Carnegie Mellon University
Software Engineering Institute

A Disciplined Approach to Making and Analyzing Architectural Decisions

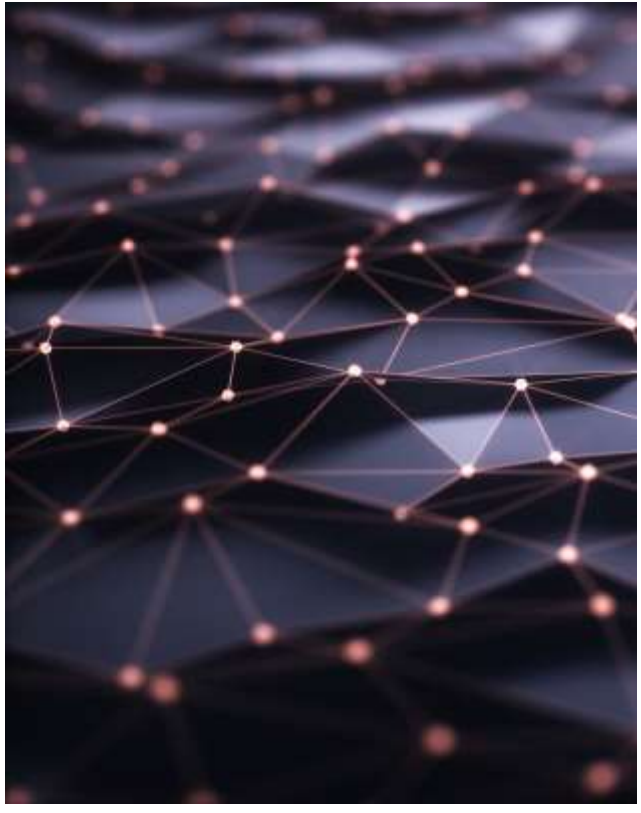
SEPTEMBER 18, 2023

Rick Kazman, Phil Bianco, Sebastián Echeverría, James Ivers, John Klein

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. The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

DM23-0942

Architecture as an Enabler

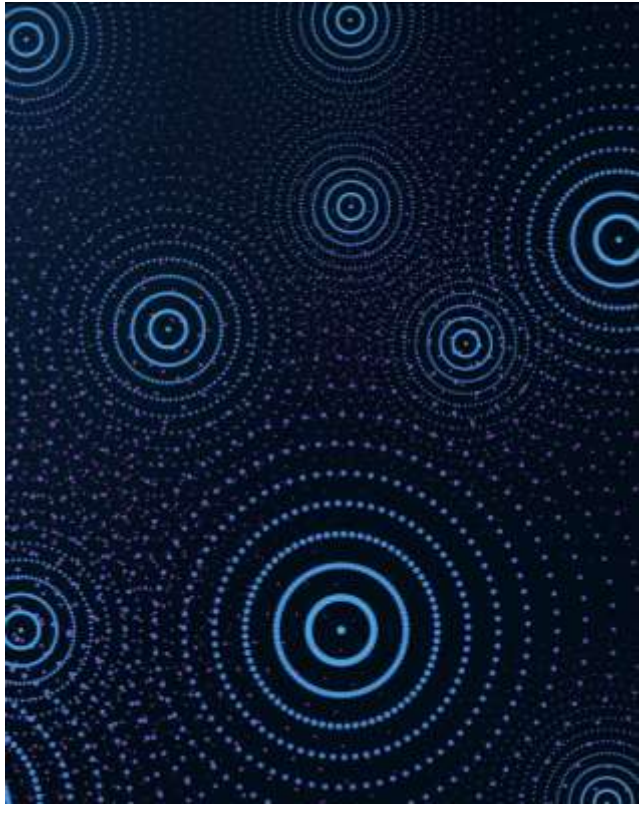


Architecture is key to realizing driving quality attribute a (maintainability, robustness, extensibility, integrability, etc.) that directly impact long-term cost and schedule goals.

Architecture helps organizations proactively manage changes in requirements and technology, providing value throughout a system's lifetime.

As such, architectures need to be updated and used for new analyses over time to help keep systems healthy.

Architecture Tradeoffs



When creating and evolving an architecture, tradeoff decisions are inevitable.

- A decision made to satisfy one requirement may undermine another.

However, the impact of decisions across multiple requirements are not always explicitly assessed and are sometimes difficult to assess at all.

As such, architectural decisions may be inappropriate when first made or may be undermined by other decisions as a system evolves, scales, and is maintained.

Tradeoff Examples

Context is vital in architecture.

- implementation time vs. maintainability
- cost and complexity vs. throughput and robustness
- extensibility vs. latency
- maintainability vs. robustness
- portability vs. performance
- implementation time vs. run-time performance
- latency vs. space
- testability vs. latency

The Problem

How can we gain more confidence that architectural decisions are appropriate?

