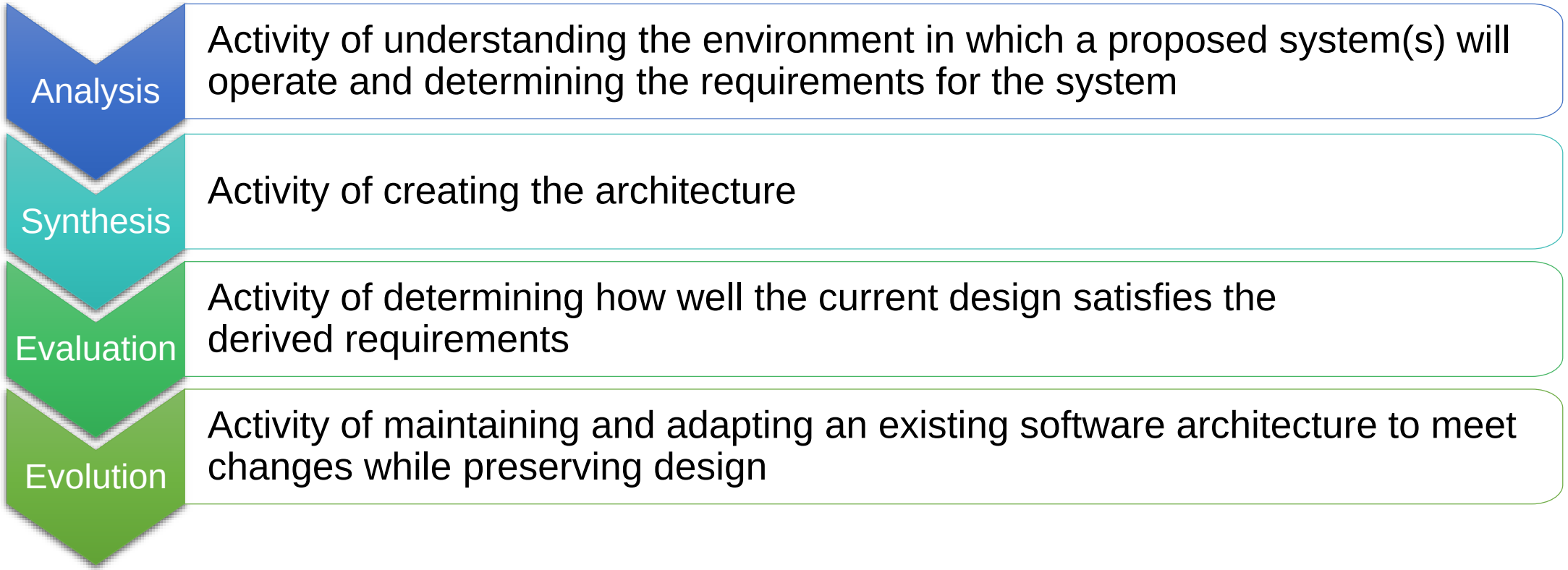
Architectural evaluation

Prof. Dr. Everton Cavalcante

http://www.dimap.ufrn.br/~everton/

Architectural activities

The four core architectural activities



Quality-based software architectures

Several architectures can be designed to satisfy functional requirements, but they may vary in the degree in which non-functional requirements are satisfied

– A quality-based architecture is one designed to satisfy a quality attribute or a set of quality attributes

– In most cases, it is impossible to maximize all quality attributes and hence the architect should consider a trade-off


PORTABILITY
Installability
Replaceability
Adaptability
FUNCTIONAL SUITABILITY
Functional correctness
Functional completeness
Functional appropriateness
MAINTAINABILITY
Modularity
Reuseability
Modifiability
Testability
Analyzability
ISO/IEC 25010:2011
Systems and software Quality Requirements and Evaluation (SquaRE) products model
PERFORMANCE EFFICIENCY
Capacity
Resource utilization
Time behavior
COMPATIBILITY
Interoperability
Co-existence
SECURITY
Integrity
Confidentiality
Non-repudiation
Accountability
Authenticity
USABILITY
Operability
User error protection
RELIABILITY
Availability
Recoverability
Maturity
Fault tolerance

Goals

- To present the architectural evaluation activity and its outcomes
- To understand the importance of evaluating a software architecture
- To introduce some architectural evaluation methods

Objetivos

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

Objetivos

What are the aims of architectural evaluation?

What are the limitations of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

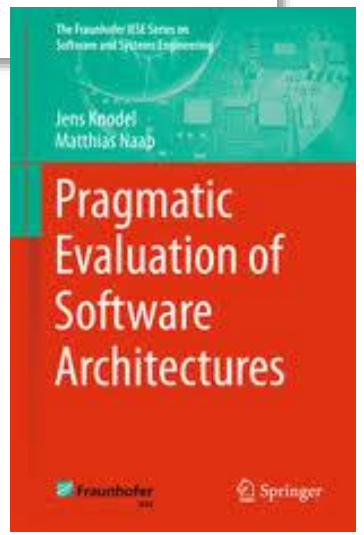
When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

Aims of the architectural evaluation

Architectural evaluation

Architectural evaluation does not aim at answering the question "Is this design (or the decisions leading to it) good or bad?" It rather evaluates whether the architecture is adequate to address the stakeholder concerns or not.



