Does Architecture Crumble in Agile Development?
Long term impacts of agile transformation

Pekka Abrahamsson
University of Helsinki

June-8th, 2009, Software Architecture Challenges in the 21st Century workshop, USC, CA, USA
ABRAHAMSSON’S PROFILE

- Professor of computer science in University of Helsinki, Finland
  - Chief Scientist (adjunct) in SINTEF, Norway
- Project lead of FLEXI-ITEA2, 40MEUR Flexible and Agile Global Product Development (38 partners, 7 countries, 2007-2009)
- Project lead of ITEA-Achiement Award (silver) winner in 2007, AGILE-ITEA (2004-06), 22MEUR
- XP2006, XP2009 program chair, XP2008 general chair, member of steering committee
- Nokia Foundation Award recipient in 2007
- IEEE 1648 Agile Standard Working Group Chair
- IEEE SW Advisory Board Member
- 80+ scientific publications
CONCLUSIONS

- Architecture is primarily a vehicle for communication between different stakeholders. As such, it most likely will initially crumble in agile development, by necessity. Later, it is not even mentioned in the top-20 problem lists.

- "When we started the Agile, we went the opposite direction. We did not write any documents, and that proved not to be a very good thing. There is a danger that the information about the key architectural decisions is only in the heads of the people who are doing the work. Now, after three years of learning we have found a kind of way to use the documents the right way."

Quote source: Babar et al (2009), to appear, a case study, an interview with an architect
CONCLUSIONS, cont.

- Agile transformation is the process by which an organization builds its agile capabilities.
- This process takes several years and bears impact far beyond the immediate R&D settings.
- As a result, thus, companies using “agile methods” are not using agile methods *per se*! In fact, they cannot; it is impossible!
- Organizations with high agile capabilities can be called learning organizations.
- As a summary, learning to learn is a process where even the architectures are likely to crumble!

Finally, I will demonstrate that actually the chief architects and system designers are one of the most satisfied personnel groups in agile transformation.
Change is the only certainty in software design & development

The production of Technical Specifications for a 3rd Generation Mobile System based on the evolved GSM core networks.

6-month rolling average of approved CRs (excl Cat A) by Release
Benefits of implementing Agile*

![Bar charts showing benefits of Agile implementation](image)

*Michael Mah’s research (Cutter Consortium, 2007)  
Courtesy of Kari Känsälä/Nokia
PROCESSES CANNOT BE COPIED!

Industry best-practice?

We should know this?
CONCRETE TRANSFORMATION CASES
CASE PROFILE: TELECOM

Nokia Siemens Networks

#2 in wireless networks
#2 in operator services
#3 in wireline networks

1400 customers in 150 countries
75 of top-100 operators as customers
Over 1 billion people connect through our switches

MOTIVATION FOR AGILE AT NOKIA SIEMENS NETWORKS

Problems that triggered the change
- Huge bureaucracy (handover by documents)
- Poor visibility to progress (late by 1.5 years)
- No flexibility of content (fixed at E1)
- R&D driven (Product management not very active)
- Work load in peaks (specification, design, coding, test)

Source: http://www.turkuagileday.fi/media/Petri_Taavila-from_waterwall_to_agile.pdf
MOTIVATION FOR F-SECURE’S AGILE TRANSFORMATION

Product life-cycle and product realization cycle
PRINCIPLE CHALLENGES FACED

FACED CHALLENGES

- Frozen requirements changed continuously
- Heavy planning and documentation throughout the development
- Workload for change management grew exponentially
- 90% done syndrome
- Validation phase was always much longer than planned
- Culture: relaying on individuals who are high on org. rank
## FLEXI (2007-2009) PARTNERS

<table>
<thead>
<tr>
<th>Application partners</th>
<th>Technology partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spain</td>
<td>Telefonica, Indutaux, DS2, NSWARE Technologies, ESI, SQS, INDUSTRIA, UPM</td>
</tr>
<tr>
<td>Netherlands</td>
<td>Philips</td>
</tr>
<tr>
<td>Finland</td>
<td>F-Secure, Nokia, on2 Technologies, VTT, reaktor Innovations, UNIVERSITY of HELSINKI</td>
</tr>
<tr>
<td>Sweden</td>
<td>ABB</td>
</tr>
<tr>
<td>Ireland</td>
<td>Rovsing, Fidelity Investments, exoftware, Tampere University of Technology, UNIVERSITY of HELSINKI</td>
</tr>
<tr>
<td>Norway</td>
<td>confirmit, Geomatikk, Objectnet, KONGSBERG</td>
</tr>
<tr>
<td>Belgium</td>
<td>Callatay &amp; Wouters, SCIA, sirris</td>
</tr>
<tr>
<td>Israel</td>
<td>OPCAT</td>
</tr>
</tbody>
</table>