Specifically, how do we

- Elicit architectural requirements?
- Make architectural decisions?
- Analyze the decisions made?

The Solution

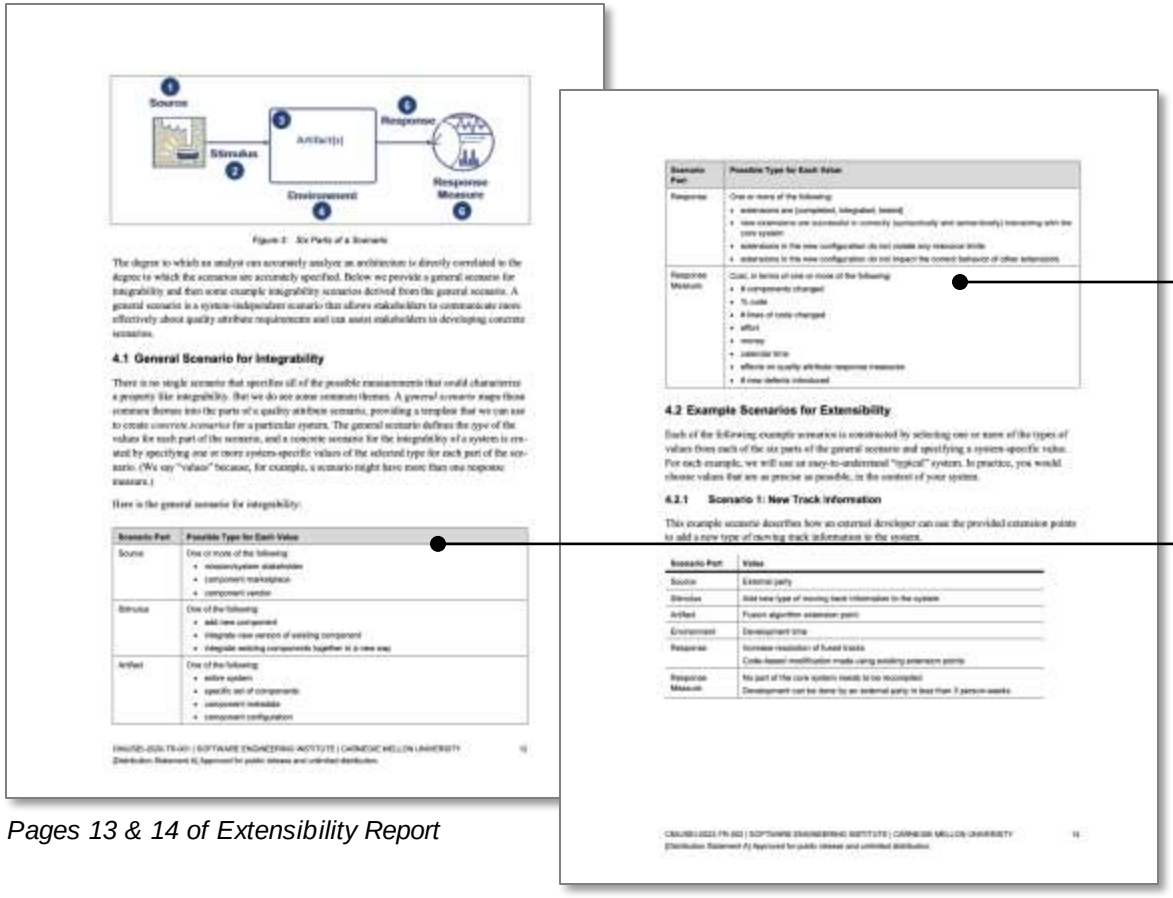
We offer a **disciplined** approach to making and analyzing architectural decisions, available as reports on **architecting to predictably achieve critical quality attributes**.

Report Structure

Each report provides

- a set of **definitions**, **core concepts**, and a **framework for reasoning** about quality attributes and their satisfaction by an architecture and a system
- a **template** for eliciting quality attribute requirements as scenarios
- a set of mechanisms—**patterns and tactics**—that are commonly used to satisfy quality attribute requirements
- a step-by-step method—a **playbook**—that analysts can use to determine whether the architectural decisions contain serious risks relative to the quality attribute requirements

