SAFe Architectural Runway

Al Shalloway
CEO, Net Objectives

Al Shalloway
CEO, Founder

alshall@NetObjectives.com
@AlShalloway
a day of SAFe

what to watch if interested in SAFe

- Executive Overview
- Achieving Scale
- Architectural Runway
- Kanban
- Q&A

what to watch if not interested in SAFe

- Executive Overview
- Achieving Scale
- Architectural Runway
- Kanban
**Desired Outcomes**

- A Vision – context for local decisions
- Enable **new implementations** in the system
- Enable **extensions** to the system
  - Functional
  - Performance / scalable
  - Security
- Foster **testability**
- De-couple systems from the application
- Facilitate **reuse** of common functions across teams
- Provide an **example of standards**

all with minimal risk

---

**Principles of Agile Architecture**

1. **Design emerges**
2. Bigger systems require longer runways
3. Quality high, complexity minimized
4. Build & test together
5. Holistic innovation
6. Implement architectural flow
Emergent Design

Build to what you know
Maintain high quality
Learn as you go
Mitigate risk

Question to Ask

When working on a mature system consider when adding a new function...
which task area takes more time
  – writing the new function or
  – integrating it into the system?

- Why?
- Suggest due to
  – Redundancy
  – Lack of encapsulation
**architecture**

what is architecture?

- Structure
- Layers
- Underlying technology

What can’t be changed?

**architecture**

what if...

integrating functionality wasn’t hard?

we weren’t tied to our system’s architecture?

we were truly object-oriented?
Not trying to anticipate change
Cannot prevent change
Designing to accommodate change requires
  • Layered architectures
  • Code clarity and maintainability

architecture
How do we allow for evolutionary architecture?
How do we make system architecture decisions in an Agile environment?

there is a parallel between agile architecture and agile discovery
The day I lost an (imaginary) argument with “Uncle” Bob Martin.

Framework Needed

Several UIs
Several different business logics
Brain overload

Lessons learned:
• Solve smaller problems
• While preparing for what’s coming
## Types of Refactoring

<table>
<thead>
<tr>
<th>Refactoring Bad Code</th>
<th>Refactoring Good Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code &quot;smells&quot;</td>
<td>Code is &quot;tight&quot;</td>
</tr>
<tr>
<td>Improve Design without changing Function.</td>
<td>A new Requirement means code needs to be changed</td>
</tr>
<tr>
<td>Refactor to improve code quality</td>
<td>Design needs to change to accommodate this.</td>
</tr>
<tr>
<td>A way to clean up code without fear of breaking the system</td>
<td>A way to make this change without fear of breaking the system</td>
</tr>
</tbody>
</table>

The day I “kidnapped” Martin Fowler
When your design is insufficient, **refactor it** before adding new functionality

Emergent design alone is insufficient at scale

Intentional architecture is necessary
Create big picture for local decisions
Factors outside of Agile Release Trains drive intentional architecture
- Usability, extensibility, performance
- Cross-cutting patterns
Intentional architecture

Driven by event horizon
- Close in work building now
- Prepare for next PSI
- Conceptualize anything further out

Similar to decomposition of functional stories

Collaboration across the value stream

Program Portfolio Management
- Direct trains to work on most important items

Enterprise Architect
- Ensure proper architecture is in place to allow for this

Product Management
- Sequence work to improve value delivery & lower risk

All collaborate together

Agile Teams
System Architect
System Team
When decisions arise at the portfolio

Enterprise creates context for programs

When decisions arise at program

Communicated to enterprise for future roll-out to other programs
Responsibilities of Enterprise Architect

- Initiatives & strategy of enterprise architecture
- Drive architectural epics (and manage WIP)
- Synchronizes production infrastructure & NFRs across solutions whenever applicable

Principles of Agile Architecture

1. Design emerges
2. **Bigger systems require longer runways**
3. Quality high, complexity minimized
4. Build & test together
5. Holistic innovation
6. Implement architectural flow
Use Spikes To Prepare Stories

- Spikes investigate unknowns
- Spikes are estimated, time-boxed and use up velocity
- Architect provides guidance

Other Methods

Mocks
Defer Decisions
Principles of Agile Architecture

1. Design emerges
2. Bigger systems require longer runways
3. **Quality high, complexity minimized**
4. Build & test together
5. Holistic innovation
6. Implement architectural flow

Design Simplicity

Best code quality for today, none for tomorrow
High code quality means easier extensibility

Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

*Principle 2 of Agile Manifesto*
Best Practices

Follow mandates of design patterns
- Separation of concerns
- Delegation
- Encapsulation
- No redundancy
Test-First

Principles of Agile Architecture

1. Design emerges
2. Bigger systems require longer runways
3. Quality high, complexity minimized
4. **Build & test together**
5. Holistic innovation
6. Implement architectural flow
Fundamental Agile test philosophy

Those who build it shall test it.

Test-first.

Automated testing, continuous integration.

Teams must comply with Release Train, enterprise-wide conventions

Attending to testability is design

High testability results in high code quality
  • Follows mandates from design patterns

Testability requires separation of:
  • UI from business logic
  • System architecture from application architecture
  • Use from creation
Principles of Agile Architecture

1. Design emerges
2. Bigger systems require longer runways
3. Quality high, complexity minimized
4. Build & test together

5. Holistic innovation
6. Implement architectural flow

Innovate at all levels

- Teams must have time to keep quality high
- Hardening, Innovation, Planning (HIP) sprints are one opportunity
- We write quality code all of the time

“Inertia is the residue of past innovation efforts. Left unmanaged it consumes the resources required to fund next-generation innovation.”

Geoffrey Moore
Enterprise Scale Initiatives

Architectural epics represent cross-cutting initiatives and are peers to business epics (Enterprise architect is owner)
May span multiple Agile Release Trains (ARTs) and require coordination
Enterprise Architect leads the architect team that drives the implementation of epics in ARTs
Effective flow of architectural work is achieved via Epic Kanban

Principles of Agile Architecture

1. Design emerges
2. Bigger systems require longer runways
3. Quality high, complexity minimized
4. Build & test together
5. Holistic innovation

6. Implement architectural flow
Architectural Epics

Peers to business epics
Enable sustainability of business value
Cross-cutting: products, systems, solutions

Arise from:
• New opportunities
• Changing technologies
• Improving existing systems
• Reduction of duplication

Examples:
• Frameworks
• Cross-organization systems
• Replacing systems

Architectural Epics Managed
Similarly, but Separately

<table>
<thead>
<tr>
<th>Business Epics</th>
<th>Architectural Epic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portfolio Kanban</td>
<td>Portfolio Kanban</td>
</tr>
<tr>
<td>Product Manager</td>
<td>Enterprise Architect</td>
</tr>
<tr>
<td>Deliver in Stages</td>
<td>Run across PSIs as needed</td>
</tr>
<tr>
<td>Holistic View</td>
<td>Across Enterprise</td>
</tr>
<tr>
<td>Runs across trains as needed</td>
<td>Cross-train impact</td>
</tr>
</tbody>
</table>
Use Spikes To Prepare Stories

- Spikes investigate unknowns
- Spikes are estimated, time-boxed and use up velocity
- Architect provides guidance

Architectural Runway
A pragmatic view to Agile architecture
Risk driven
Quality driven
Avoids waste of too much & not enough
Thank You

Al Shalloway
email: alshall@netobjectives.com
Twitter tag @alshalloway

Register at www.netobjectives.com/register to receive notices of monthly webinars
See www.netobjectives.com/resources

See upcoming SPC courses in Seattle (1/14-17) and Atlanta (2/18-21)
www.netobjectives.com/events
Questions

email: alshall@NetObjectives.com
twitter: @alshalloway

Public Course
11 Dec **Essence of Agile: From Team to Full Stream**

SAFe Courses
14 Jan **SAFe Program Consultant Certification (Seattle)**
18 Feb **SAFe Program Consultant Certification (Atlanta)**