Source: [www.flexi-itea2.org](http://www.flexi-itea2.org)
POSSIBLE IMPACT ARENAS

- **Business & Customer**
  - Profitability and business case of the product
  - Developing high value features and functionality
  - Understanding customer needs and the value of a feature to customers

- **Visibility & Flexibility**
  - Visibility to actual status of development
  - Flexibility, ability to respond to changes
  - Time to get releases or features out
  - Visibility to problems in e.g. tools, practices, plans

- **Traditional**
  - Predictability of development
  - Quality of code
  - Early detection of bugs, errors and defects
  - Productivity of development teams
  - Cost of development

Source: Vilkki (2009), FLEXI-Newsletter, 02/2009, pp. 5-6, online: www.flexi-itea2.org
POSSIBLE IMPACT ARENAS

- Trust, Communication & Teamwork
  - Trust between product management and development teams
  - Team work, communication and cooperation of development teams
  - Motivation and enthusiasm of the development team members
  - Self-organization and autonomy of development teams

- Other
  - Level of technical competence
  - Sustainable pace and balanced workload

Source: Vilkki (2009), FLEXI-Newsletter, 02/2009, pp. 5-6, online: www.flexi-itea2.org
Nokia Siemens Networks: Timeline of Agile transformation in a NPM Product Programme

Source: http://www.turkuagileday.fi/media/Petri_Taavila-from_waterwall_to_agile.pdf
## SCRUM PRACTICES

<table>
<thead>
<tr>
<th>Practice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort Estimation</td>
<td>Effort needed for each functional item is iteratively estimated.</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>A complete collection of the functionality needed in the system.</td>
</tr>
<tr>
<td>Sprint</td>
<td>A 30-day period of development.</td>
</tr>
<tr>
<td>Sprint Planning Meeting</td>
<td>The goal and functionality for each Sprint is planned among team members and customer representatives.</td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>The functionality to be implemented in a Sprint.</td>
</tr>
<tr>
<td>Daily Scrum Meeting</td>
<td>The Scrum team has a 15-minute meeting every day.</td>
</tr>
<tr>
<td>Sprint Review Meeting</td>
<td>The results of the Sprint are reviewed, and further action agreed upon.</td>
</tr>
</tbody>
</table>
# SCRUM PRACTICES IN USE

<table>
<thead>
<tr>
<th>Practice</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort Estimation</td>
<td>Effort needed for each functional item is iteratively estimated.</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>Complete collection of the functionality needed in the system.</td>
</tr>
<tr>
<td>Sprint</td>
<td>Daily period of development.</td>
</tr>
<tr>
<td>Sprint Planning Meeting</td>
<td>The goal and functionality for each Sprint is planned among team members and customer representatives.</td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>Functionality to be implemented in a Sprint.</td>
</tr>
<tr>
<td>Daily Scrum Meeting</td>
<td>OK, but a daily stand-up with 20+ people does not make sense</td>
</tr>
<tr>
<td>Sprint Review Meeting</td>
<td>The results of the Sprint are reviewed, and action agreed upon.</td>
</tr>
</tbody>
</table>
SCRUM ROLES AND RESPONSIBILITIES

Management

Product Owner
Responsible for the project, and the realization of the requirements into features

Customer
Defines the set of requirements for the system (the Product Backlog list)

Development Team

Scrum Master
Ensures the project runs under Scrum principles, runs the Scrum Team

Support

Scrum Team
Project team, with authority to decide on actions

Other
**SCRUM ROLES AND RESPONSIBILITIES**

**Management**
- Management
  - In charge of final decision making

**Customer / Business Competence**
- Product Owner
  - Responsible for the project, and the realization of the requirements into features

- Customer
  - Defines the set of requirements for the system (the Product Backlog list)

**Scrum Team**
- Scrum Team
  - Project team, with authority to decide on actions

**Development Team**
- Two steering groups
- One team
- No Scrum Master
- One team
- 20-30 people

- Business involvement/OK
- Business target did was not clear in the project, however (cf. 4 sprints --> 80% of requirement

**Support**

**Other**
ROADS TO PROCESS IMPLEMENTATION

- Cross-roads: Possible
- Highway: Guaranteed
- Dead-end street: Not possible
- Country road: Possible

PROCESS PUSH (MANAGEMENT)

ROADS TO PROCESS IMPLEMENTATION

DEPLOYMENT STRATEGY

Framework from Abrahamsson (2006)
DEPLOYMENT STRATEGY

Framework from Abrahamsson (2006)
WHICH STRATEGY WORKED THE BEST?

- Regardless of the chosen strategy, each company was able institutionalize an agile process in place.
- F-Secure, Finland
  - All R&D in agile mode (took 2½ yrs)
- Nokia Siemens Networks, Finland
  - Agile as the operation mode de-facto, 2 yrs
- Nokia, Finland
  - Agile as the operation mode de-facto, 2 yrs
- Philips, Netherlands
  - 16 projects use agile & iterative processes (2½ yrs)
Agile transformation in NSN

• Each Business Unit or Product (Line) decides if and when to do the transformation
  – No enforcement, let the teams choose what practices to use!