Jens Knodel, Matthias Naab. **Pragmatic evaluation of software architectures.** Switzerland: Springer International Publishing, 2016

Aims of the architectural evaluation

– To determine the quality of the (envisioned) software system
  - *How well are the stakeholders' requirements (driving the architecture) understood and agreed on?*
  - *How well are the solution concepts and design decisions of the architecture suited to adequately addressing the requirements?*
  - *How well are the solution concepts manifested in the implementation?*

Aims of the architectural evaluation

– To determine the quality of the artifacts created during architecting or derived from the architecture
  - *How well is the documentation of the architecture structured and how consistent is it?*
  - *How well is the source code of the software system structured and how readable is it?*

Aims of the architectural evaluation

– To determine if the right design decisions have been manifested correctly in the implementation
  - *Is the software architecture suitable to system for which it was designed?*
  - *Will the software achieve the quality requirements with that implementation?*

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

The importance of evaluating software architectures

– It represents a valuable, useful, and worthwhile instrument to mitigate risks when making decisions about software systems at any time in their life cycle
  - It requires investments, but it saves time and money by preventing wrong or inadequate decisions
  - It supports informed decision-making by creating awareness of risks
  - It allows understanding potential trade-offs and side effects in decision-making

The importance of evaluating software architectures

- It allows predicting properties of software systems before they are built or by answering questions about existing systems
  - The earlier a problem is found, the better
- It creates awareness of
  - ambiguities, unclarities or drift in stakeholder concerns and architecture
  - inadequate architecture decisions
  - deficiencies and inconsistencies in the architecture documentation
  - architectural drifts between the intended and the implemented architecture
  - architectural bad smells

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

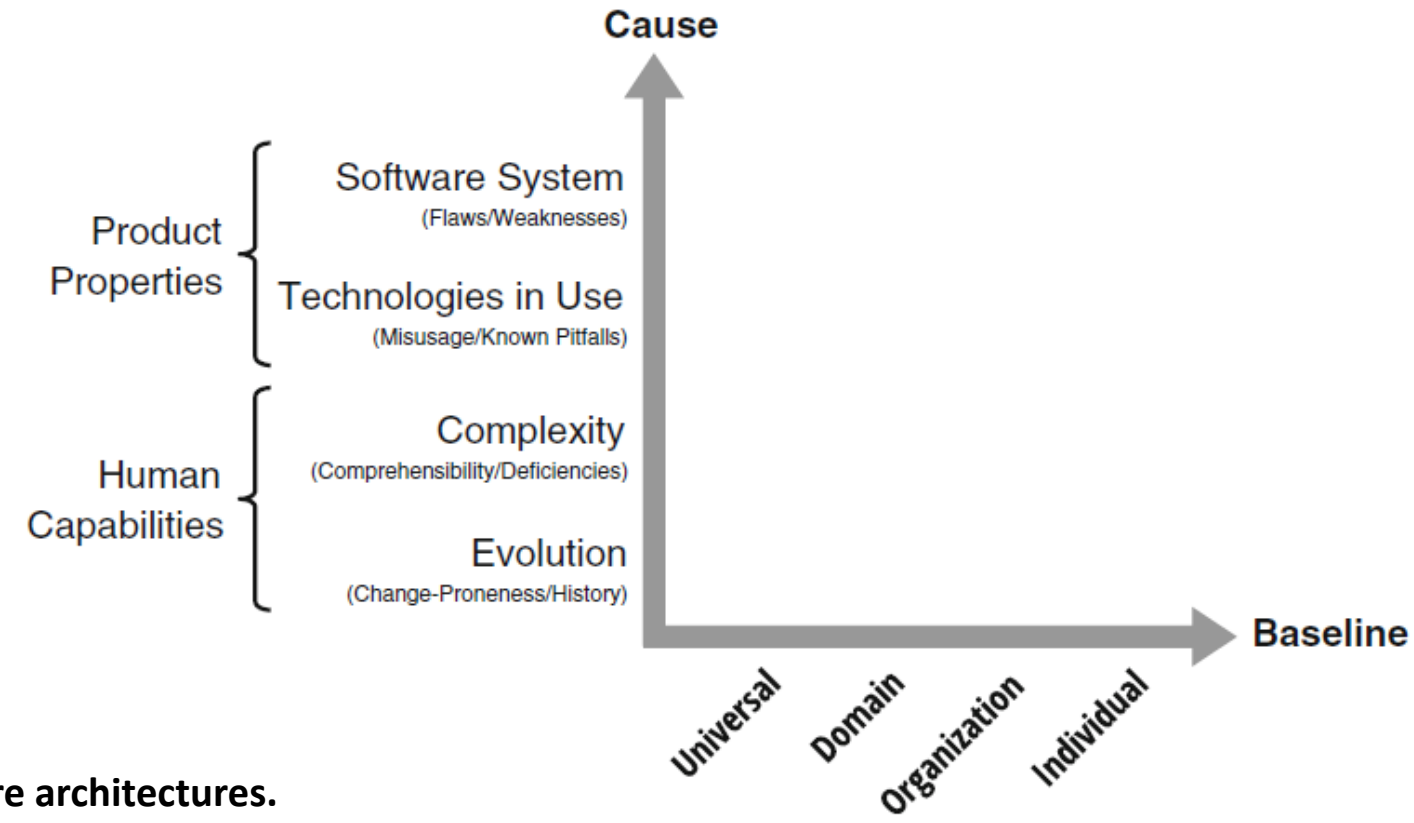
What are the outcomes of architectural evaluation?

