Continuous Architecture and Emergent Design

Scott W. Ambler
Senior Consulting Partner

scott@scottambler.com

@scottwambler
We’re going to cover a lot of ground
Agenda

1. Myths and misunderstandings
2. Ideas from Disciplined Agile Delivery (DAD)
3. Architecture and design strategies
4. Initial architecture
5. Emergence
6. Stuff happens
7. Parting thoughts
Myths and Misunderstandings
Common Myths Surrounding Agile Architecture

1. Agilists don’t “do architecture”
2. Agilists start coding right away
3. Agilists don’t follow enterprise architecture strategies
4. Agilists don’t model
5. Agilists don’t document
Realities of Agile Architecture

1. Architecture is so important agilists address it throughout the entire lifecycle
2. Disciplined agilists invest in some up front envisioning
3. Discipline agilists work closely with enterprise professionals
4. Disciplined agilists explicitly model in a lightweight manner throughout the lifecycle
5. Disciplined agilists write deliverable documentation continuously
DAD Role: Architecture Owner

• Guides the creation and evolution of the solution’s architecture
• Mentors and coaches team members in architecture practices and issues
• Understands the architectural direction and standards of your organization and ensures that the team adheres to them
• Ensures the system will be easy to support by encouraging appropriate design and refactoring
• Ensures that the system is integrated and tested frequently
• Has the final decision regarding technical decisions, but doesn’t dictate them
• Leads the initial architecture envisioning effort
DAD Teams Are Enterprise Aware

- DAD teams strive to leverage and enhance the existing organizational eco system wherever possible

- Implications for architecture and design:
  - Work with enterprise architects
  - Follow existing roadmap(s) where appropriate
  - Leverage existing assets
  - Enhance existing assets

Community Awareness
“How can I give back to my community?”

Enterprise Awareness
“How can I help my organization?”

Team Awareness
“How can I help the team?”

Individual Awareness
“How can I be the best me?”

Copyright 2013 Disciplined Agile Consortium
A High Level Lifecycle
Scrum Construction Lifecycle

A good start…
A Scrum Delivery Lifecycle

...but this is how agile teams actually work...
Unbranded Agile Delivery Lifecycle

...and it’s time to abandon the branding.
Governed Delivery Lifecycle

Disciplined agile teams are guided by senior management…
…and realize they work in an organizational ecosystem.
Disciplined Agile Delivery: Lean Lifecycle

DAD doesn’t prescribe a single lifecycle…
The Phases Disappear Over Time

First release: Inception Construction Transition

Second release: I Construction T

Third release: I Construction T

Nth+ releases: C C C C

…and promotes continuous learning and improvement.
Disciplined Agile Delivery: Lean Continuous Delivery Lifecycle

New Features

Business Value

Fixed Delivery Date

Expedite

Intangible Options

Replenishment modeling session

Work items are pulled when capacity is available to address them

Daily work

Release solution

Operate and support solution in production

Retrospective

Learnings

Strategy

Coordination Meeting

Feedback

Demo

Enhancement Requests and Defect Reports

Construction

Continuous stream of development

Sufficient functionality

Production ready

Delighted stakeholders

A good end goal
DAD is Goal-Driven, Not Prescriptive

<table>
<thead>
<tr>
<th>Goals for the Inception Phase</th>
<th>Goals for Construction Phase Iterations</th>
<th>Goals for the Transition Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Form initial team</td>
<td>- Produce a potentially consumable solution</td>
<td>Ensure the solution is consumable</td>
</tr>
<tr>
<td>- Develop common project vision</td>
<td>- Address changing stakeholder needs</td>
<td>- Deploy the solution</td>
</tr>
<tr>
<td>- Align with enterprise direction</td>
<td>- Move closer to deployable release</td>
<td></td>
</tr>
<tr>
<td>- Explore initial scope</td>
<td>- Improve quality</td>
<td></td>
</tr>
<tr>
<td>- Identify initial technical strategy</td>
<td>- Prove architecture early</td>
<td></td>
</tr>
<tr>
<td>- Develop initial release plan</td>
<td>- Form work environment</td>
<td></td>
</tr>
<tr>
<td>- Secure funding</td>
<td>- Identify risks</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ongoing Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Fulfill the project mission</td>
</tr>
<tr>
<td>- Grow team members</td>
</tr>
<tr>
<td>- Address risk</td>
</tr>
</tbody>
</table>

