

PLM and Innovation  
Excellence

Learning Campus

Your partner for  
Business Learning

Siemens  
Core  
Learning  
Program

# Architecture Quality and Architecture Reviews

Authors: Wieland Eckert, HC | Christa Schwanninger, HC |  
Peter Zimmerer, CT

## Reviews: Technical and interpersonal challenge

Although the architect is absolutely central, this does not mean that the architect does not need tools, techniques and mechanisms to turn aspects of the job from art to science or, preferably, engineering.

[Jan Bosch, 2000]



The notion of collaboration is important: to work together on a joint intellectual effort. Architecture reviewers should be collegial, should trust and share information. If review is in collaborative environment, there's little friction, there's mutual respect, and there's give and take.

[Rebecca Wirfs-Brock, 2009]



# Architecture quality and architecture reviews

## Learning objectives

- Know how architecture reviews can help improve quality
- Know quantitative and qualitative methods for reviewing architectures
- Know when and why to conduct architecture reviews

# Architecture Quality and Architecture Reviews

## Agenda

### Architecture Quality

Quantitative Reviews

Qualitative Reviews

Experience-based Reviews

Scenario-based Reviews

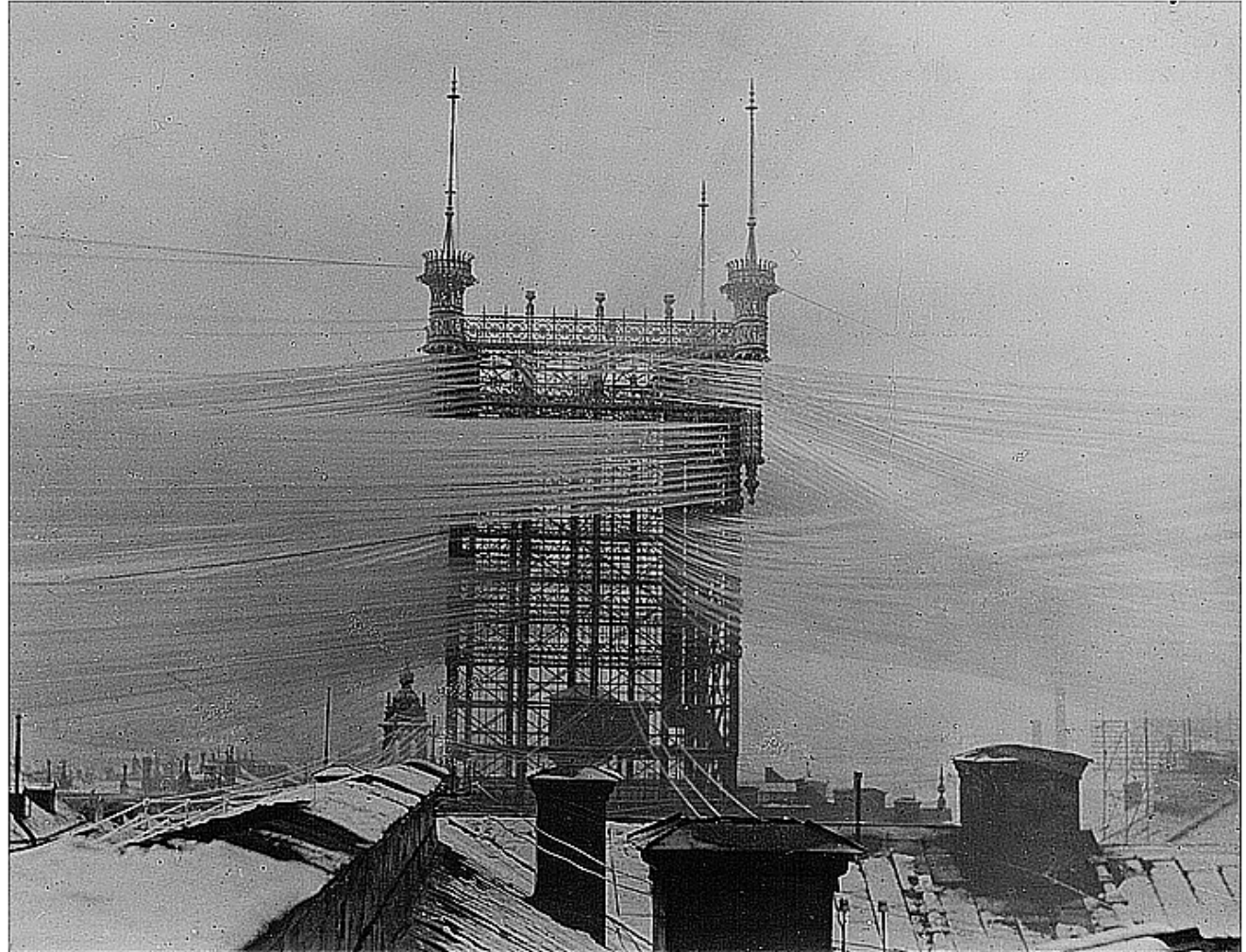
Summary

## Architecture Quality?

Stockholm Telefontorget  
ca 1890

Central telephone exchange  
Capacity ~5000 phone lines

Completed	1887
Renovated	1890
Decommissioned	1913
Destroyed	1953



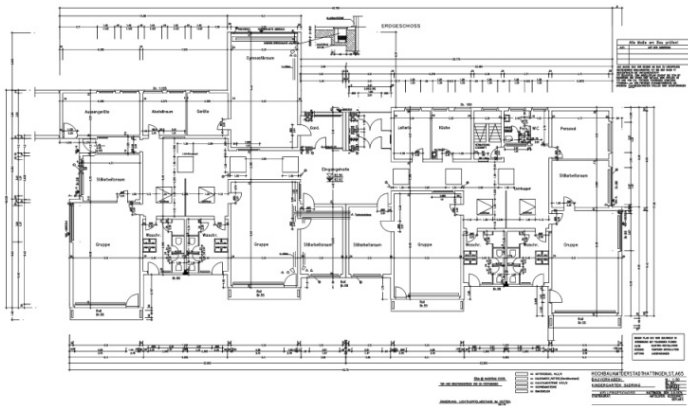
Source: <http://www.thisiscolossal.com/2014/09/telefontorget-stockholm/>

## The architect's role in architecture quality assurance

The architect is responsible for designing the architecture, but also for assessing and maintaining it.

He needs techniques to answer the questions:

- Is the architecture good enough?
- Is the architecture implemented correctly?  
(including 80-20 rule)



**Your responsibility!**





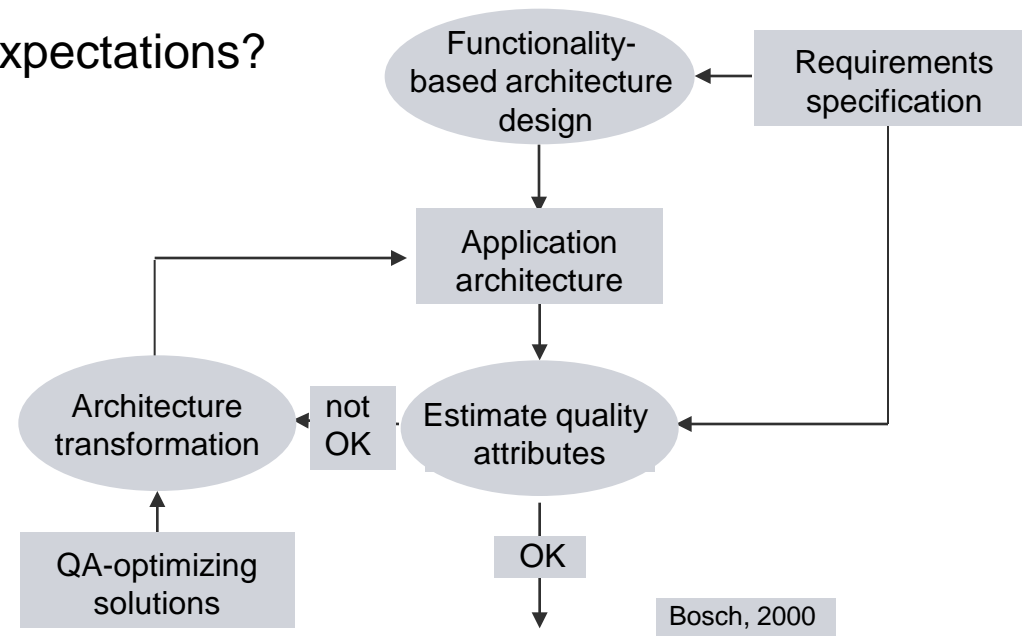
## Initial architecture quality assurance