Outcomes of architectural evaluation

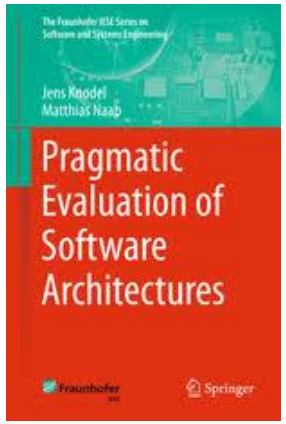
- An architectural evaluation does not tell “yes or no”, “good or bad” or a numeric score, but it rather tells what and where are the risks
- An architectural evaluation may reveal positive and negative findings (e.g., risks identified, assumptions made, scaling factors, limitations, trade-offs, violations, etc.) about both the system quality and the artifact quality
  - The interpretation of the findings is context-dependent
  - Negative findings represent the starting point towards deriving improvement actions

Outcomes of architectural evaluation

Nature of findings



© Fraunhofer IESE



Jens Knodel, Matthias Naab.
Pragmatic evaluation of software architectures.
Switzerland: Springer International Publishing, 2016

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

The moment to perform an architectural evaluation

- Before and during architectural analysis and synthesis
- Right after making decisions or when comparing and selecting architecture alternatives
- At certain milestones
- During implementation, mainly for checking compliance between the intended architecture and the implemented one
- Before making major decisions about the acquisition, migration, or retirement of components or whole systems
- During larger maintenance, integration, or migration projects needing architectural guidance

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

Common mistakes in practice

- Having no idea about architectural evaluation at all, thus hampering its usefulness as an instrument for quality assurance
- Having no clear goals for an architectural evaluation
- Not being able to engage in an architectural evaluation
- Not having the skills or the right people to execute an architectural evaluation

Common mistakes in practice

- Focusing only on metrics (especially supported by tools) that are indeed indicators of potential problems, but not more than that
- Evaluating only a small set of artifacts, e.g. architecture descriptions
- Having no systematic architectural evaluation approach
- Inadequately interpreting the evaluation results
- Doing no architectural evaluation at all

Que erros são frequentemente cometidos na prática?

Objetivos

What are the limitations of architectural evaluation?

What are the aims of architectural evaluation?

Why evaluating software architectures is important?

What mistakes are frequently made in practice?

When should an architectural evaluation be conducted?

What are the outcomes of architectural evaluation?

Limitations of architectural evaluation

- Its absoluteness is typically not possible to achieve
- It results in a set of findings requiring interpretation, and such an interpretation may fail due to human beings
- It requires cooperation, i.e., the person in charge of it cannot perform it alone
- It cannot guarantee quality by itself
- It does not often consider multiple quality attributes and trade-offs among them
- It does not have “the gold method” so far


Nothing is Perfect
matankids.org

Architectural evaluation methods

- There is a large body of proposals on how to evaluate software architectures for various concerns
  - The usefulness of having systematic evaluation approaches is acknowledged by both academia and industry
- There is some sort of tendency towards focusing on evaluation methods that can simultaneously address several quality attributes and possible trade-offs among them

Architectural evaluation methods

Informal approaches rely on the analysis of architectural models by human stakeholders for specific properties, but they require manual effort and significant experience of the evaluation team

- Experience-based evaluations are based on the previous experience and domain knowledge of developers or consultants
- Scenario-based evaluations try to evaluate a particular quality attribute by creating a scenario depicting an interaction of one of the stakeholders with the system as well as addressing a particular quality