- Improve team process and environment
- Leverage and enhance existing infrastructure
- Coordinate activities

© Scott Ambler + Associates
Disciplined Agilists Take a Goal Driven Approach

- **Goal**

- **Issue**

- **Option**
  - Default Option

- **Advantages**
- **Disadvantages**
- **Considerations**

**Explore the Initial Scope**
- Form the Initial Team
- Address Changing Stakeholder Needs

**Source**
- Team size
- Team structure
- Team members
- Geographic distribution
- Supporting the team
- Availability

**Co-located**
- Partially dispersed
- Fully dispersed
- Distributed subteams
General Solution Architecture Principles

Architecture owners should understand why

Look beyond technology

Adopt a multi-view approach

Architecture is driven by requirements

Architectures should be loosely coupled

Architectural elements should be highly cohesive
Principles of Agile Solution Architecture

Think about the future, but wait to act

Learn early, learn fast

Simplicity first

Build quality in

Model in an agile manner

Be enterprise aware
Survey Says: Initial Architecture Efforts

- 97% performed some initial modeling OR worked from an existing legacy architecture
- 92% performed some form of up-front architecture modeling
- 77% performed high-level initial architecture envisioning
- 28% performed detailed initial architecture modeling
- 25% use enterprise or industry architecture models as references

Source: SA+A 2013 Agile Project Initiation Survey
Ambysoft.com/surveys/
Goal: Identify Initial Technical Strategy

- **Level of Detail**
  - High-level overview
  - Detailed interface
  - Detailed end-to-end
  - None

- **View Types**
  - Technology
  - Business Architecture
  - User interface (UI)

- **Modeling Strategy**
  - Informal modeling sessions
  - Formal modeling sessions
  - Single candidate architecture
  - Multiple candidate architectures

- **Delivery Strategy**
  - Extend existing solutions
    - Configure a COTS package
    - Extend a COTS package
    - Build from scratch
Goal: Align With Enterprise Direction

- **High level** Detailed Optional
  - Enforced None
- **Minimal** Comprehensive None
- **Collaborative Continuous**
  - Gated Formal None
- **Managed – Collaborative**
  - Managed – Formal Ad hoc None
- **Agile**
  - Traditional None

- Adopt Common Guidelines
- Adopt Common Templates
- Coordinate With Enterprise Teams
- Reuse Existing Infrastructure
- Adopt Governance Strategy
Non-functional requirements are critical drivers of your architecture
You’ll Need Several Views

Some Architecture Frameworks:
Zachman
TOGAF
DoDAF
4+1
Continuous Architecture Practices

Architecture owner facilitates architectural decisions throughout Construction

Architectural vision guides development efforts

Initial architectural envisioning

Architecture spikes to explore a technical issue

Architecture handbook and models updated as required

Reduce risk early by proving the architecture works
Emergent Design Practices

Discuss **design implications** during iteration planning/modeling.

Look-ahead **modeling** for upcoming complex work items.

Consider **design issues** of incoming requests from production.

Test-Driven Design (TDD) throughout Construction.

Iteration planning session to select work items and identify work tasks for current iteration.