Does the architecture fulfill all functional requirements?

- Assignment of responsibilities to test system components

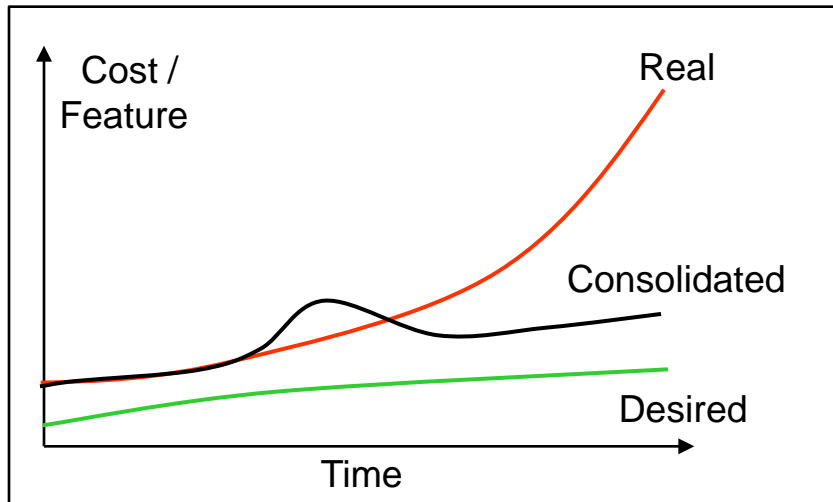
Does the architecture meet the quality expectations?

- Clarification of quality goals
- Agreement on priorities among qualities
- Comparison of architectural options
- Verification of tradeoffs
- Early identification of technical risks
- Support for life cycle issues available?