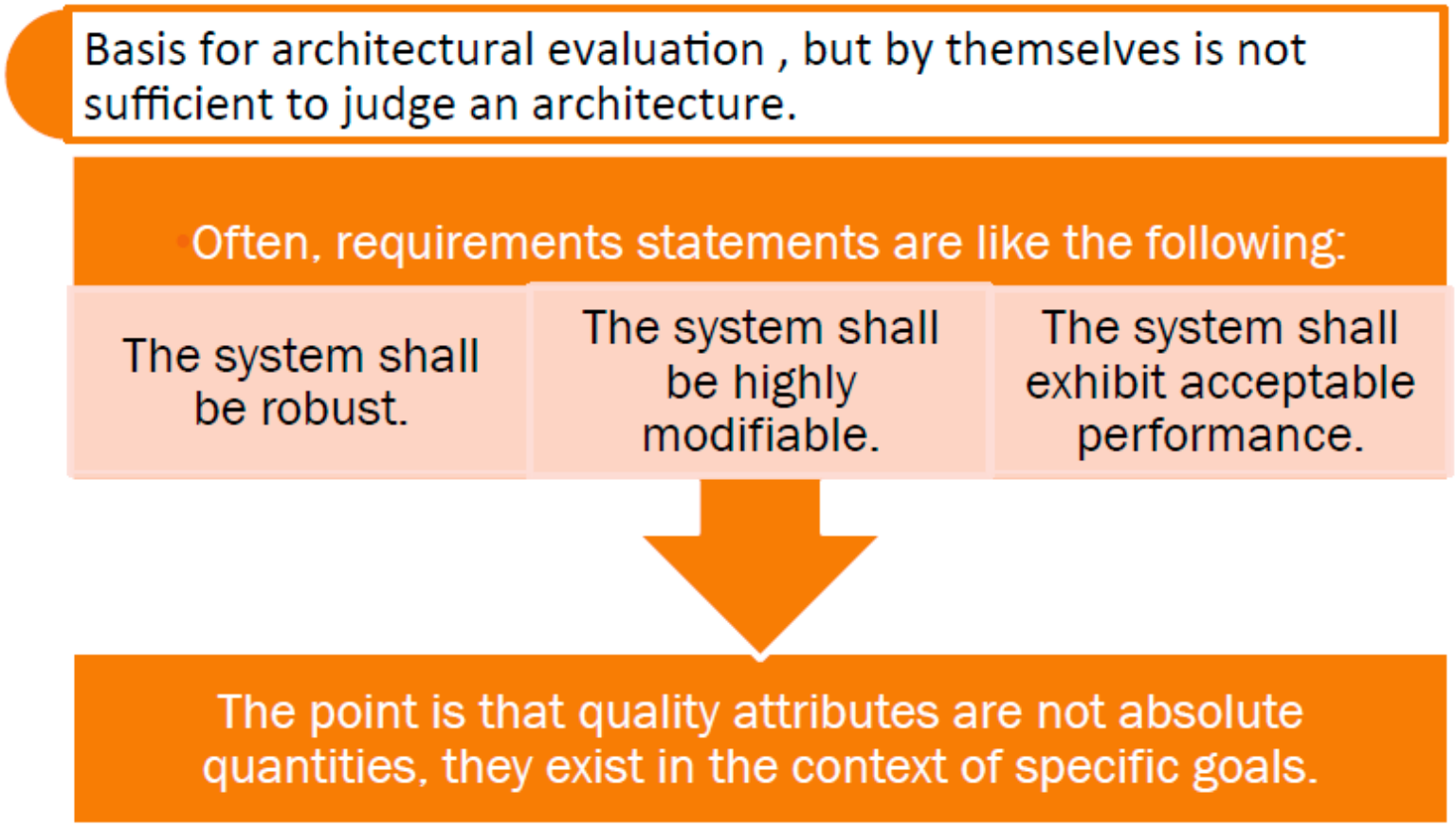
Architectural evaluation methods

Formalized automated approaches are objective and require specific architectural models, even though only specific quality concerns are addressed

- Simulation-based evaluations rely on a high-level implementation of some or all the components in the software architecture and its environment
- Mathematical modelling uses mathematical proofs and methods for evaluating mainly operational quality requirements such as performance and reliability

Architectural evaluation methods

Solely considering quality requirements is not enough as they may be vague, incomplete or ambiguous



SAAM – Software Architecture Analysis Method

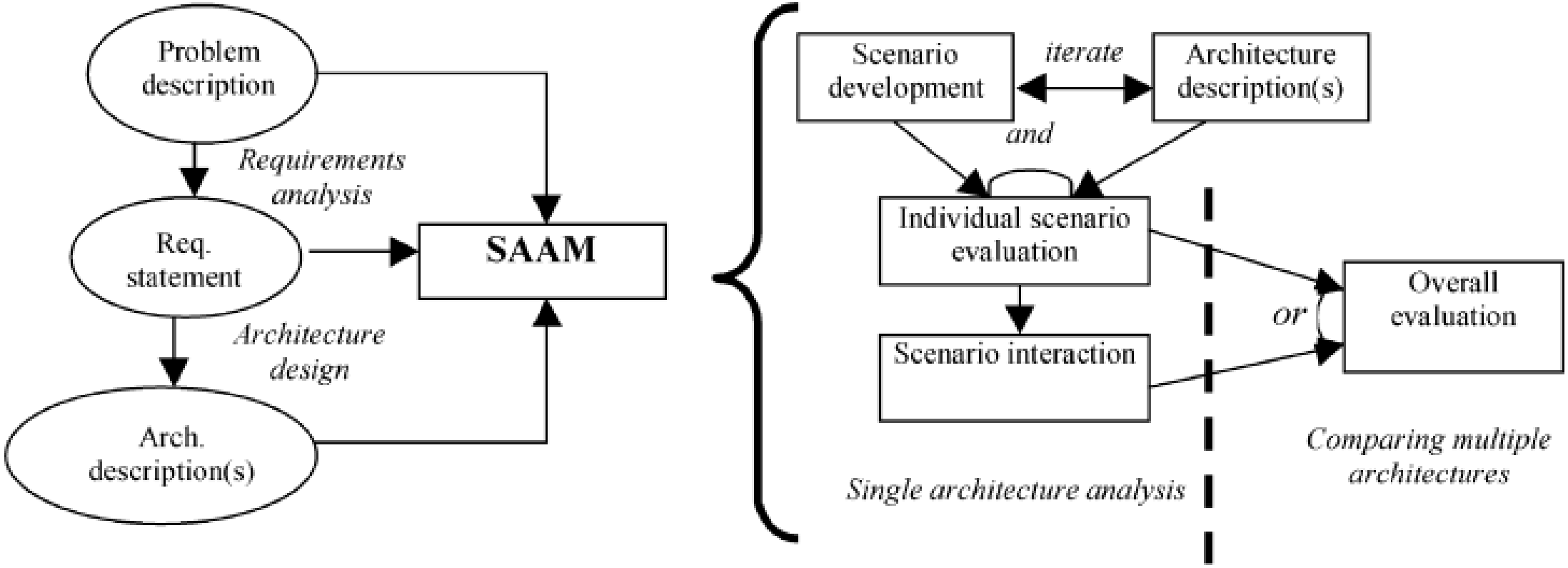
It is a scenario-based software architecture evaluation method targeted for evaluating a single architecture or making several architectures comparable using metrics such as coupling between architecture components

