An Overview of Architecture Practices for Achieving Agile at Scale

Robert Nord

In collaboration with Stephany Bellomo, Ipek Ozkaya, SEI and Philippe Kruchten, University of British Columbia

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0365

Abstract

The phrase *“agile architecture”* evokes two concepts:

1. An architecture that is versatile, easy to evolve, and easy to modify, while resilient enough not to degrade after a few changes.
2. An agile way to define an architecture, using an iterative lifecycle, allowing the architectural design to tactically evolve over time, as the problem is better understood.

In the best of worlds, we’d like an agile process that leads to a flexible architecture.

This presentation will introduce attendees to basic architecture concepts that developers use to develop large-scale systems in an agile lifecycle. These concepts include: the business case for architecture, architecture essentials, and architecting with just enough anticipation as an enabler for agile at scale.

Topics

Motivation

Why? The roles of architecture

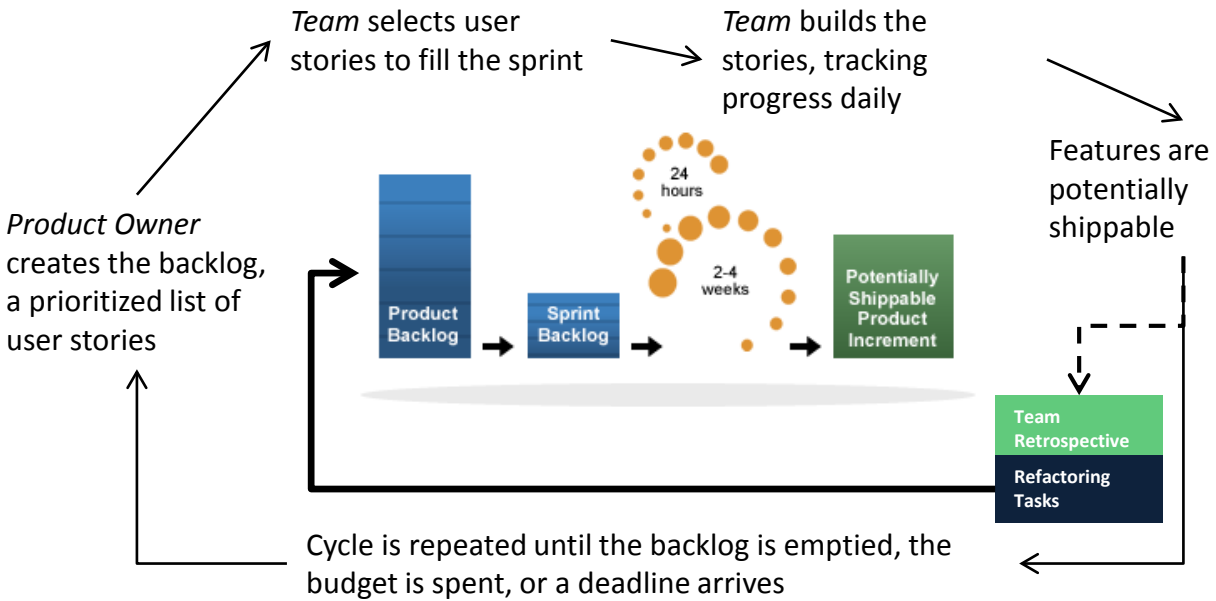
How? Essential activities

When? Release planning

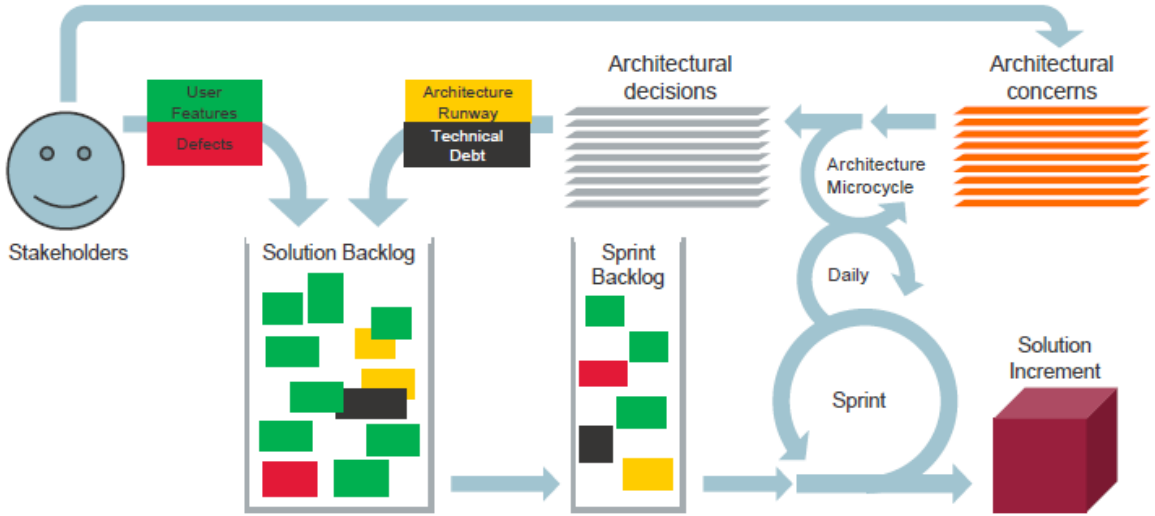
Who? Necessary organic capabilities

Take away

How Do You Adapt Scrum?