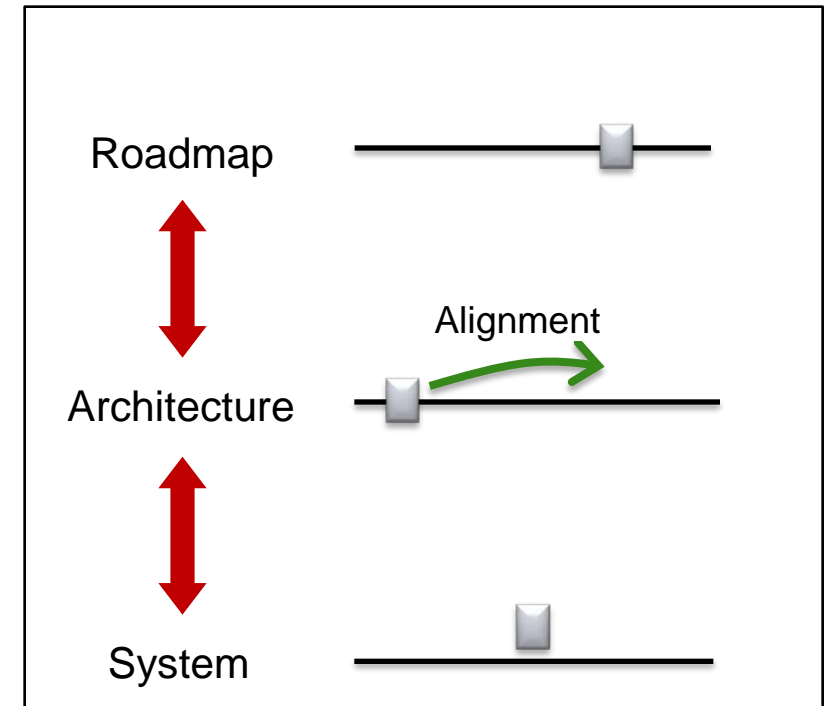


## Ongoing architecture quality assurance

Over its lifetime, architecture has to cope with changes.



Architecture reviews are the first step for aligning an architecture with changing business goals.



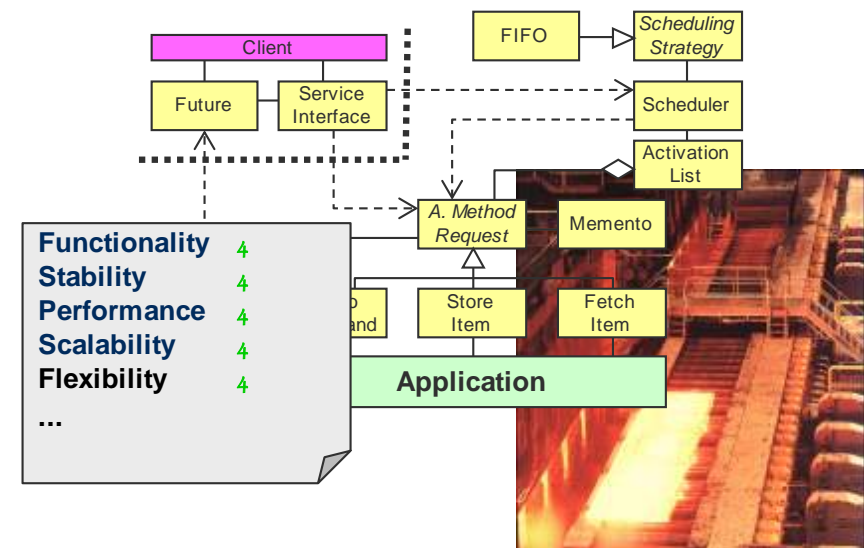
Management needs facts to decide about investment in an architecture.



## Architecture and code reviews

Architecture and code reviews are a retrospective approach to assess the quality of a software architecture and its implementation

- Reviews are an effective tool for enforcing an architecture vision and controlling its feasibility
- Reviews should be performed regularly but not too often: conducting a review can be effort-intensive
- Use tool support to automate and assist architecture and code reviews



Perform reviews for platform and product line engineering from the perspective of usability in realizing the key products to be supported, because a platform does not provide end-to-end functionality and a product line architecture is not executable out of the box