SAAM: A Method for Analyzing the Properties of Software Architectures

Rick Kazman — Department of Computer Science, University of Waterloo
Len Bass, Gregory Abowd — Software Engineering Institute, Carnegie Mellon University
Mike Webb — Texas Instruments Inc., Dallas, TX, U.S.A. 75265

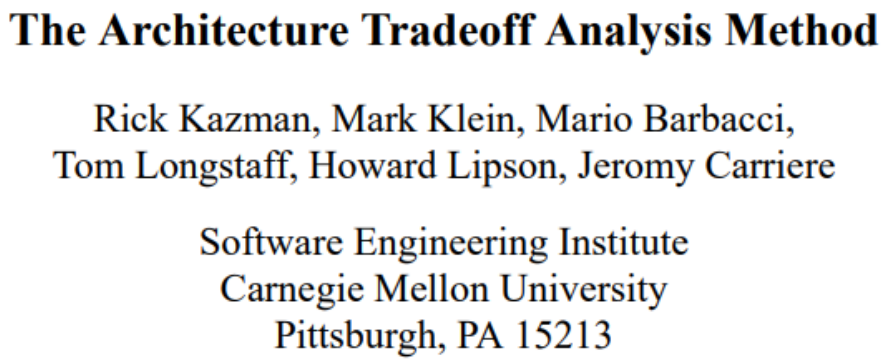
Rick Kazman, Len Bass, Gregory Abowd, Mike Webb. **SAAM: A method for analyzing the properties of software architectures.** Proceedings of the 16th International Conference on Software Engineering (ICSE 1994), Sorrento, Italy. USA: IEEE Computer Society, 1994, pp. 81-90

SAAM – Software Architecture Analysis Method



ATAM – Architecture Trade-off Analysis Method

It is a scenario-based software architecture evaluation method aimed to evaluate architecture-level designs with multiple quality attributes



The Architecture Tradeoff Analysis Method

Rick Kazman, Mark Klein, Mario Barbacci,
Tom Longstaff, Howard Lipson, Jeromy Carriere

