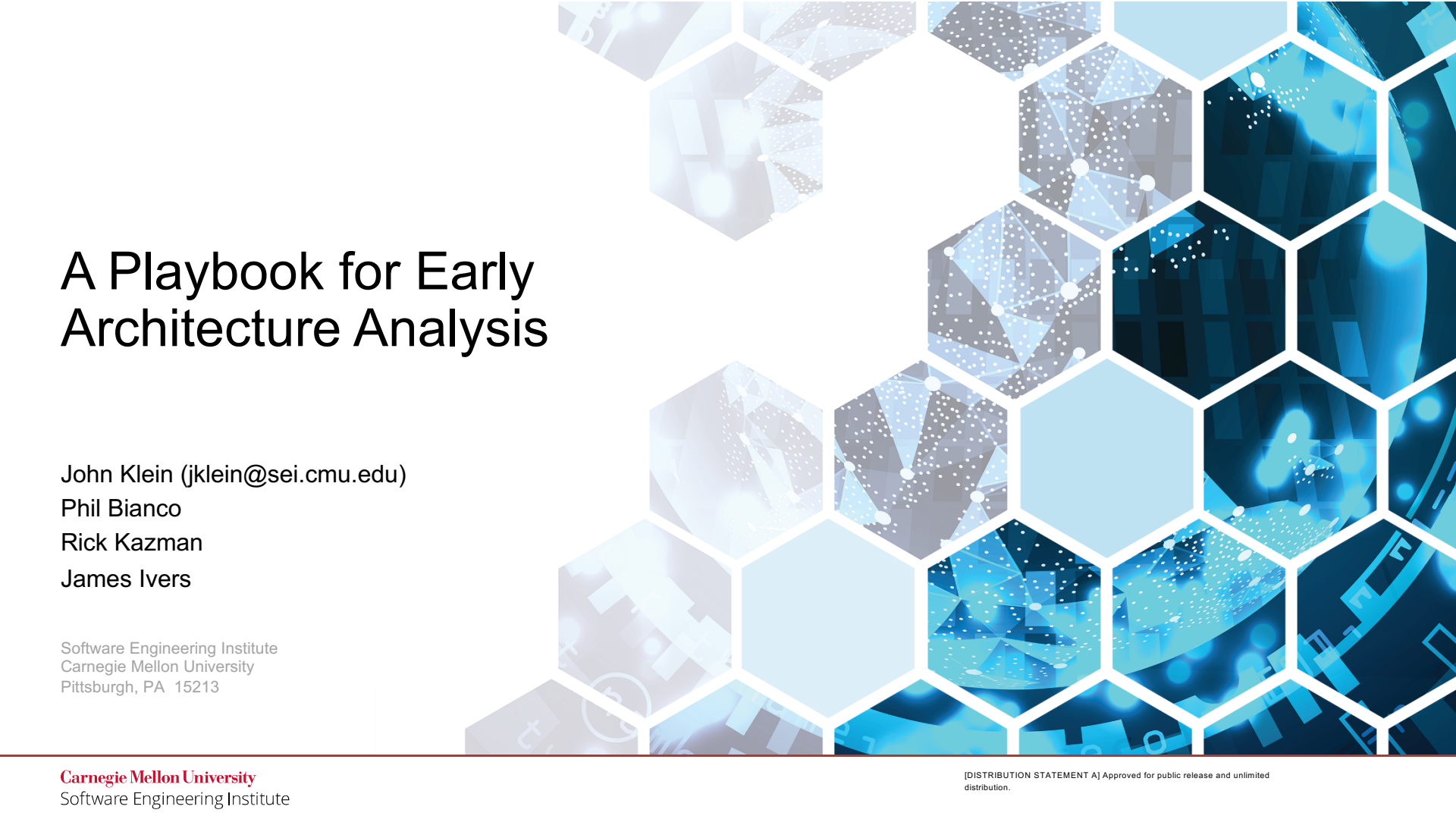




A Playbook for Early Architecture Analysis

John Klein (jklein@sei.cmu.edu)

Phil Bianco

Rick Kazman

James Ivers

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Copyright 2020 Carnegie Mellon University.