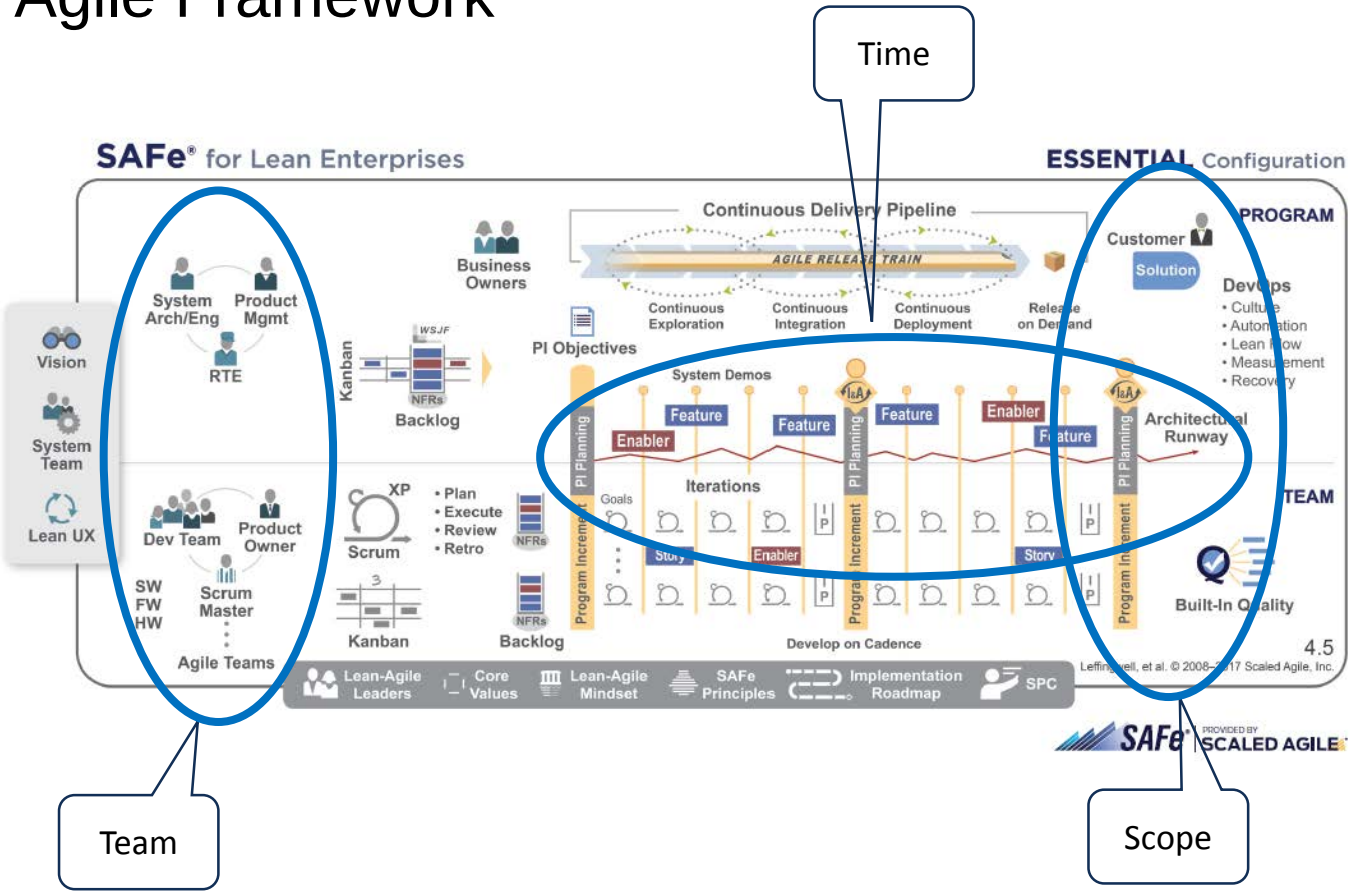


SCRUM and the Architecture Microcycle



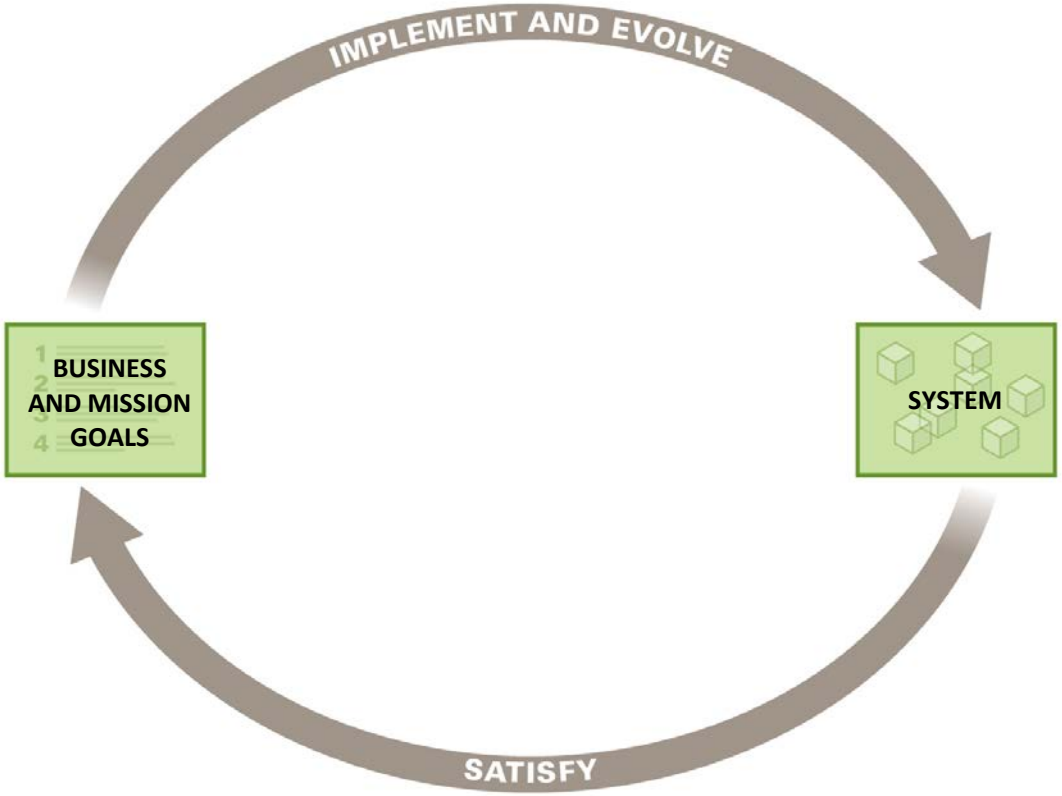
Poort, E. Selling the Business Case for Architectural Debt Reduction, *Ninth International Workshop on Managing Technical Debt* – XP 2017

