Agile Methods

Introduction to Agile Methods
by Pietari Kettunen
Me

● Pietari Kettunen
● Software person since 2003
● TietoEnator, Solita, Swisscom and Sqooba
● Agilist since ~ 2005
● “Certified” Scrum Master, Product Owner, Scrum Practitioner, Kanban
● @Pietrotull
Sqooba

- Big Data company
- Founded 2016
- Employees > 20
- Based in Bern, branch office in Munich
Sqooba Customers
Agenda

● Traditional software engineering
  ○ Problems
● Transition to agile
● Values and principles of agile
● Agile process
  ○ Scrum
  ○ Kanban
● Technical Practises
  ○ TDD
  ○ CI/CD
Traditional S.E.

- Modelled after construction engineering
  - Architect
  - Software Architecture
  - Build tools
Not without issues...
The Problem

“Only thing that is constant is change”
- Heraclitus
Construction vs Software

low design cost & high build cost
vs
high design cost & minimal “build” cost
The Solution: agile

- Adaptation
- Feedback loops
- Defer decisions
Agile Manifesto

- February 2001

- Kent Beck
- Mike Beedle
- Arie van Bennekum
- Alistair Cockburn
- Ward Cunningham
- Martin Fowler
- James Grenning
- Jim Highsmith
- Andrew Hunt
- Ron Jeffries
- Jon Kern
- Brian Marick
- Robert C. Martin
- Steve Mellor
- Ken Schwaber
- Jeff Sutherland
- Dave Thomas
Agile Manifesto

*Individuals and interactions over processes and tools*

*Working software over comprehensive documentation*

*Customer collaboration over contract negotiation*

*Responding to change over following a plan*
12 Principles of Agile

- customer satisfaction
- embrace change
- frequent delivery
- collaboration
- motivated individuals
- face to face

- working software
- sustainable development
- technical excellence
- simplicity
- self-organizing team
- retrospection
Holy Trinity of Software

- Lean Software Development
- Agile
- Software Craftsmanship
Agile

- Technical Practises
- Process Management
Technical practices

- Pair programming
- Test driven development
- CI / CD
  - Continuous Integration
  - Continuous Deployment
- Behaviour driven development / specification by example
Process Methodologies

- Scrum
- Kanban
- XP
Process Methodologies

- Just In Time decisions
- Pull -mechanism / slack
- Visualization
- Transparency
- Splitting work into smaller pieces
- Limiting work in process
- Kaizen - continuous improvement
Scrum
Scrum
Pioneering Scrum

Scrum (early 90’s)
- Jeff Sutherland (Easel Corp)
- Ken Schwaber (Advanced Development Methods)

Jeff & Ken collaborated to present
- Scrum methodology at OOPSLA ’95
Scrum

- Agile process for producing business value
- Iterative (sprints)
- does NOT prescribe technical practices
Scrum in action

Product Backlog → Sprint Backlog → Sprint → Working increment of the software

- 24 h
- 30 days
Scrum Roles

- Product owner
- Scrum master
- Team
Product Owner

- Decides priorities / order
- Vision of the product
ScrumMaster

- Takes care of the process
- Enables the team to do their work
- Owns the impediment list
Team

- Makes the magic happen
- Cross Functional
  - includes all the skills to finish the product
- Self-organizing
Scrum Ceremonies

1. Sprint planning
2. Daily scrum meeting
3. Sprint review
4. Sprint retrospective
1. Sprint planning

Who

- Team, ScrumMaster & Product Owner

Agenda

- Discuss top priority backlog items
- Team selects which to do

Why

- Know what will be worked on
- Understand it enough to do it
2. The daily scrum

Parameters:

- Daily
- max 15 minutes
- standing

Not a problem solving meeting

- Whole world is invited
- only team, ScrumMaster, Product Owner can talk
2. The daily scrum

Questions

1. What did you do?
2. What will you do?
3. Is there anything stopping you? (impediment)
3. Sprint review

- Inspect and adapt the product
- Team presents what was accomplished
- Typically involves a demo
- Informal
4. Retrospective