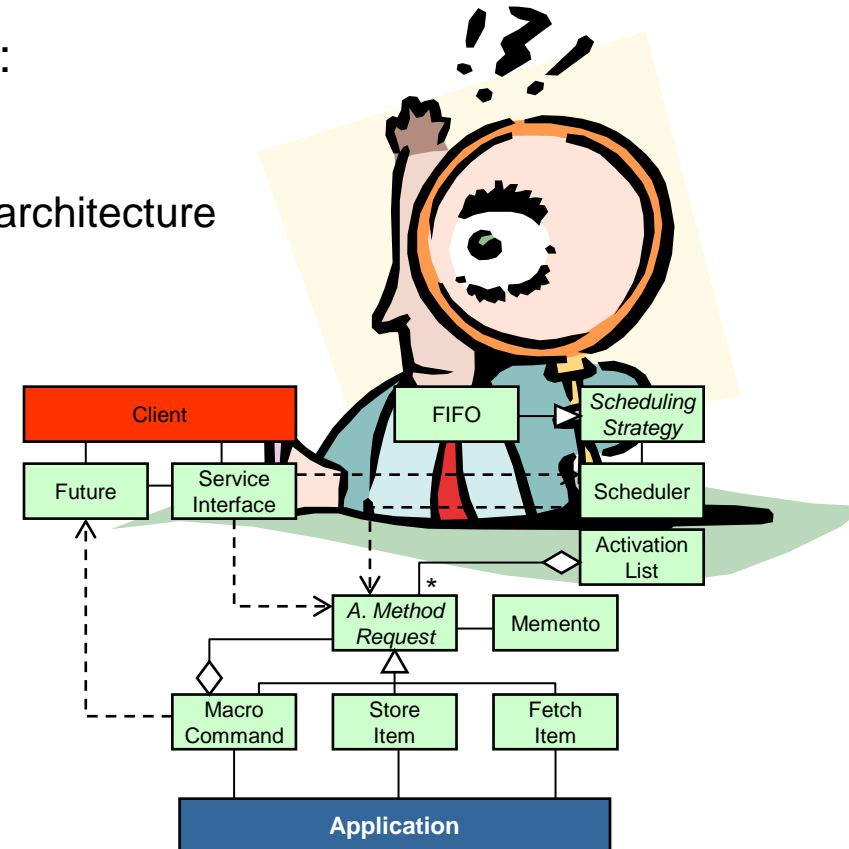


## Architecture reviews follow a defined structure

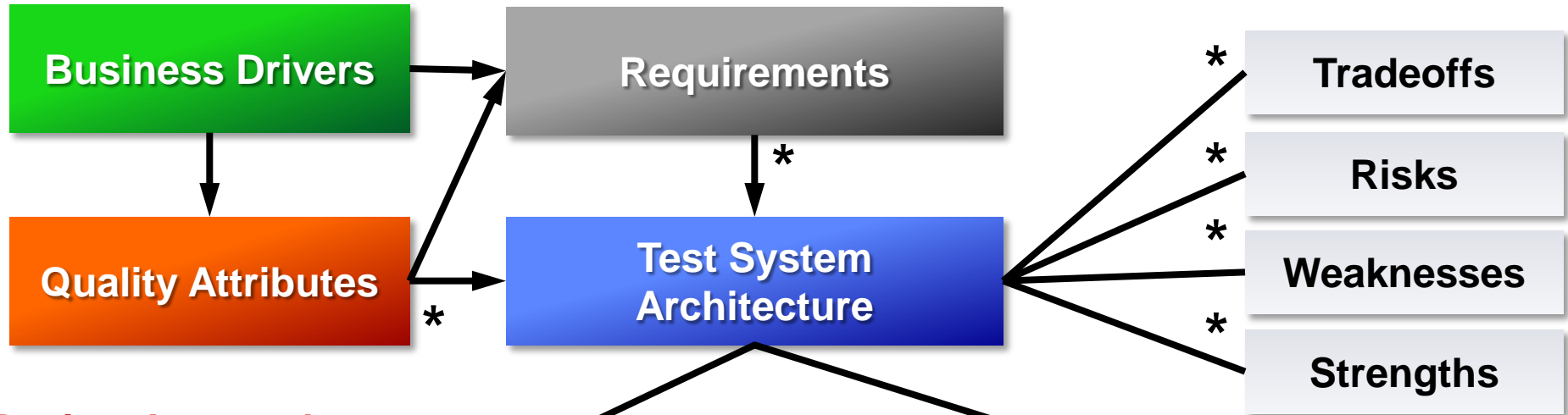
A proper architecture review comprises four steps:

- **Scoping:** what is the review all about?
- **Collection:** collect and retrieve information about the architecture with an emphasis on the review's focus.
- **Evaluation:** how well meets the architecture the issues of interest. If it does not, how can it be improved so that it gets back on track?
- **Feedback:** report the evaluation results back to the customer and the development team.

