Agile Architecture Masterclass

Presented by Mishkin Berteig
with Pedram Maleknejad and Forbes Benning
About Us

- Mishkin Berteig
- Pedram Maleknejad
- Forbes Benning
Review of Agility
And Architecture
Conway’s Law

• "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations." - Melvin Conway (1967)
Agility and Business

- Volatility
- Uncertainty
- Complexity
- Ambiguity

- The “horizon of predictability” is extremely short
# Agility: Scrum and Kanban

<table>
<thead>
<tr>
<th>Scrum</th>
<th>Kanban</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous improvement focus</td>
<td>Continuous improvement focus</td>
</tr>
<tr>
<td>2-level cadence</td>
<td>Multi-level cadence</td>
</tr>
<tr>
<td>Synchronized planning+delivery</td>
<td>De-coupled planning+delivery</td>
</tr>
<tr>
<td>Disruptive change</td>
<td>Evolutionary change</td>
</tr>
<tr>
<td>Improve product development</td>
<td>Improve service delivery</td>
</tr>
<tr>
<td>Focus on people and interactions</td>
<td>Focus on systems thinking</td>
</tr>
<tr>
<td>Time-boxed work</td>
<td>Work-in-progress limits</td>
</tr>
<tr>
<td>Start with the teams</td>
<td>Start with management</td>
</tr>
<tr>
<td>Self-organization</td>
<td>Acts of leadership</td>
</tr>
<tr>
<td>Out-of-the-box roles, artifacts + events</td>
<td>Practices introduced contextually</td>
</tr>
</tbody>
</table>

*BERTEIG*
Agility and Architecture

- From the Agile Manifesto for Software Development:

  “Continuous attention to technical excellence and good design enhances agility”

  “Simplicity, the art of maximizing the amount of work not done, is essential”

  “The best architectures, requirements and designs emerge out of self-organizing teams”

http://www.agilemanifesto.org
Technical Agility

- **Interaction Practices**
  - Pairing / Mobbing
  - User Stories
  - Specification by Example

- **Quality Practices**
  - Refactoring
  - Test Driven Development
  - Continuous Integration

- **Infrastructure Practices**
  - No Branching
  - Feature Switches
  - Continuous Delivery

- **Design Practices**
  - Agile Modelling
  - Spikes
  - YAGNI / Simplicity
False Metaphor

- Software development is **NOT** like building construction
- The metaphor has many problems:
  - Different cost of change
  - Different incremental cost of duplication
  - (Lack of) success of building construction
  - Misunderstood role of the architect
  - Complexity vs. complicated (Cynefin)
Case Studies
The Origin Story

- Charles Schwab
- 2000-2004
- 100s of applications, dozens of server technologies
  - Incremental delivery
  - Scrum
The Story of Rework

- Capital One
- 2005
- Data warehouse with 20k+ tables
- Embrace rework
- Lean-Agile
The Story of Delay

- Siemens / Trango
- 2006
- Full product re-write
- Just start
- Scrum
The Story of Urgency

- Equitable Life
- 2010
- Awful time-to-market
- Permission
- OpenAgile
Scrum Elevator Pitch

- Ken Schwaber
- “Not what I need”
- Escalation of commitment
- Fast feedback
Workshop
Question Zero: Current Architecture System

What is the current organizational system for architectural responsibilities and decision-making?

- Processes
- Artifacts
- Roles
- Policies
- Customers / stakeholders
Question One: Problems/Opportunities

Where is the current system strained? What are some of the complaints from your customers? What problems or opportunities for improvement do you see internally?
Question Two: Architecture in Transition

What aspects of your enterprise architecture responsibilities and decision-making can be shifted to:

- Architectural Runway
- Architectural Community of Practice
- Architecture Support Group
- JIT Cross-Training
- Embedded Architect
- Fully-emergent from self-organizing teams
Question Three: Change Experiments

Design three simple change experiments to improve architectural agility. For each experiment, include:

- Problem/opportunity
- Proposed experiment
- Hypothesis
- Testing/measurement of results
- Risks and permission
Q&A / Panel

www.berteig.com
www.agileadvice.com