Scaled Agile Framework

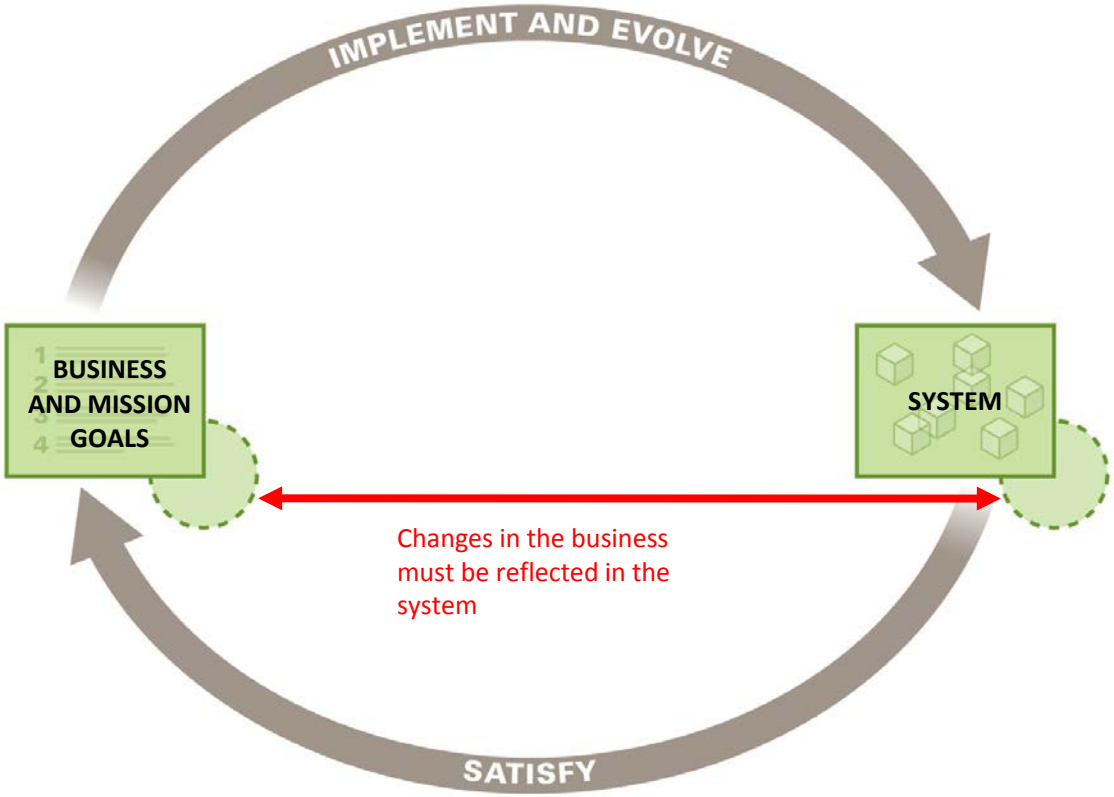


Why? The roles of architecture

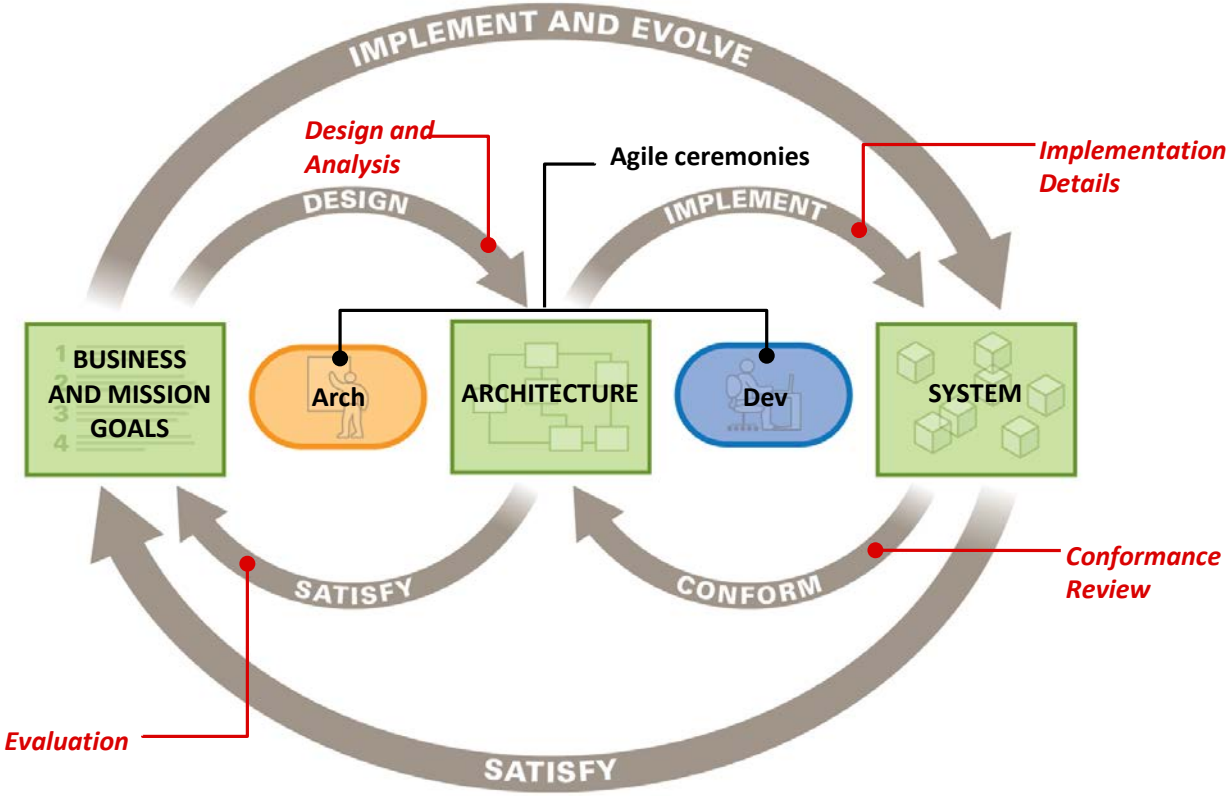
Architecture Practice



Architecture Practice



Architecture Practices



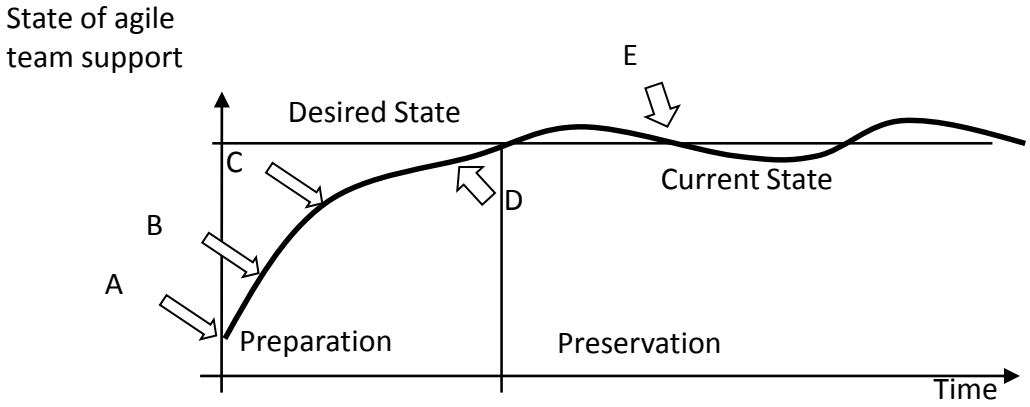
Value Proposition for Architecture

Architecture practice enables the ongoing cost-effective achievement of system-related business goals.

- Sound structure analyses provide objective confidence for achieving system quality.
- Appropriate flexibility enables cost-effective system maintenance and evolution.