- Inspect and adapt the process
- Everyone can participate
Scrum Artifacts

- Product backlog
- Sprint goal
- Sprint backlog
- Burndown chart
- Impediment List
Product backlog

- List of desired work
- Ordered / prioritized by the product owner
- Reorganized at the start of each sprint
Sprint goal

High level summary of where the focus is for given sprint

For the “high level” boss
Sprint backlog

- Evolves
- Team maintains
  - can add tasks
  - can remove tasks
  - re-estimate
- The team owns the sprint backlog
- “Best guess” what the team needs to do
- Progress visible in the task board
Burndown chart
Scaling Scrum

- Several Scrum teams
- Scrum of scrums
  - coordination over several Scrum teams
- LeSS
- SAFe
Kanban
Kanban

- Kanban = “signal card”
- Originally by Taichi Ono (Toyota)
- Software Kanban by David Anderson
- Evolutionary approach
Kanban method

1. Start with what you have
2. Agree to pursue incremental, evolutionary change
3. Respect current process, roles & titles
4. Leadership at all levels
Kanban 6 practises

1. Visualize workflow
2. Limit work in progress
3. Manage flow
4. Explicit policies
5. Implement feedback loops
6. Improve collaboratively, evolve with experiments
1. Visualize workflow

- Analyze work states
- Define work item types
- Make problems visible
2. Limit work in progress

- Prevent multitasking / context switching
- Less work in progress = less waste
3. Manage flow

- Sustainable pace
4. Explicit policies

- No secrets
- Few clear rules
5. Improve collaboratively

"To be termed scientific, a method of inquiry must be based on gathering empirical and measurable evidence subject to specific principles of reasoning”

- Isaac Newton

- Collaboration with all stakeholders
- avoid local optimizations
Everything has its own rhythm
Estimation

Lead time distribution (calendar days)

Occurrences

Lead time in days
Scrum vs Kanban

**Scrum**
- WIP per sprint
- Sprint content set
- Task size
- Cross Functional teams
- Timeboxed
- Velocity
- Fair amount of rules

**Kanban**
- WIP per stage
- No untouchable tasklist
- Task size unlimited
- Allows specialist teams
- No time limits
- Lead time
- Very few rules
Tool for the job?
Agile Documentation

- Documentation is a poor substitute for conversation
- UI mockups
- Only code is up to date
- Comprehensive test suite - living documentation
public List<int[]> getThem() {
    List<int[]> list1 = new ArrayList<int[]>();
    for(int[] x : theList) {
        if (x[0] == 4) {
            list1.add(x);
        }
    }
    return list1;
}
public List<int[]> getFlaggedCells() {
    List<int[]> flaggedCells = new ArrayList<int[]>();
    for(int[] cell : gameBoard) {
        if(cell[STATUS_VALUE] == FLAGGED) {
            flaggedCells.add(cell)
        }
    }
    return flaggedCells;
}
Test Driven Development
Created by Kent Beck

- Write failing test first
- Write code to make the test pass
- Refactor
- Rinse and repeat
Do the right thing
Do the right thing
Minimum Viable Product

- pareto principle (80/20)
- Just In Time
- Has to be viable
## Full of features (2007)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Nokia N95</th>
<th>Competitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Camera</td>
<td>5mp</td>
<td>2mp</td>
</tr>
<tr>
<td>Memory card</td>
<td>microSD up to 32GB</td>
<td>no</td>
</tr>
<tr>
<td>MMS</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>3rd party apps</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Video out</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>VoIP</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Video calls</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Instant Messaging</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
N95 vs iPhone
Common Pitfalls

- Estimates as deadlines
- Illusion of Importance
- Scrum Master Manager
- Find and Replace
- Detailed plans
- Agile sells

“predicting is very difficult, especially if it involves the future”  
- Niels Bohr
Tips

- Split to tasks
- Validation, feedback loops
- Definition of done
- Minimum viable product
- Measure progress
- Discipline
- Plan just enough
- Start implementation early (not without planning)
Sources

- https://www.scrumguides.org
- https://edu.leankanban.com/
The End...

Thank you for your attention...