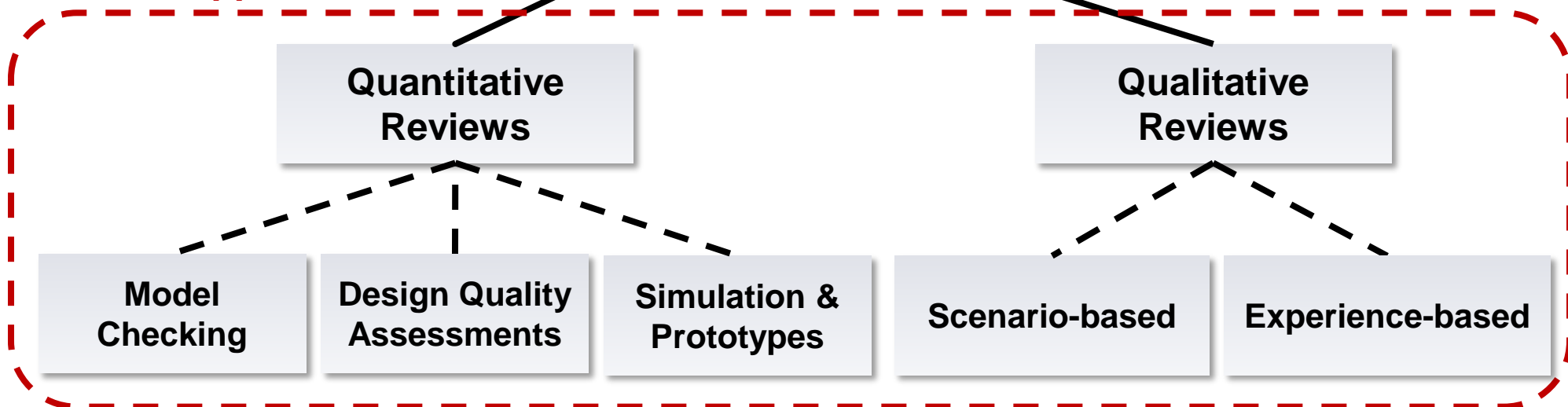
There are many different concrete review methods, such as ATAM and SAAM, but all methods follow this basic structure



## System Architecture Reviews



## Review Approaches



## Architecture Quality for TeA?

Why do we deal with architectural quality?

- Remember “spaghetti code”  
*Why is this bad?*
- A device is constructed with 17 different types of bolts.  
Same device could be constructed with 3 different types of bolts.  
*Why is this better?*
- In order to exchange the clutch in a car, you have to remove the engine first.  
*Is this good or bad design?*

Measurements, e.g.

- Cohesion of components
- Coupling of components
- Complexity of components
- “The time it takes to comprehend the fundamental architecture” ...

**Its all about operational and developmental qualities!**

For example: learnability, maintainability, modularity, modifiability, extensibility, ...

## Architecture review of your test system

Developmental qualities of the test system are **YOUR** responsibility!

Consult your product architects, developers, testers, test managers to review

- Compatibility of external interfaces (e.g. unit test mockup)
- Coverage of important test cases of the product (e.g. performance testing)
- Long term evolution of test system architecture (e.g. incremental integration)

TeA should also participate in architecture reviews of the product

- Gain insight in the product architecture
- Demonstrate that you are a *partner* and share the responsibility

Architecture are a great way of communication, too!