- Early identification and mitigation of design risks result in fewer downstream problems and cost savings in integration, test and deployment.

How? Essential activities

How Much Support for Agile Development?

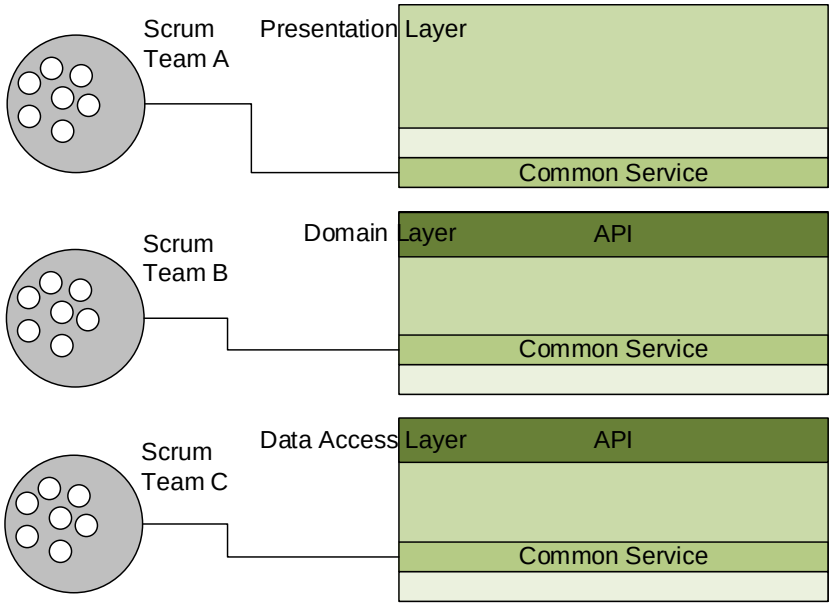


A – No support
B – Most important parts
Ready for the first feature
C – Almost ready with the support
Feature development starts

D – Desired state reached
E – Sustain the state

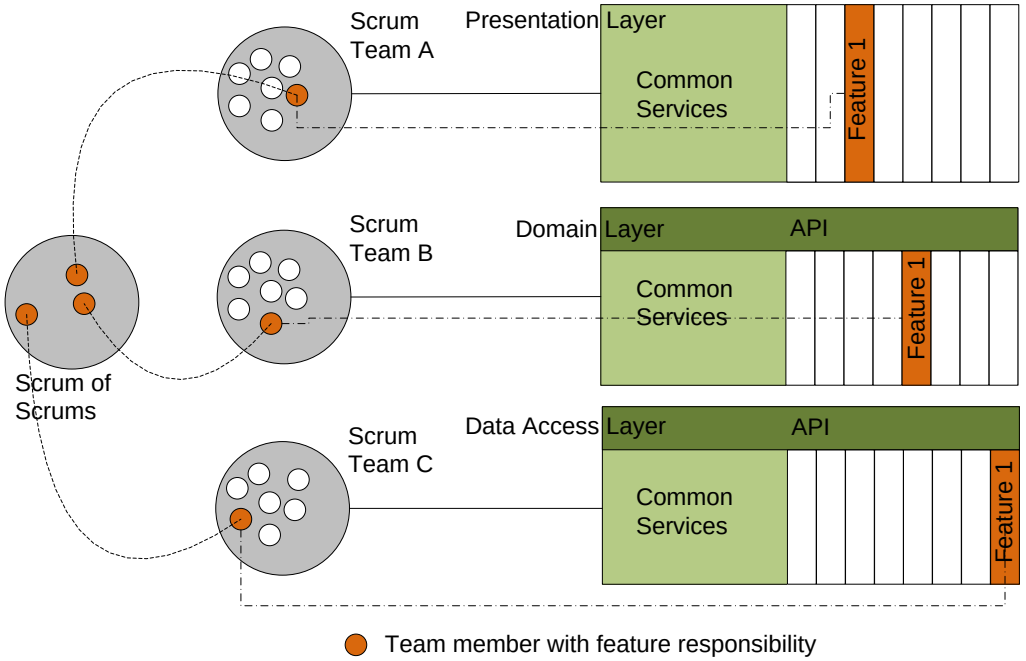
Bachmann, B., Nord, R.L., and Ozkaya, I. "Architectural Tactics to Support Rapid and Agile Stability," *Crosstalk*, May/June, 2012.