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.

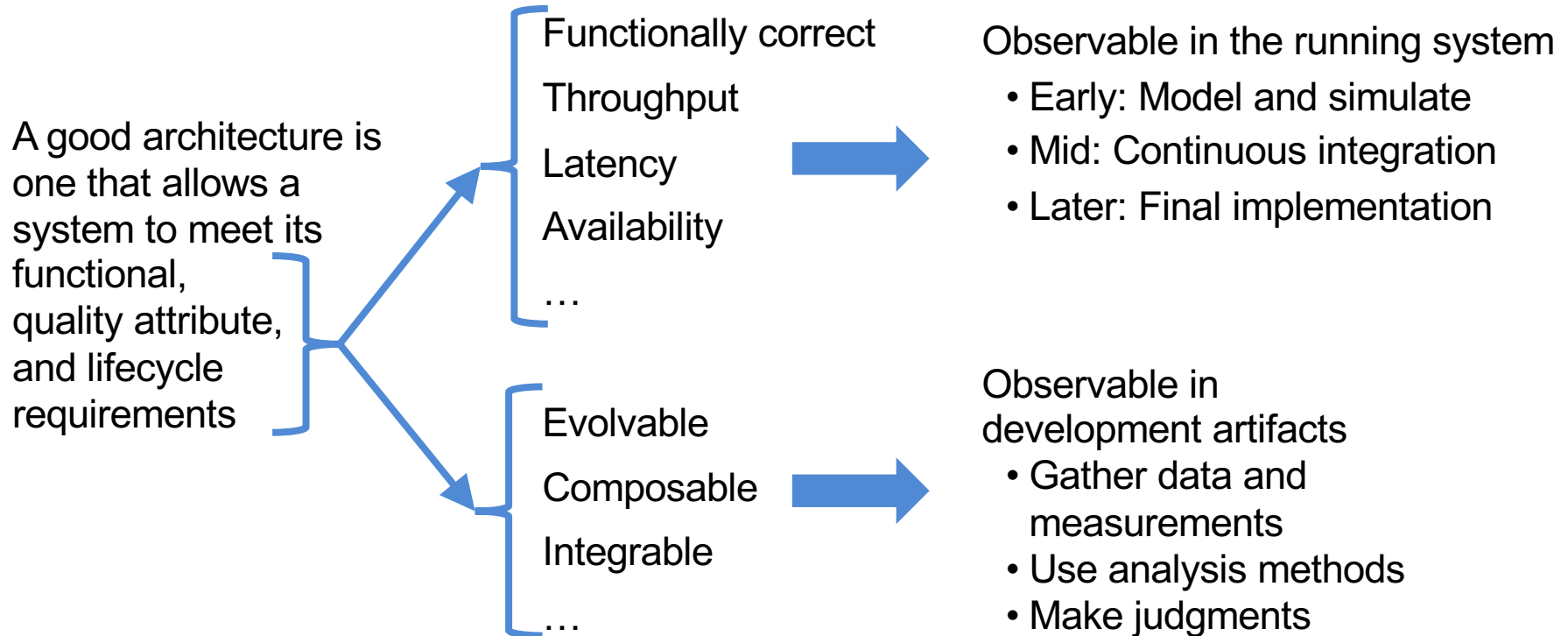
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM20-0963

An architecture is not inherently good or bad – It is *fit for purpose*



Architecture Analysis Challenge

Challenge: Make judgments about architectural qualities such as integrability...