Your reviewers will learn about your architecture

# Architecture Quality and Architecture Reviews

## Agenda

Architecture Quality

**Quantitative Reviews**

Qualitative Reviews

Experience-based Reviews

Scenario-based Reviews

Summary

## Types of quantitative review

### Model Checking

- Analysis on the (test) system models: completeness, constraints, estimations (e.g. performance, cost, resources, MTBF, MTTR, FTA, FMEA)

### Design quality assessment

- Structure analysis, architecture conformance/violations, graph analysis and design structure matrix (complexity, dependencies), bill-of-materials analysis (cost, number of components), mechanical tolerances, change and impact analysis (see session Internal Quality in workshop 3)

### Simulations and Prototypes

- See session Test Environment in workshop 2

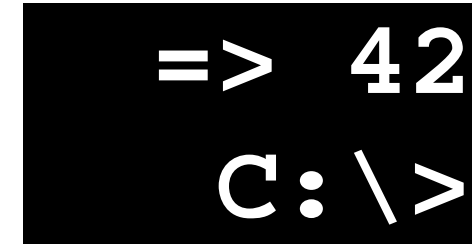
MTBF:	Mean Time Between Failures
MTTR :	Mean Time to Repair
FTA :	Failure Tree Analysis
FMEA:	Failure Mode and Effect Analysis



## Quantitative review

### Benefits

- Yield "hard" results
- Quantifiable, objective means for selecting alternatives
- Experiments by altering the parameters relatively easy