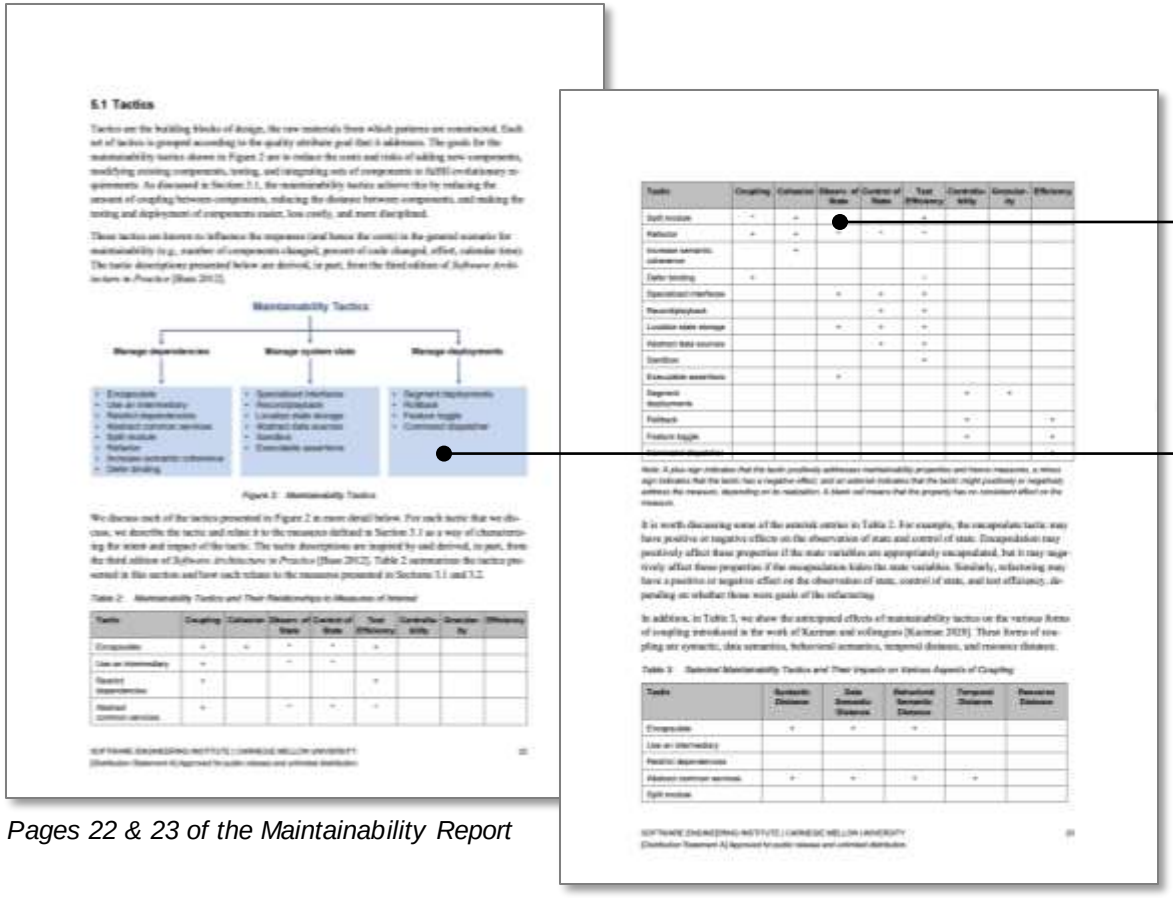
Extensibility Scenario Template



Pages 13 & 14 of Extensibility Report

EXTENSIBILITY SCENARIO TEMPLATE
Architecture requirements can be captured as structured quality attribute scenarios.

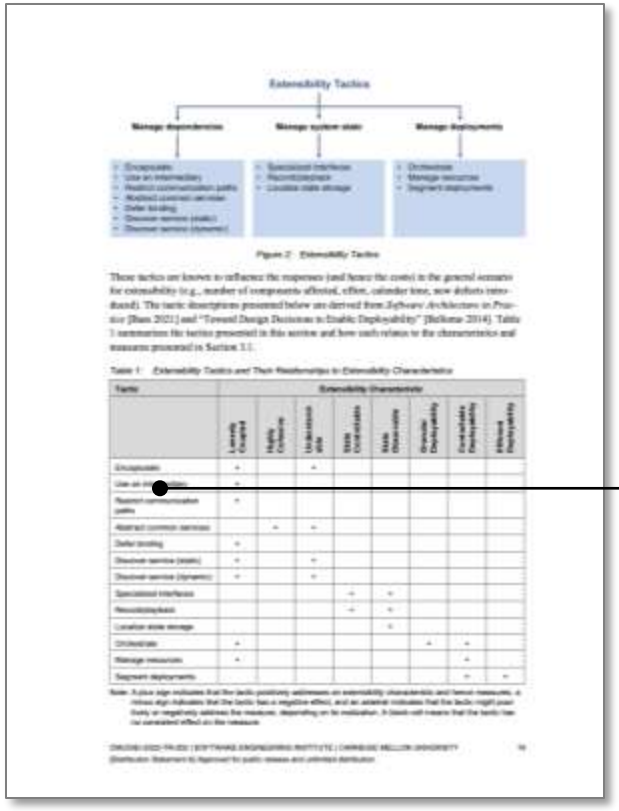
Maintainability Tactics



Pages 22 & 23 of the Maintainability Report

MAINTAINABILITY TACTICS
Tactics are building blocks of design decisions for achieving quality attribute requirements.