Applying the Practices in Concert -1



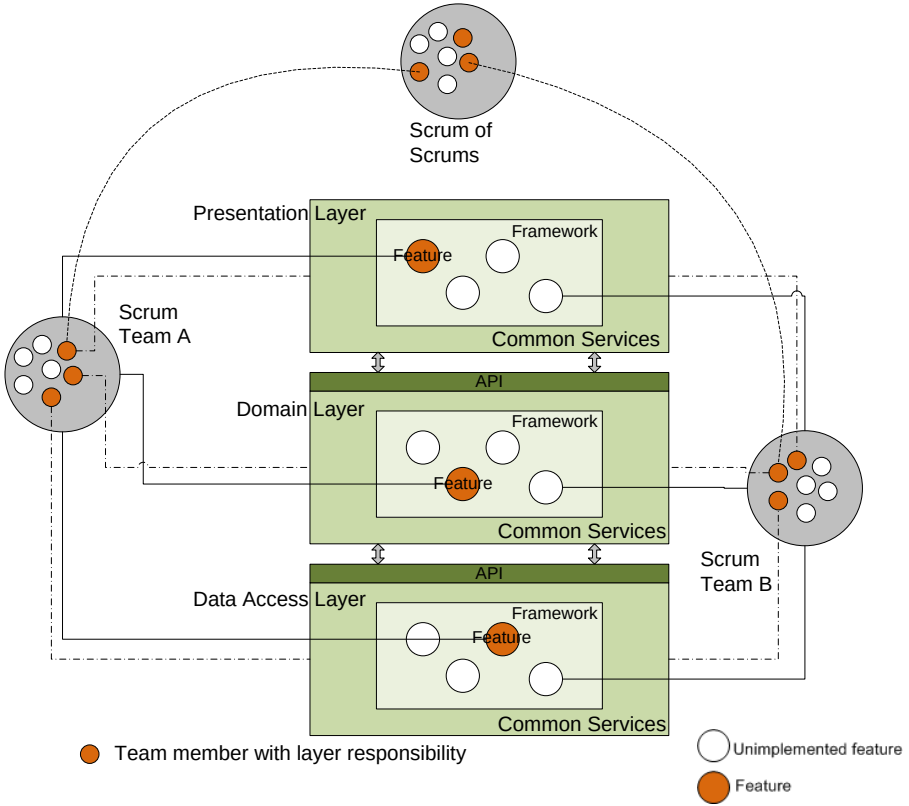
State A – Establishing the infrastructure

Applying the Practices in Concert -2



State B – Progressing architecture and feature development in parallel

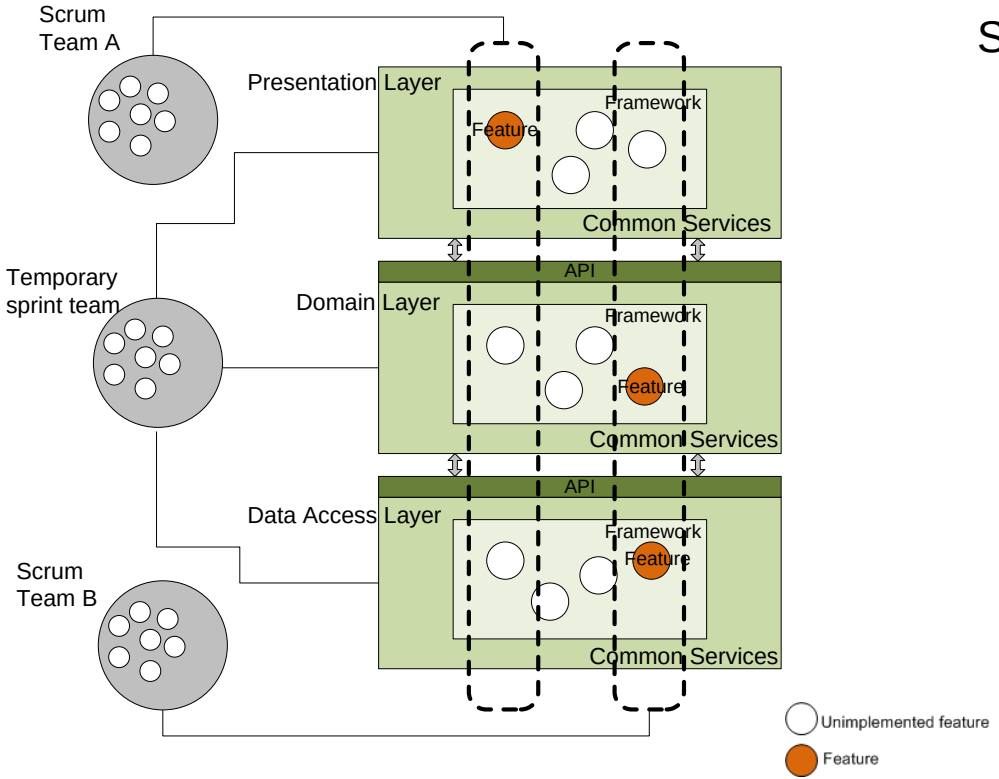
Applying the Practices in Concert -3



State C – Features

- different teams are assigned to different features,
- some team members keep layers and framework consistent