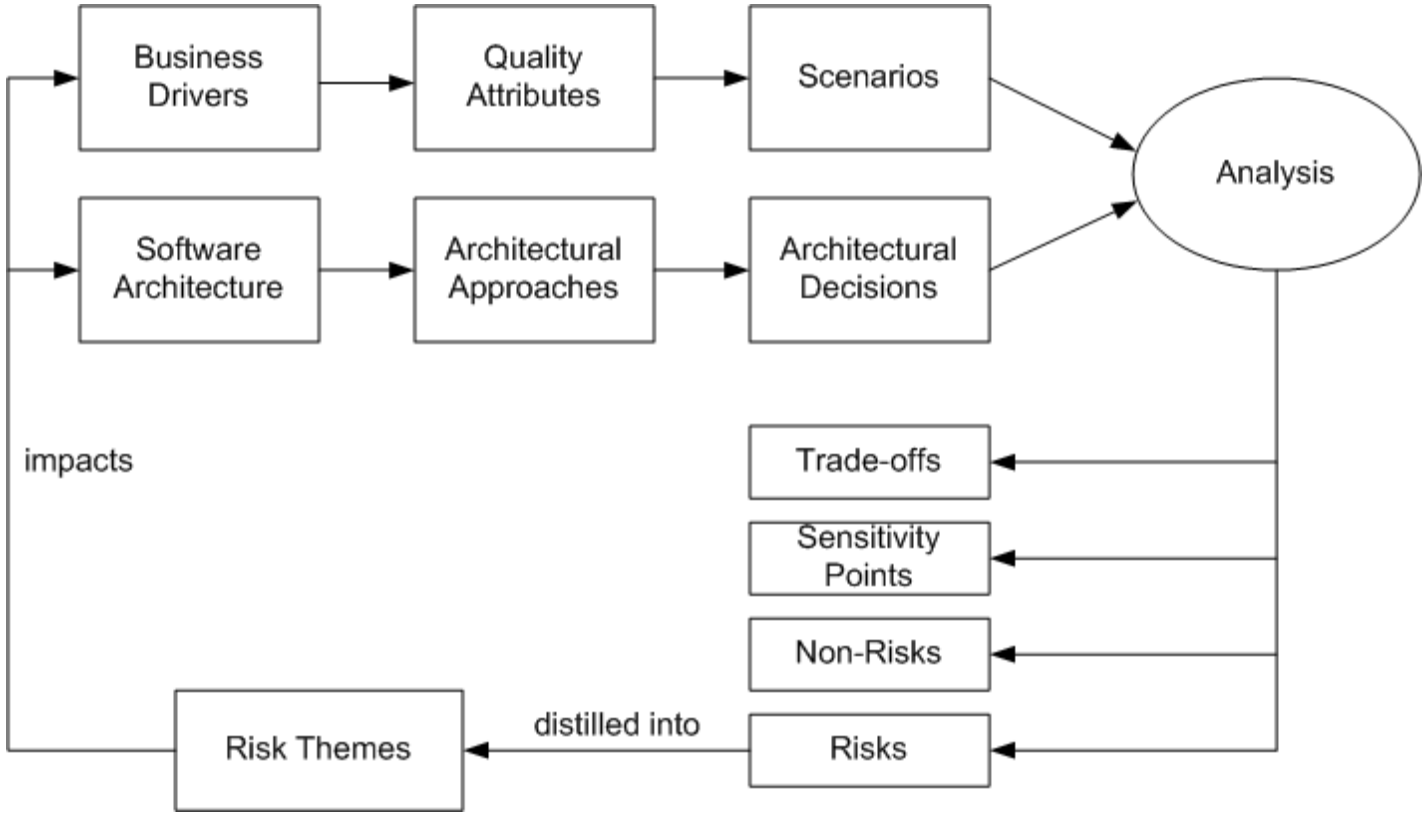
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Rick Kazman, Mark Klein, Mario Barbacci, Tom Longstaff, Howard Lipson, Jeromy Carriere.
The Architecture Tradeoff Analysis Method. Proceedings of the 4th International Conference on Engineering of Complex Computer Systems (ICECCS 1998), Monterey, CA, USA. USA: IEEE Computer Society, 1998, pp. 68-78

ATAM – Architecture Trade-off Analysis Method

It results in

- prioritized quality attribute requirements expressed as quality attribute scenarios
- a set of risks and non-risks
- a mapping of architectural decisions to quality requirements
- a set of identified sensitivity and trade-off points regarding quality attributes



DCAR – Decision-Centric Architecture Review

It is a method with focus on determining the soundness of architectural decisions

- The evaluation starts when stakeholders select a set of decisions to analyze in the context of relevant project- and company-specific decision forces
- As its main drawbacks, the concept of decisions is often not used in industry and it does not take future concerns into account


