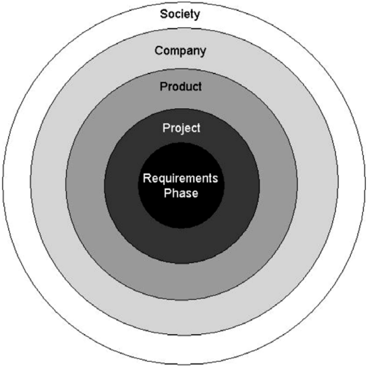
DevOps

Summary

Further Reading

Value

T. Gorschek, A.M. Davis, *Requirements Engineering; In Search of the Dependent Variables*, Information and Software Tecnology 50(2008):67–75.



(+ Process, which is not visible in this figure but neatly bisects it.)

Agile Value

- Business Value
- Lean: Reduce costs, remove waste. What remains must be of value.
  - VSM
  - JIT production
  - Double duty of artefacts (e.g. test cases as requirements)
- How to assess Value?
  - Benefit Points vs Story Points

Continuous Deployment, Continuous Experimentation

- “We have no clue what is value, let’s experiment”
- Big Data?

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

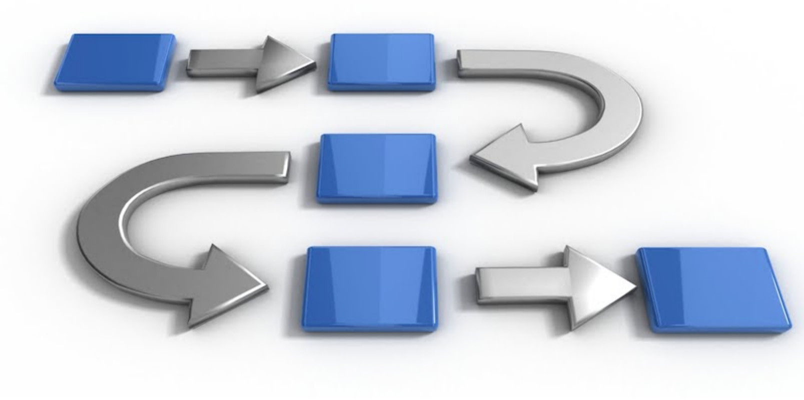
DevOps

Summary

Further Reading

Lean and Flow

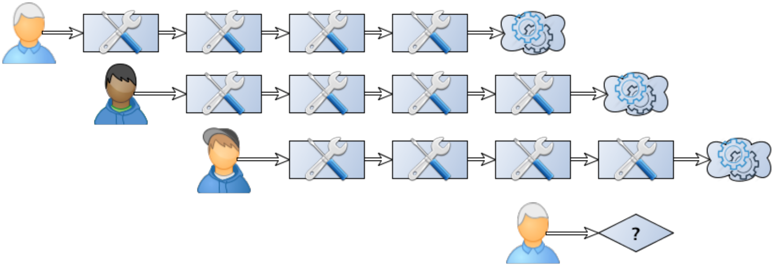
- Lean: Optimise the end-to-end *Flow*



Infrastructure Challenges

No longer a common startup phase; individuals work on separate flows.

- Ensure a *common* work environment (Develop, Test, Deploy)
- Ensure a *reproducible* work environment
- One-Click-Setup for the *next iteration*
- Related: Avoid Code Regressions.



DevOps and Continuous Integration

Solution to challenges: Automation
5S practices:

- Sort - Remove unnecessary items
- Set in order - Organise what's left, e.g. with build scripts, shortcuts, labelling, deploy to QA, etc.
- Shine - Tidy up and clean your new workspace
- Standardize - Document work processes
- Sustain - Regularly apply standards

5S: Sort

- Remove unnecessary items
- For code: Actually a really big and hairy topic.
- For workplace, regularly rollback your workplace to a steady state.

5S: Set in Order

- Repeatable Provisioning
  - Make sure your changes survive a reboot
    - No live changes!
    - Configurations under Version Control
  - Make sure all developers see the real cost of the application
    - No live changes: Fix it in repeatable configuration instead.
  - Focus on Flow - One Script away:
    - New server
    - Clean database
    - Rollback database / server / workplace
- `make`
- `make test`
- `make deploy`

5S: Shine

- Get your development and test environments as close to deployment state as possible

5S Standardise

- Document how to start a new iteration
- Document how to rollback
- Document how to deploy
- Document *rules* about who may touch production servers, and how.

5S: Sustain

- Continuous Build and Testing
  - Fosters confidence in the development team: *We know what is going on!*
  - Developers can focus on the flow of creating new features, and are confident that nothing old breaks.
  - Create regression tests
    - Initially
    - After exploratory testing
    - For each bug!
    - With Configuration Management!

Summary

A stable build and deployment environment (which can still be flexible!):

- Grows confidence that it can be changed.
- Avoids *some* risky deployments.
  - `Development env === Deployment env`
- “It works on my machine”
  - `Development env === Deployment env`
- Rules (and version control for configurations) for production servers ensures
  - Repeatability
  - Homogeneity
- Bottom line: Fix the infrastructure, focus on flow.

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

DevOps

Summary

Further Reading

DevOps Approaches

- Mix responsibilities of both dev and ops to all engineers
- Mix Teams, but keep responsibilities
- Separate “bridge team” btw. dev and ops.
- Infrastructure as Code

DevOps Values

- Emphasise communication, collaboration, and integration betwen Developers and IT operations.
    - Culture
    - Automation
    - Measurement
    - Sharing

Cultural Impacts

- Trust (handing over the keys to production servers)
- Eye-Opening: “Drink your own Champagne”
- Deployment/Integration awareness
    - Problems with deploying own software
    - Problems with deploying tool stack
- Added responsibilities
    - Build Master
    - Mixed responsibilities not only good
- Admin rights on production servers gives easy access to
    - statistics (ok)
    - debugging (:cringe:)
- Req. common ways of working

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

DevOps

Summary

Further Reading

Research Ideas

- Which Agile methods are actually used, and to what exent (Fully/partially)
  - Any particular practices?
- Kanban vs. Scrum – which is more productive?
- Global Agile – how does Agile work when half your developent team sits in Southeast Asia?
- Test Framework for DevOps “Infrastructure as Code” – develop and evaluate
- Evaluation of Frameworks for Continuous Deployment/Continuous Experimentation
- Software Architecture for Automated Tests
- Sustainable Agile: How big or how old do your product need to be before you need to add more documentation?
  - What is the right amount of documentation?

in real life

Topic

Background

Current Research Trends

Value

Lean Flow and DevOps

DevOps

Summary

Further Reading

Further Reading

- XP/agile universe conference series, DBLP link: `http://dblp.uni-trier.de/db/conf/xpu/`
- Ramtin Jabbari, Nauman bin Ali, Kai Petersen, and Binish Tanveer. 2016. What is DevOps?: A Systematic Mapping Study on Definitions and Practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016 (XP '16 Workshops)*. ACM, New York, NY, USA, , Article 12 , 11 pages. DOI: `http://dx.doi.org/10.1145/2962695.2962707`