Rolling out mission critical system in an agilish way

Reflections on building a large-scale dependable information system for public sector

RCoSE 2015
Session 2: “Processes”
23 May 2015, Firenze, Italy

aapo.koski@insta.fi
Context: the new Finnish ERC IS

System facts:
- 4 M calls per year
- 1,6 M dispatched tasks
- 6 ERC’s, 2 datacenters
- up to concurrent 200 operators

Development facts:
- 40-50 people, 5-6 agile teams
- ~1 mio LOC (Java+Javascript)
- ~20 3rd party systems integrated
- 6 subscontractors

SYSTEM PROVIDED AS A SERVICE
The starting point

- I want you to build this system for us…
- And we can do it together…

- However,
  - The budget is fixed…
  - And the schedule is fixed…
  - And we really cannot drop any features anymore…

- But we should execute this project in an **agile** way!
In an agile way?

• Only three things:
  – User involvement
  – Iterative and incremental development (frequent deliveries)
  – Constant adaptation to the situation at hand

Common factor: feedback
The original setup

Customer team

- Requirements

Product Owner

- User stories

Development team

- Team (tech.) lead
- Developers

Delivery every 3-4 weeks
Dark clouds in the horizon…

<table>
<thead>
<tr>
<th>Symptoms:</th>
<th>User stories do not get finished</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unsatisfying sprint demos</td>
</tr>
<tr>
<td></td>
<td>Progress not clear</td>
</tr>
<tr>
<td></td>
<td>Gut feeling of increasing TD</td>
</tr>
<tr>
<td></td>
<td>Change in working atmosphere</td>
</tr>
<tr>
<td></td>
<td>…</td>
</tr>
</tbody>
</table>
The main pain points

- Inability to adapt the development model continuously
- Too slow responses to changing needs
- Too long development cycles and time to delivery
- Estimates treated as deadlines
- Insufficient developer productivity
- Insufficient level of testing
- Unsatisfying end user experience, bad overall quality
- Insufficient visibility to project state and progress
- Weak diagnosis of the delivered SW performance
- …

OBVIOUSLY:
WE WERE NOT RAPID/AGILE ENOUGH FOR THE MARKETPLACE!
Quality is a problem – who to blame?

Process?
Developers?
Characteristics of large SW systems?
Customer?
Something else?
Lessons learned

It is not enough to do the things you do almost right or by the book.

- Everyone involved must keep raising the bar every day, in a disciplined and continuous way
Development organization – the current setup

**Strengthening the role of the PO**

**Dual-track requirement handling**

**Asking the right questions**

**Strengthening the role of the UX**
The way ahead…

Constant focus on the way we work

• Everyone’s responsibility to find ways to improve, continuously

Improvements on the testing effort

• Test automation
• Testers involved in the definition of the acceptance criteria

Enhancing developer productivity

• Wideband communication, ease of communication
• Collaboration on every level
• Feedback in a constructive way
The most important issues – no surprise…

- **Direct access to real customer**
  - Someone with the true capability to make decisions

- **Short feedback loops**
  - As short as possible, and even shorter

- **Environment to support effective collaboration**
  - Instant and easy access to up-to-date information

- **Emphasis on quality-driven development**
  - Understanding what quality means to the customer
  - Have effective means to assure the quality of the deliverables

**ENABLERS OF SUCCESS, NOT GUARANTEE!**