Current SAT Work in Architecture Evolution

Rick Kazman

representing the work of:

Joe Batman, Andrés Díaz-Pace, Mark Klein, Robert Nord, Ipek Ozkaya

Software Engineering Institute | Carnegie Mellon
Current SAT Work in Architecture Evolution

Carnegie Mellon University, Software Engineering Institute (SEI), Pittsburgh, PA, 15213


Approved for public release; distribution unlimited
Our View of Architecture Evolution

Architecture evolution is the process of designing an architecture to meet today’s and tomorrow’s business goals, while maximizing expected value, in the face of uncertainty.

Architecture evolution therefore has two foundations:

- Architecture design
- Software engineering economics

In this talk I will sketch our approaches to both.
Evolution with a Design Focus

The *design aspect* of architecture evolution investigates how to measure and analyze the difficulty of reaching a desirable future state given today's state and how to design the system so that desirable states are easier to reach.

We wish to be able to answer questions such as:

- *How can the quality of a deficient architecture be improved?*
- *What alternative architectures provide the greatest flexibility to support future requirements?*
- *How resilient is the system to change?*
Consider the following design scenario:

- We have an important feature to implement for our new system.
- We intend to implement it in the 2nd release. This feature depends on an “unimportant” piece of infrastructure, which doesn't get done in release 1.
- This means that you have to do the unimportant infrastructure in release 2, and delay the feature to release 3 (by which time it is of less value).

What is the point of this example?

In designing for evolution we need to think about trajectories and dependencies, not single design steps.

*Shameless plug*: we will be discussing just such situations in the Evolution Workshop. Come join us!
Evolution with an Economics Focus

Software architecting cannot take place in a value-neutral setting.

Architecture decisions are the results of trade-offs and economic factors influence such trade-offs constantly.

Economics-driven architecting is a direction that must be rigorously pursued to produce systems that meet all their business goals. It is based on the following principles about architecture:

- *Principle 1*: Quality attribute requirements are a driving force for software and system architecture design.
- *Principle 2*: Using architectural patterns and tactics is a way to achieve quality in software.
- *Principle 3*: Business and mission goals endow quality with value.
Evolution with an Economics Focus - 3

Architecture evolution involves:

- exploring the design space, looking for risks and opportunities
- maximizing expected value of “reachable” designs under conditions of uncertainty
- minimizing the expected loss of “unreachable” designs under conditions of uncertainty
- staging of architectural strategies and features (release planning)
## Related Work in Evolution

### APPROACHES

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Transformation</th>
<th>Value</th>
<th>Sequencing</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>practice</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>code metrics and refactoring</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product lines *</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quality-attribute design and analysis*</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATAM® and QAW*</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>enterprise framework and migration planning*</td>
<td>□</td>
<td>□</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>transition</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>model transformations</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tactics and patterns *</td>
<td>■</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>impact analysis</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>plateaus and waves</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value-based software engineering</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBAM*</td>
<td>□</td>
<td>□</td>
<td></td>
<td></td>
</tr>
<tr>
<td>history-based feature analysis</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conceptual framework for uncertainty</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**KEY:** Relevance to architecture evolution: ■ - very relevant, □ - relevant, □ - somewhat related.
Square size indicates amount of activity/efforts. * - SEI-related approaches.
Our Work in Evolution (with an Economics Focus)

Cost Benefit Analysis Method (CBAM)/Utility theory
Real options for valuing flexibility
Release planning and optimization
Architecture patterns/tactics as value-creating design operations.

Common to all the work is the application of value-based activities

• that are practical and easily implemented
• on a firm, principled basis
• with simple, clear rationale

Consider how economics enters into evolutionary design decisions...
Different patterns exhibit different quality attributes which in turn offer different levels of utility.

Future utility curves are affected by uncertainty.
It is necessary, but not sufficient to consider the strengths and weaknesses of patterns with respect to quality attributes.
Key Concepts in Evolution

Evolution trajectory

System

Business and Mission Goal

Quality Attribute Concern

Quality Attribute

Uncertainty

Utility

Architecture

Design strategies

Resources

Feature

Function

Quality Attribute Scenario

Explicit focus and use of techniques to better incorporate economic considerations into the process of architecting

Design the evolution trajectory, which endows multiple instances of system through out the life cycle of the system

Focus on architecture centric practices to design the evolution trajectory for each instance of the system

Design the evolution trajectory, which endows multiple instances of system through out the life cycle of the system
A Proto Architecture Evolution Method

ELICIT THE BUSINESS AND VALUE CONTEXT

- Elicit business goals
- Elicit quality attribute scenarios
- Elicit cost and schedule constraints

Generate Utility Curves

Business and mission goals induce a preference relation over quality attributes.

EXPLORE THE DESIGN CONTEXT

- Review current architectural approaches
- Generate future state architectural approaches
- Identify interactions between approaches

EXPLORE EVOLUTION DIRECTIONS

- Explore Opportunity Enablement

EXPLORE REAL OPTIONS

- Use real options to develop architectural strategies for managing uncertainty for future evolution.

IDENTIFY EVOLUTION STEPS

- Use architecture to understand cost and resource constraints. Use release planning to design the architecture evolution trajectory.

ALLOCATE ARCHITECTURE TRAJECTORY

- Architecture design exploration and analysis may reveal exploitable core competencies.
Goals of Our Ongoing Work

Here are some of our goals for the coming year:

- Package our existing techniques as a method.
- Give guidance to practitioners to allow them to reason about uncertainty vis-a-vis architecture.
- Give guidance on the ramifications of design decisions and trajectories of design decisions.
- Aid practitioners in making a business case for the value of architecture.
- Provide tool support for exploring the design space.
Further Reading


- [http://www.sei.cmu.edu/architecture/products_services/cbam.html](http://www.sei.cmu.edu/architecture/products_services/cbam.html)