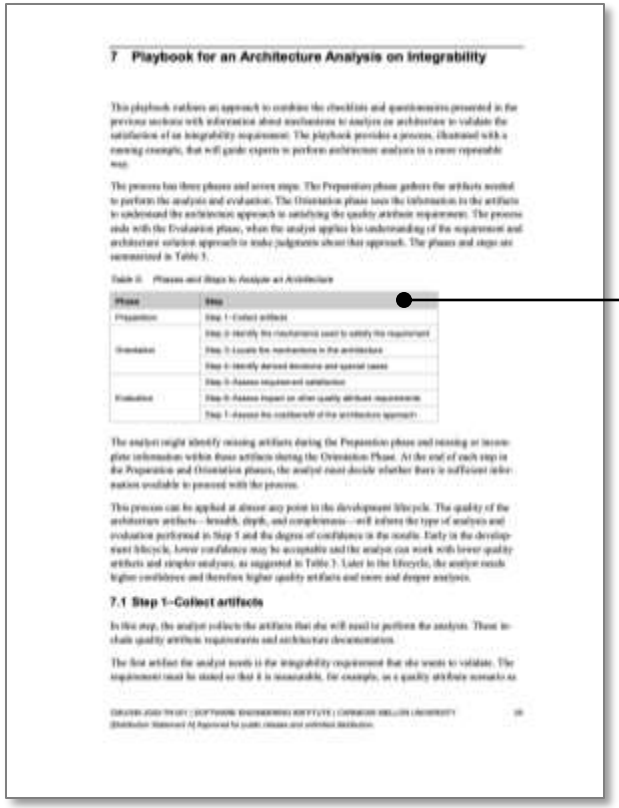
Characteristics of Extensibility



Page 18 of the Extensibility Report

QUALITY ATTRIBUTE MATRIX
Quality attributes are complex, and tactics influence distinct characteristics.

Integrability Playbook



7 Playbook for an Architecture Analysis on Integrability

This playbook outlines an approach to combine the checklists and questionnaires presented in the previous sections with information about mechanisms to analyze an architecture to validate the satisfaction of an integrability requirement. The playbook provides a process, illustrated with a running example, that will guide experts to perform architecture analyses in a more repeatable way.

The process has three phases and seven steps. The Preparation phase gathers the artifacts needed to perform the analysis and evaluation. The Orientation phase uses the information in the artifacts to understand the architectural approach to satisfying the quality attribute requirements. The process ends with the Evaluation phase, where the analyst applies his understanding of the requirement and architecture solution approach to make judgments about that approach. The phases and steps are summarized in Table 3.

Table 3: Phases and Steps to Analyze an Architecture

Phase	Step
Preparation	Step 1: Collect artifacts
Orientation	Step 2: Identify the mechanisms used to satisfy the requirement
	Step 3: Locate the mechanisms in the architecture
	Step 4: Identify derived decisions and special cases
Evaluation	Step 5: Assess requirement satisfaction
	Step 6: Assess impact on other quality attribute requirements
	Step 7: Assess the cost/benefit of the architecture approach

The analyst might identify missing artifacts during the Preparation phase and missing or incomplete information within those artifacts during the Orientation Phase. At the end of each step in the Preparation and Orientation phases, the analyst must decide whether there is sufficient information available to proceed with the process.

This process can be applied at almost any point in the development lifecycle. The quality of the architecture artifacts—breadth, depth, and completeness—will inform the type of analysis and evaluation performed in Step 1 and the degree of confidence in the results. Early in the development lifecycle, lower confidence may be acceptable and the analyst can work with lower quality artifacts and simpler analyses, as suggested in Table 3. Later in the lifecycle, the analyst needs higher confidence and therefore higher quality artifacts and more and deeper analyses.

7.1 Step 1—Collect artifacts

In this step, the analyst collects the artifacts that she will need to perform the analysis. These include quality attribute requirements and architecture documentation.

The first artifact the analyst needs is the integrability requirement that she wants to validate. The requirement must be stated so that it is measurable, for example, as a quality attribute scenario as

[DISTRIBUTION STATEMENT] (Approved for public release and unlimited distribution.)

Page 39 of the Integrability Report

PLAYBOOK
Each report includes a playbook that illustrates a step-by-step method for analyzing architectural decisions with an example.

For More information

This QR code will take you to the set of reports: