

Workshop: Architecture in an Agile World

Dennis Mancl

Alcatel-Lucent
Murray Hill, NJ, USA
dennis.mancl@alcatel-lucent.com

Steven D. Fraser

Cisco Systems
San Jose, CA, USA
sdfraser@acm.org

Bill Opdyke

JP Morgan Chase
Chicago, IL, USA
opdyke@acm.org

Abstract

Agility is important in the business world – but in many problem domains, *architecture* is valuable too. The combination of agile and architecture-driven approaches is often essential to success – it creates some opportunities for discovering potential problems early in the development cycle.

Categories and Subject Descriptors: D.2.9 Management – programming teams; K.4.3 Organizational Impacts – computer-supported collaborative work; K.6.3 Software Management – software process

General Terms: Management

Keywords: agile; architecture

1. Agility and Architecture

Agility is important in the business world – but in many problem domains, *architecture* is valuable too. The combination of agile and architecture-driven approaches is often essential to success – it creates some opportunities for discovering potential problems early in the development cycle.

For example, Agile development techniques are best for highlighting issues that are linked to the “uncertainties in customer requirements.” Architecture practices can also help surface some of the key “technical unknowns” in a complex product development effort.

The Architecture in an Agile World workshop is exploring the issues and obstacles in doing agile development – with an eye on building and maintaining a sound architecture. The discussion in this workshop attacks the following subjects:

- Understanding some of the risks and opportunities of blending agile and architecture-driven software development practices.

- Determining how architects establish credibility with software development teams.
- Definitions of some key characteristics of good architects and good agilists.
- Discussion of why agile groups have problems with architecture-driven processes and why architects have problems with agile project organizations.
- How both agile and architecture-driven practices are used in cloud computing and software as a service.
- Exploration of some of the “good practices” that should be part of the toolkit of agilists and architects – “how to get just the right amount of architecture.”

2. Why Agility and Architecture?

The world is moving faster than ever, and our software development techniques are struggling to keep up. We feel we need to have an agile feature set, but without a well-defined and understandable architecture, we feel like everything is in chaos. How do we manage the balance between architecture and agility?

This workshop explores the clash between agile and architecture-centric philosophies, and some practical ideas for combining the two approaches. This workshop continues a discussion started in an OOPSLA 2009 workshop and panel session.

The report of the OOPSLA 2009 workshop can be found at <http://mysite.verizon.net/dennis.mancl/oopsla09/index.html>.

3. What do we know about Agility and Architecture?

The obvious conclusion in the OOPSLA 2009 workshop was that “pure agile” and “pure architecture-driven” approaches are not the way to go.

- In a “pure agile” approach, the system would have an “emergent architecture” – because the most extreme agilists would consider any serious up-front architecture planning and modeling as unnecessary big design up front.
- In a “pure architecture-driven” approach, all architecture choices should be made using a top-down devel-

opment approach instead of using code-level experimentation and refactoring.

Some early attention to architecture planning will contribute to better communication within a development team. Even in a one-person project, architecture planning helps – it gives the developer an opportunity to think through important development issues, instead of merely implementing his or her first idea.

4. The architect’s role in an Agile world

In an agile world, an architect (or any development team member who is involved with software architecture or high-level design) has to be connected with development teams.

An architect needs to overcome the perception of being an “outsider” to the development work. One agile approach to reduce the distance between architects and developers is to include architects as members of a development team, with specific development and testing tasks assigned to them in each development iteration. There is at least one big benefit of this model: it makes the architects into users of their own documents and models, so they aren’t just off in an “ivory tower.” Also, as a member of an agile team, they will have frequent discussions with other development team members, which should improve communication about the architecture.

On the other hand, the idea of “making an architect a member of a development team” is not perfect. Some architects have poor development skills. The distractions of the development process may limit the amount of time they can spend on architecture planning. In some large systems, it is necessary for architects to spend time thinking about larger issues – not just the development tasks of a single development team.

5. Building bridges between architects and agilists

Agile development teams have many complaints about architects. The top two problems:

- Agilists feel that architects are “always trying to push everyone around.”
- In addition, agilists feel that architects often “overdesign.”

These are valid complaints. Architects really do try to control things too much – and an agile viewpoint could really help architects do their job better.

Architects often complain about agile teams – some issues related to lack of forward vision and feedback:

- Most agile teams have trouble answering the question: “How much road do you have to pave ahead of you?” If the design vision only extends to the end of the next iteration, it might be easy for the design to fall off a cliff.
- Sometimes agile teams fail to get adequate feedback from stakeholders and customers. It isn’t always the agile team’s fault – but teams should always try to do better.

6. Which issues are we going to discuss?

The SPLASH 2010 workshop will have a similar discussion agenda, but it will inevitably find some different issues and new conclusions – which will be reported in the workshop poster and final report. Please check the workshop website after the conference for the results:

<http://mysite.verizon.net/dennis.mancl/splash10/index.html>