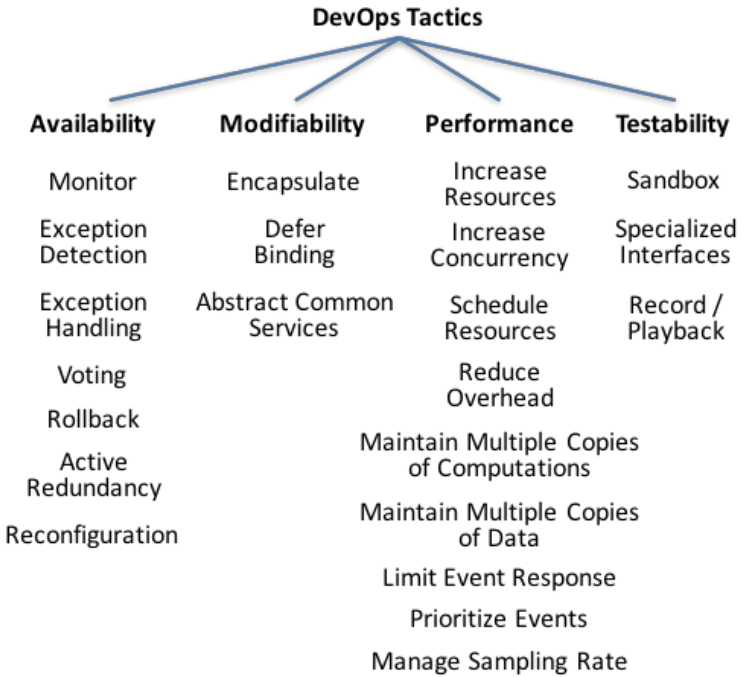
Applying the Practices in Concert -4



State D – Preservation

- different teams are assigned to different features,
- a temporary team prepares layers and frameworks for future feature teams.

Deployability Tactics



Tactics are design decisions that enable quality attributes.

There are tactics for

- Availability
- Interoperability
- Modifiability
- Performance
- Security
- Testability
- Usability

Bellomo, S., Kazman, R., Ernst, N., Nord, R.: Toward Design Decisions to Enable Deployability: Empirical Study of Three Projects Reaching for the Continuous-Delivery Holy Grail. In: *First International Workshop on Dependability and Security of System Operation*, pp. 32–37. IEEE Press, New York (2014)

When? Release planning

A Success Story

Challenges

- Measuring, planning, estimating, and tracking architectural design activities
- Integrating architectural design activities with iterative/incremental development
- Improving the as-practiced fidelity of the architecture development process

Project

- Industry funded project to build the worlds fastest stock trading engine.
- Challenging trading engine performance targets
- Aggressive project performance targets

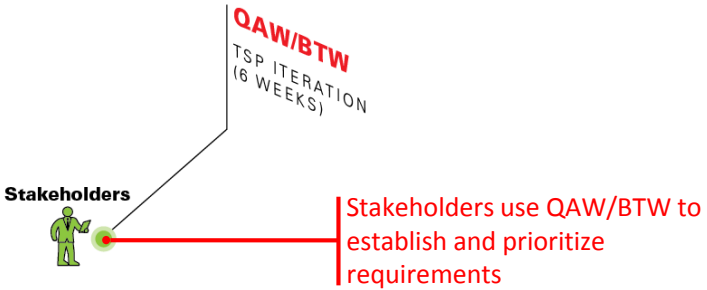
Results	Target	Actual
Latency	1ms	0.1ms
Throughput (transactions per sec.)	1,000	200,000
Schedule (months)	18	17
Quality (defects/KLOC found in validation)	0.25	0.1

Accomplishments:

- Successfully integrated
  - software process framework.
  - requirements and design practices.
  - quality practices.
- Architectural design practices
  - 12% of the total cost and were key in meeting the technical requirements
  - estimated to reduce implementation costs by 15%

Bachmann, F., Carballlo, L., McHale, J., and Nord. R. "Integrate End to End Early and Often." IEEE Software, July/August 2013.

Iterations

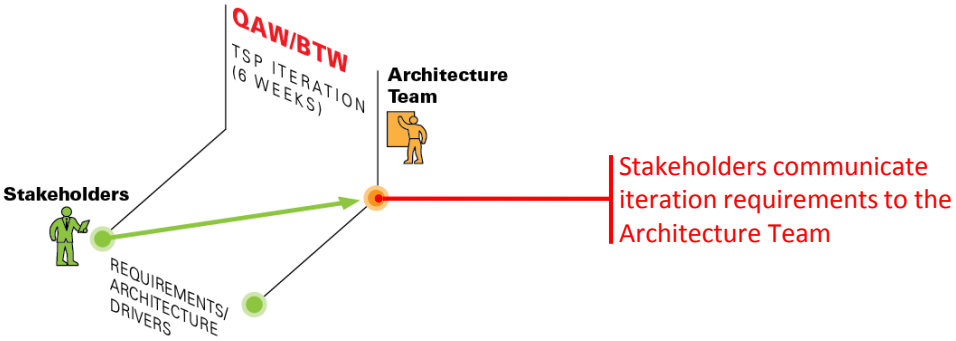


Nord, R., McHale, J., and Bachmann, F. Combining Architecture-Centric Engineering with the Team Software Process (CMU-SEI-TR-031). Software Engineering Institute, December 2010.