- More systematic and repeatable
- Less reliant on the expertise of the analyst

Solution: Develop a *playbook* for the analysis process for each quality

- Applicable throughout the lifecycle, from early concept through implemented system
- Include practical information about specifying requirements, design mechanisms, measurements, and evaluation checklists and questionnaires

Outcome: Acquirers can verify that they are getting an architecture that will allow the system to meet its functional, quality attribute, and lifecycle requirements

Architecture Analysis Playbook

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 quality of the architecture artifacts—breadth, depth, and completeness—sets the degree of confidence in the results

- Early: Lower confidence may be acceptable → simpler analysis in Step 5 using lower quality artifacts
- Later: Higher confidence needed → deeper and more precise analysis using higher quality artifacts

Phase 1 – Preparation

Step 1 – Collect artifacts:

- Requirement to validate – playbook guidance includes quality attribute general scenario to specify a measurable requirement
- Other requirements – architecture design is a tradeoff process, improving one quality can reduce another
- Architecture documentation, for example:
 - Early – list of architecture approaches
 - Later – complete architecture views

Does the quality of the artifacts support the needed degree of confidence in the analysis results?

Phase 2 – Orientation

Step 2 – Identify the mechanisms used to satisfy the requirement

- Mechanisms: Patterns, tactics, frameworks
- Look at rationale discussions, trade studies, architecture views

Step 3—Locate the mechanisms in the architecture

- How is the mechanism used?
- For example, the trade study says that the architecture uses a “pipe and filter approach for the signal processing pipeline”
 - How many stages in the pipeline? Fixed or able to vary?
 - Playbook guidance includes checklists and questionnaires that identify these questions.

Step 4 – Identify derived decisions and special cases

- Derived decision: If the pipeline can vary, when is that defined – code or a configuration file?
- Special case: If configuration can change while the system is running, what happens to data in the pipeline?

What are the questions that the analysis must answer?

Phase 3 – Evaluation

Step 5 – Assess requirement satisfaction

- Answer the questions noted in Phase 2: Are the chosen mechanisms sufficient? Are derived decisions sufficient? Are any special cases addressed?
- Playbook guidance includes measurements, checklists, and questionnaires to support this step
- Identify issues: Possible gaps, deficiencies, ambiguities

Step 6 – Assess impact on other quality attribute requirements

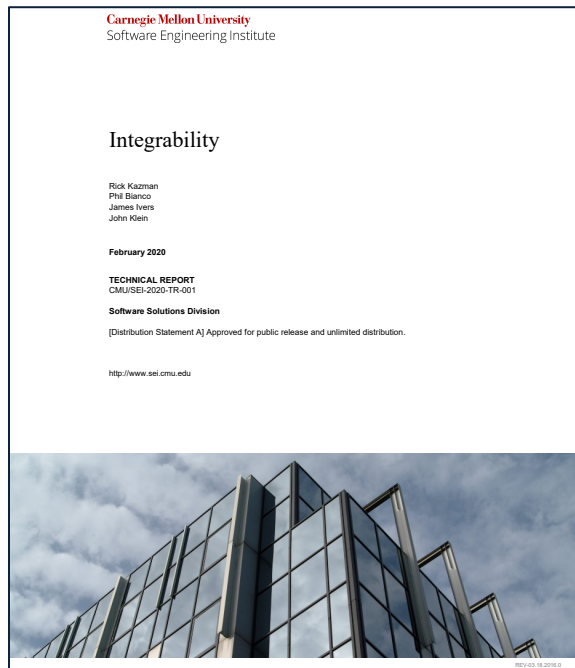
- “Everything is a tradeoff”
- How do the mechanisms and decisions assessed in Step 5 affect other qualities?
- For example, a pipe and filter mechanism can increase latency and configuration files can impact testability.

Step 7 – Assess the cost/benefit of the architecture approach

- This is a judgement call – is any added complexity needed?

Is the proposed approach sufficient and necessary?

Where to find all the details...



Series of technical reports, one per quality attribute

- *Integrability* is released, *Maintainability* and *Robustness* are coming soon

Use to construct requirements and evaluate architecture against requirements

Each report contains

- Definitions
- Measures
- Scenario templates and examples
- Mechanisms – patterns and tactics
- Analysis methods – questionnaires, checklists, measurement
- Evaluation playbook

<http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=637375>

jklein@sei.cmu.edu

info@sei.cmu.edu