Initial Architectural Vision

Initial modeling, planning, and organization

Initial requirements and release plan

Initial work items

Highest-priority work items

Iteration backlog

Iteration tasks

Consumable solution

Iteration review & retrospective: Demo to stakeholders, determine strategy for next iteration, and learn from your experiences

Release solution into production

Daily work

Daily coordination meeting

Funding & feedback

Inception

One or more short iterations

Stakeholder agreement

Proven architecture

Construction

Many short iterations producing a potentially consumable solution each iteration

Project viability (several)

Sufficient functionality

Transition

One or more short iterations

Production ready

Delighted stakeholders

© Scott Ambler + Associates  www.synerzip.com
Agile Model Driven Development (AMDD): Project Level

- Identify the high-level scope
- Identify initial “requirements stack”
- Identify an architectural vision

- Modeling is part of iteration planning effort
- Need to model enough to give good estimates
- Need to plan the work for the iteration

- Work through specific issues on a JIT manner
- Stakeholders actively participate
- Requirements evolve throughout project
- Model just enough for now, you can always come back later

- Develop working software via a test-first approach
- Details captured in the form of executable specifications
Goal: Prove Architecture Early

- **End-to-end working skeleton**
  - Architectural spikes
  - Solution bake-off
  - Pilot testing
  - **Stakeholder demos**
    - Informal reviews
    - Formal reviews

- **Validation**
  - Informal reviews
  - Formal reviews

- **Verification**
Goal: Produce a Potentially Consumable Solution
Goal: Leverage and Enhance the Existing Infrastructure

- **Guidelines**
  - Adopt industry guidance
  - Evolve enterprise guidance
  - Develop new guidance

- **Systems**
  - Use existing assets
  - Adopt external assets
  - Refactor existing assets
  - Develop reusable assets

- **Data**
  - Database consolidation
  - Database refactoring
  - Use existing data sources

- **Process**
  - Tailor existing process
  - Adopt external process assets
  - Share process learnings
  - Evolve process assets
  - Adopt existing templates
Stuff Happens...
What if…

You don’t think all the issues through from the very beginning?

Another team needs to update the solution years from now?

The team leaves?

There is a major requirements change that forces you to rethink the architecture in the middle of construction?
Architecture and design are so important on agile teams that we address them every day.

Think but wait to act.

The best architectures and designs emerge over time.

Collaborative approaches work best.
Thank You!

scott [at] scottambler.com
@scottwambler

AgileModeling.com
AgileData.org
Ambysoft.com
DisciplinedAgileConsortium.org
DisciplinedAgileDelivery.com
ScottAmbler.com

Disciplined Agile Delivery
Disciplined Agile Delivery
Recommended Resources
Questions?

SYNERZIP
Agile Software Product Development Partner

www.synerzip.com
Hemant Elhence
hemant@synerzip.com
469.322.0349
Synerzip in a Nut-shell

• Software product development partner for small/mid-sized technology companies
  • Exclusive focus on small/mid-sized technology companies, typically venture-backed companies in growth phase
  • By definition, all Synerzip work is the IP of its respective clients
  • Deep experience in full SDLC – design, dev, QA/testing, deployment
• Dedicated team of high caliber software professionals for each client
  • Seamlessly extends client’s local team, offering full transparency
  • Stable teams with very low turn-over
  • NOT just “staff augmentation”, but provide full mgmt support
• Actually reduces risk of development/delivery
  • Experienced team - uses appropriate level of engineering discipline
  • Practices Agile development – responsive, yet disciplined
• Reduces cost – dual-shore team, 50% cost advantage
• Offers long term flexibility – allows (facilitates) taking offshore team captive – aka “BOT” option

Synerzip in a Nut-shell

www.synerzip.com
Our Clients

www.synerzip.com
Thanks!

Call Us for a Free Consultation!

SYNERZIP
Agile Software Product Development Partner

www.synerzip.com

Hemant Elhence
hemant@synerzip.com
469.322.0349
Backup Slides
Disciplined Agile Delivery (DAD)

Disciplined Agile Delivery (DAD) is a process decision framework

The key characteristics of DAD:

– People-first
– Goal-driven
– Hybrid agile
– Learning-oriented
– Full delivery lifecycle
– Solution focused
– Risk-value lifecycle
– Enterprise aware
Goal Diagram Notation

Goal — Goal

Issue — Issue

Option — Option

Default Option

Indicates a preference for the options towards the top