Decision-Centric Architecture Reviews

Uwe van Heesch, Capgemini Germany

Veli-Pekka Eloranta, Tampere University of Technology

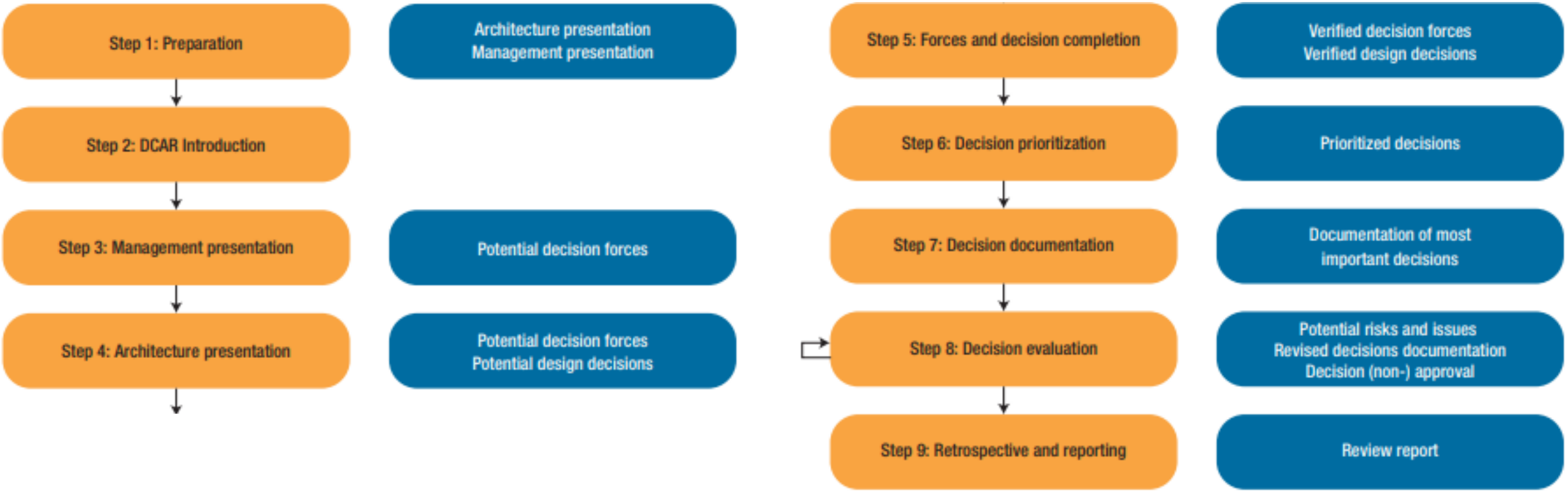
Paris Avgeriou, University of Groningen

Kai Koskimies, Tampere University of Technology

Neil Harrison, Utah Valley University

Uwe van Heesch, Veli-Pekka Eloranta, Paris Avgeriou, Kai Koskimies, Neil Harrison. **Decision-centric architecture reviews.** IEEE Software, vol. 31, no. 1, January/February 2014, pp. 69-76

DCAR – Decision-Centric Architecture Review



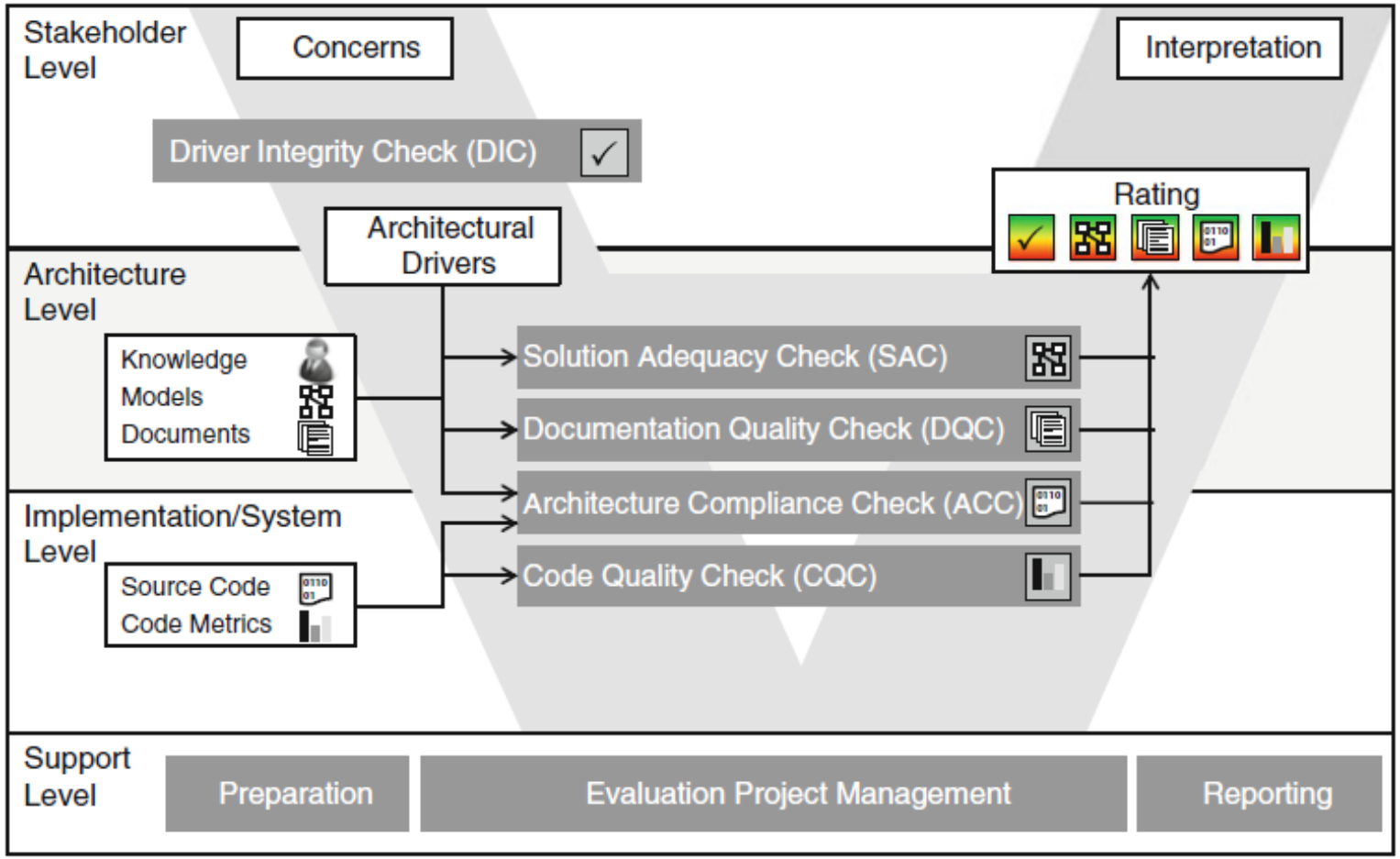
© Uwe van Heesch et al.