• Centralized support for Agile and Lean development

Source: Vilkki (2008), XP2008 Keynote,
RESEARCH METHOD FOR NOKIA SIEMENS NETWORKS’ SURVEY

- Industrial, descriptive survey distributed to 2450 staff members involved in agile transformation across several continents
- 658 valid responses collected, response rate of 27%
- Question themes
  - Length of experience in agile
  - Methods/practices in use
  - Satisfaction (on your own work)
  - Impacts in 18 pre-defined areas
  - Free comments in all of the above sections

Source: Vilkki (2009), FLEXI-Newsletter, 02/2009, pp. 5-6, online: www.flexi-itea2.org
PRACTICES IN USE

- Sustainable pace
- ATDD
- Collective code ownership
- TDD (Test Driven development)
- Refactoring
- Pair-programming
- Planning poker
- Tests written at the same time as code
- Self-organized, co-located, cross-functional teams
- User stories
- Continuous Integration (min. daily build)
- Retrospectives
- Short (max. 4 weeks) time-boxed iterations
- Scrum
- Product backlog

Source: Vilkki (2009), FLEXI-Newsletter, 02/2009, pp. 5-6, online: www.flexi-itea2.org
AGILE ADOPTION PATTERNS DISTILLED

By usage of agile practices the respondents can be divided into three groups:

- **Basic**: 42% reported using at least the NSN basic agile practices (Short time-boxed iterations, Product backlog, Continuous Integration (at least daily build), Self-organized co-located, cross-functions teams and Retrospectives)

- **Intermediate**: 23% reported using the basic practices and at least the key engineering practices (Refactoring and Test written at the same time as code or TDD or ATDD)

- **Fully**: 7% reported using so many practices that they can be defined fully agile (at least basic practices, key engineering practices and sustainable pace)

Source: Vilkki (2009), FLEXI-Newsletter, 02/2009, pp. 5-6, online: www.flexi-itea2.org
Case NPM Programme: Current standing (original problem)

- greater visibility to progress (poor visibility)
- reduced bureaucracy (huge bureaucracy)
- product management is participating actively (R&D driven)
- welcomes changes to the content (fixed content)
- work load evenly shared along the whole program (peaks in workload)

**Bonus**

- increased job satisfaction
- delivery capability almost every day

Source: http://www.turkuagileday.fi/media/Petri_Taavila-from_waterwall_to_agile.pdf
Case NPM Programme: Some Success

Rationale

- Management commitment
- Key people commitment (Program manager, SW-project manager, test PM & Product Owner)
- Reserve enough time
  - Minimum 2 years
  - Introduce new practices gradually
- Use facilitators to introduce new practices
- Have a problem to be solved
- Train people
  - Whole organization
- Tolerate the resistance, be persistence
  - First sprint(s) will fail <- moment of truth
- Have good engineering practices in place
- Lean thinking (challenge everything)

Source: http://www.turkuagileday.fi/media/Petri_Taavila-from_waterwall_to_agile.pdf
Case NMP Programme: Pain points

- Role of the solid line managers
  - There is no role in scrum!
- Self organizations scrum teams
  - People not used to take responsible
- New practices (Reviews, planning sessions, etc)
  - Facilitators needed, some agile experiences needed
- People are feeling that we are cheating by not doing so much
- Loud resistance by 10% of people
  - One by one “missionary work”
- Exposes the organization problems
- Exposes the individual performance
- Product Management role as a product owner
- Role of Program Contract = commitment in the early phase
- Management expectations
- Subcontracting

Source: http://www.turkuagileday.fi/media/Petri_Taavila-from_waterwall_to_agile.pdf
CONCLUSIONS

- Architecture is primarily a vehicle for communication between different stakeholders. As such, it most likely will initially crumble in agile development, by necessity. Later, it is not even mentioned in the top-20 problem lists.

- “When we started the Agile, we went the opposite direction. We did not write any documents, and that proved not to be a very good thing. There is a danger that the information about the key architectural decisions is only in the heads of the people who are doing the work. Now, after three years of learning we have found a kind of way to use the documents the right way.”

Quote source: Babar et al (2009), to appear, a case study, an interview with an architect
SOFTWARE PRODUCT LINE & AGILE DEVELOPMENT

Product line platform

- Mobile Windows Client
- Symbian Client
- Back End System

- Architectural overview model
- Architectural documents of sub systems
- Layered architecture
- Model-View-Controller architecture
- Interfaces between clients and the back end system
- Interfaces with external systems

Code conventions

- Common components
- Honeycomb-specific components

Source: Babar et al (2009), to appear
THANK YOU!

Questions & Comments

Contact me at:
pekka.abrahamsson@ieee.org
Tel. +358-405-415-929