Practices:

- Architecture quick look
- Training managers, developers
- Quality attributes
- Release plan

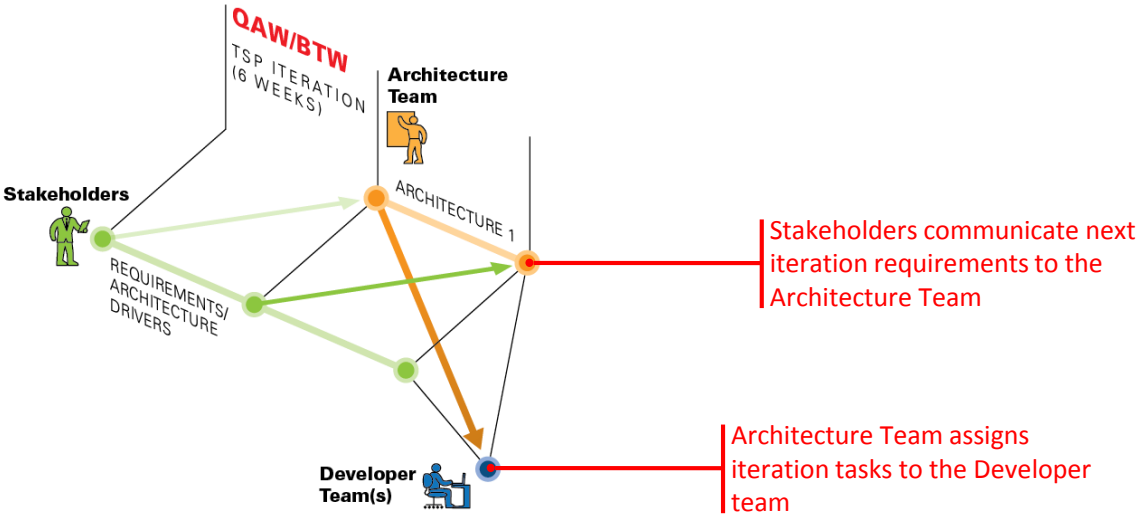
Iteration 1 – Find problems



Practices:

- Architecture design
- Design principles and tactics
- Scenario-based peer reviews
- Prototype problems

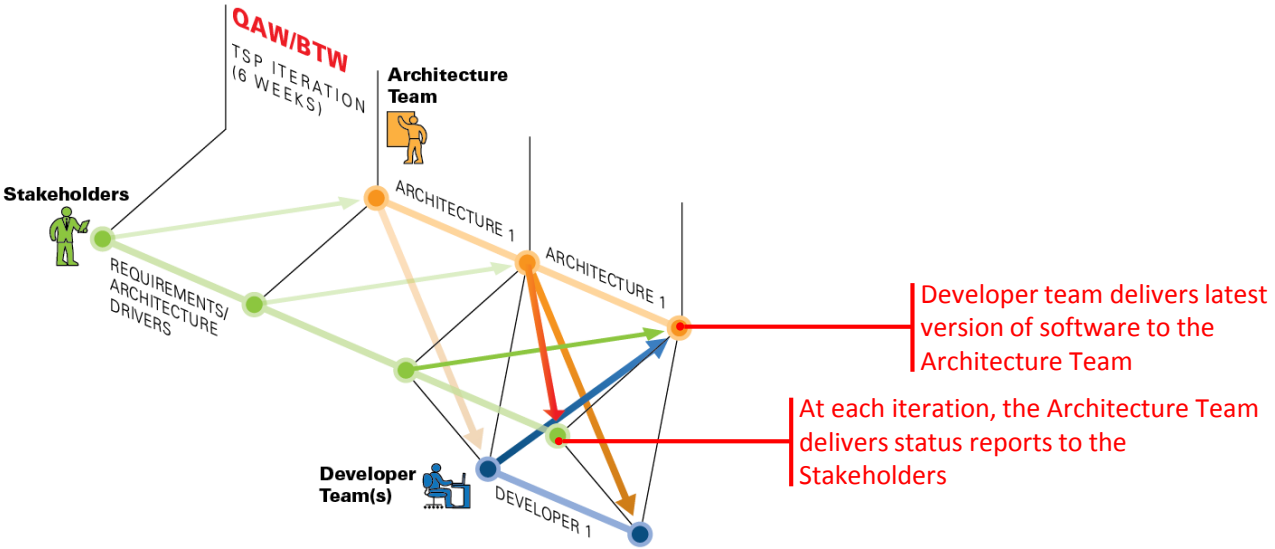
Iteration 2 – Design known



Practices:

- Architecture design
- Overall system structure
- Scenario-based peer reviews
- Architecture runway

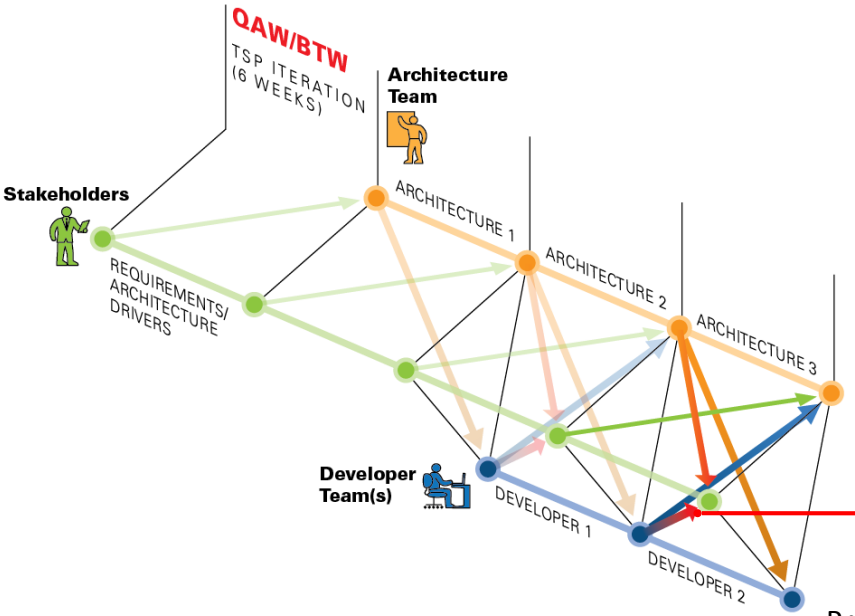
Iteration 3 – Design rest



Practices:

- Architecture design
- Allocating features
- Scenario-based peer reviews
- Features using infrastructure

Iteration 4 – Review

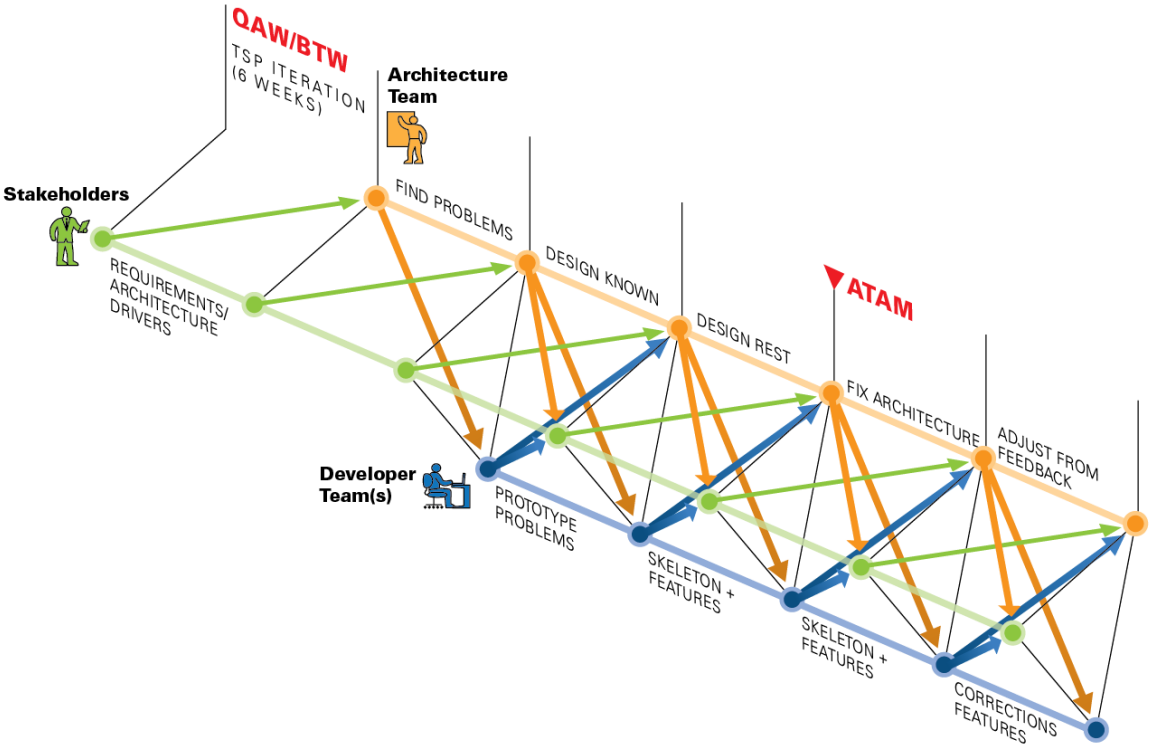


Developer Team now releases version updates to the Stakeholders

Practices:

- Architecture content
- Architecture tradeoff analysis
- Active design review
- Conformance guidelines

Iterations – Summary



Who? Necessary organic capabilities

Architects: Anchors or Accelerators to Organizational Agility?

You can’t outsource architecture oversight

- You need the organic capabilities to own the architecture, though you can get help and contract out tasks
- Architecture is strategy
- System outlives contracts

How can architects accelerate agility in organizations?

- Be agile
- Be architects of structure, time, and transition
- Create agile design guidelines

Jim Highsmith 2010

Architecture Quick Look

Business and development stakeholders

- investigate agile software development and architecture principles and practices
- identify risks and factors worthy of attention
- set priorities in improving software development

Why? The risks uncovered in this analysis will guide the application of the other architecture practices.

Ozkaya, I., Gagliardi. M., and Nord, R.L. 2013. Architecting for Large Scale Agile Software Development: A Risk-Driven Approach, *Crosstalk* 26, 3 (May/June 2013): 17-22.

Agilely architecting an agile architecture

Agilely architecting an agile architecture has four key requirements:

1. Focus on ***key quality attributes*** and incorporate these into technical explorations within prototyping and spikes
2. Understand that a successful product is a combination of customer-visible features and the underlying ***infrastructure that enables*** those
3. Recognize that an architecture that enables ease of maintainability and evolvability is the result of ***ongoing, explicit attention***
4. Continuously manage ***dependencies*** between functional and architectural requirements and ensure that the architectural foundation is put in place in a just-in-time manner

Bellomo, S., Kruchten, P., Nord, R.L., Ozkaya, I.: How to Agilely Architect an Agile Architecture? *Cutter IT J.* 27, 12–17 (2014)

Contact Information

Robert Nord
Architecture Practices Initiative
Email: rn@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/architecture

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257

Quality Attribute Scenarios (reference)

1. ***Stimulus*** - the condition that affects the system
2. ***Response*** - the activity that results from the stimulus
3. ***Source of Stimulus*** - the entity that generated the stimulus
4. ***Environment*** - the condition under which the stimulus occurred
5. ***Artifact*** - the entity that was stimulated
6. ***Response Measure*** - the measure by which the system's response will be evaluated

Architecture Practices (reference)

Core Practices

Elicit and capture business and mission goals in the form of quality attribute scenarios.

Iteratively and incrementally transform scenarios into architectural structure and content.

Evaluate the architecture for risks to achievement of the scenarios.

Transition the architecture to implementation, build it, and verify compliance.

Supporting Methods

Quality Attribute Workshop (QAW)
Business Thread Workshop (BTW)
Architecture Roadmap

Attribute-Driven Design (ADD)
Views and Beyond (V&B)

Architecture Tradeoff Analysis Method (ATAM)
Scenario-based peer reviews

Active Design Review (ARID)
Design and implementation rules
Conformance reviews

www.sei.cmu.edu/architecture