RATE – Rapid ArchiTecture Evaluation

- It is a compilation and collection of best practices of existing evaluation approaches tailored towards pragmatic (or rapid) application in industry
- It comprises five checks, with each check serving a distinct purpose, but all of them following the same principle of work: to reveal findings aimed at confirming/improving the system's quality and/or the artifact's quality

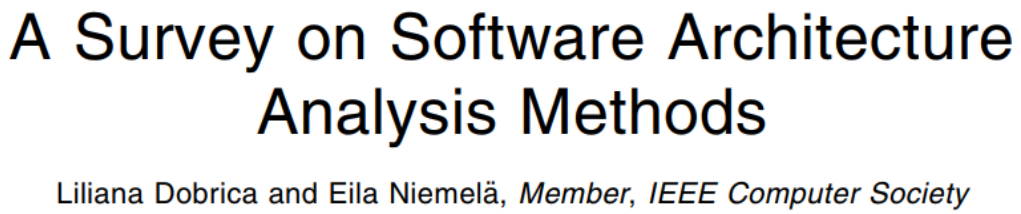
Jens Knodel, Matthias Naab. **Pragmatic evaluation of software architectures.** Switzerland: Springer International Publishing, 2016

RATE – Rapid ArchiTecture Evaluation



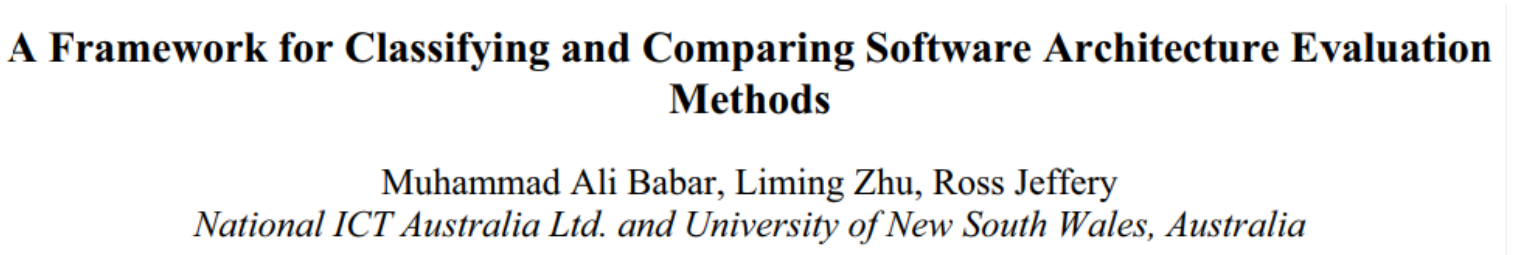
Architectural evaluation methods

How to choose?


A Survey on Software Architecture Analysis Methods

Liliana Dobrica and Eila Niemelä, *Member, IEEE Computer Society*

Liliana Dobrica, Eila Niemelä. **A survey on software architecture analysis methods.** IEEE Transactions on Software Engineering, vol. 28, no. 7, July 2002, pp. 638-653


A Framework for Classifying and Comparing Software Architecture Evaluation Methods

Muhammad Ali Babar, Liming Zhu, Ross Jeffery
National ICT Australia Ltd. and University of New South Wales, Australia

Muhammad Ali Babar, Liming Zhu, Ross Jeffery. **A framework for classifying and comparing software architecture evaluation methods.** Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), Melbourne, VIC, Australia. USA: IEEE Computer Society, 2004, pp. 309-318

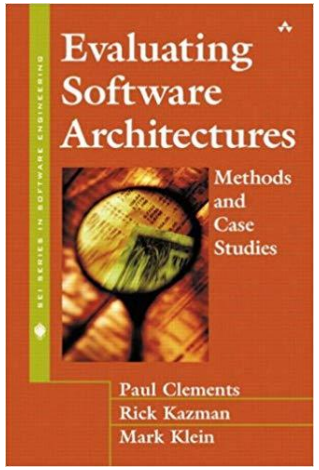

LESSONS
LEARNED

The takeaways of architectural evaluation

- Despite being quite important, architectural evaluation is neither easy nor cheap
  - The benefits typically far outweigh the drawbacks
- Early information about the system's key characteristics is indispensable
- How much evaluation?
  - Too many will expend resources unnecessarily
  - Too few will carry the risk of propagating defects into the final system
  - Wrong evaluations will have both drawbacks

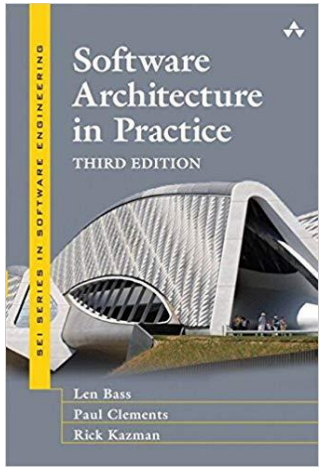
Further reading

Jens Knodel, Matthias Naab. Pragmatic evaluation of software architectures. Switzerland: Springer International Publishing, 2016



Paul Clements, Rick Kazman, Mark Klein. Evaluating software architectures: Methods and case studies. USA: Addison-Wesley, 2002

Further reading



Len Bass, Paul Clements, Rick Kazman.
Software architecture in practice – 3rd ed.
USA: Addison-Wesley/Pearson Education, Inc., 2013

Architectural evaluation

Prof. Dr. Everton Cavalcante

http://www.dimap.ufrn.br/~everton/