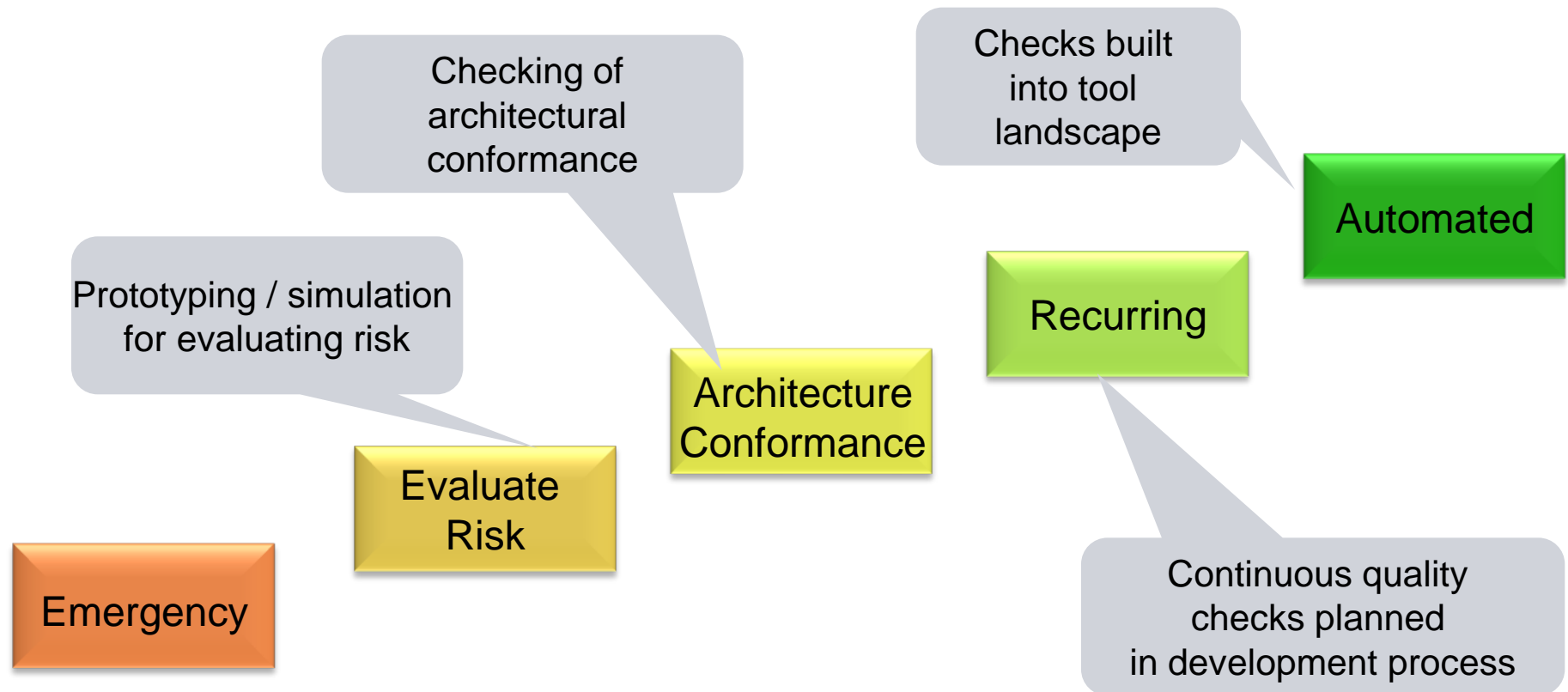
### Liabilities

- Focus on only a couple of concerns or (test) system parts
- Works only if data is interpreted correctly
- Effect on quality attributes other than the focus is unknown

Similar to test automation, the initial cost might be high, but is typically justified by early detection of conceptual faults.



## Maturity levels for quantitative reviews



# Architecture Quality and Architecture Reviews

## Agenda

Architecture Quality

Quantitative Reviews

**Qualitative Reviews**

Experience-based Reviews

Scenario-based Reviews

Summary

## Qualitative reviews

### Some general remarks:

- The main goal is to **identify** problems and risks, not to solve them
  - Solutions are the responsibility of the architects and development team
  - In some cases, suggestions and technical advice can be helpful
- Evaluate the architecture, not the people
- Involve the relevant stakeholders for the given aspect(s)
  - Refer to your Stakeholder Analysis (Workshop 1)
- Provide checklists and templates when possible
  - Use the “lessons learned” from previous projects as inspiration

## Qualitative review

### Benefits

- Involves all relevant stakeholders
- Overview of the whole (test) system
- Improve understanding for all participants
- Relatively cheap to execute
- Can be conducted as soon as high-level architecture design is available

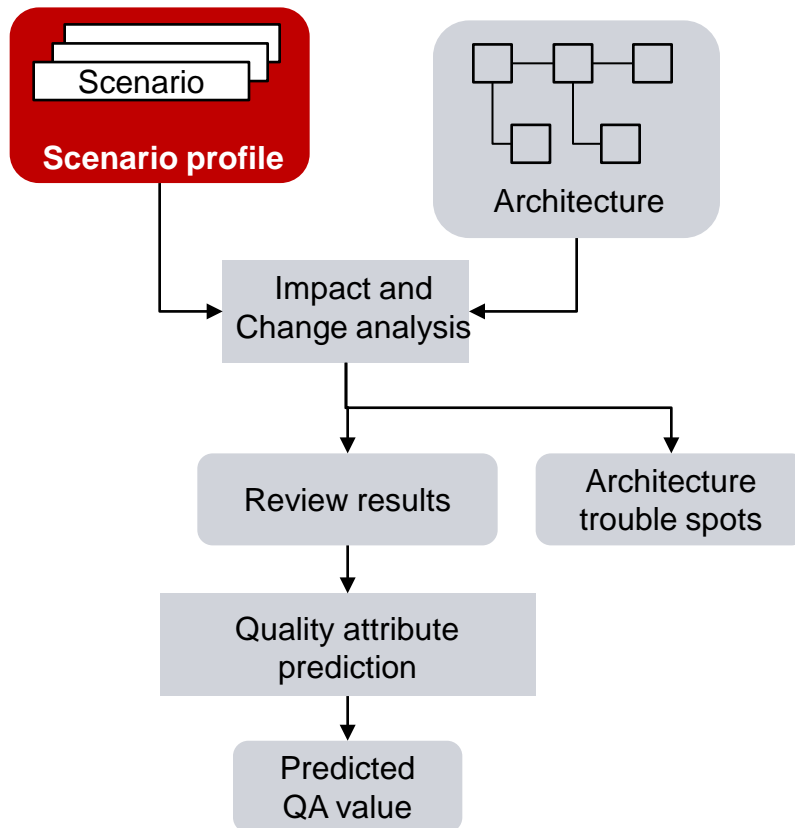


### Liabilities

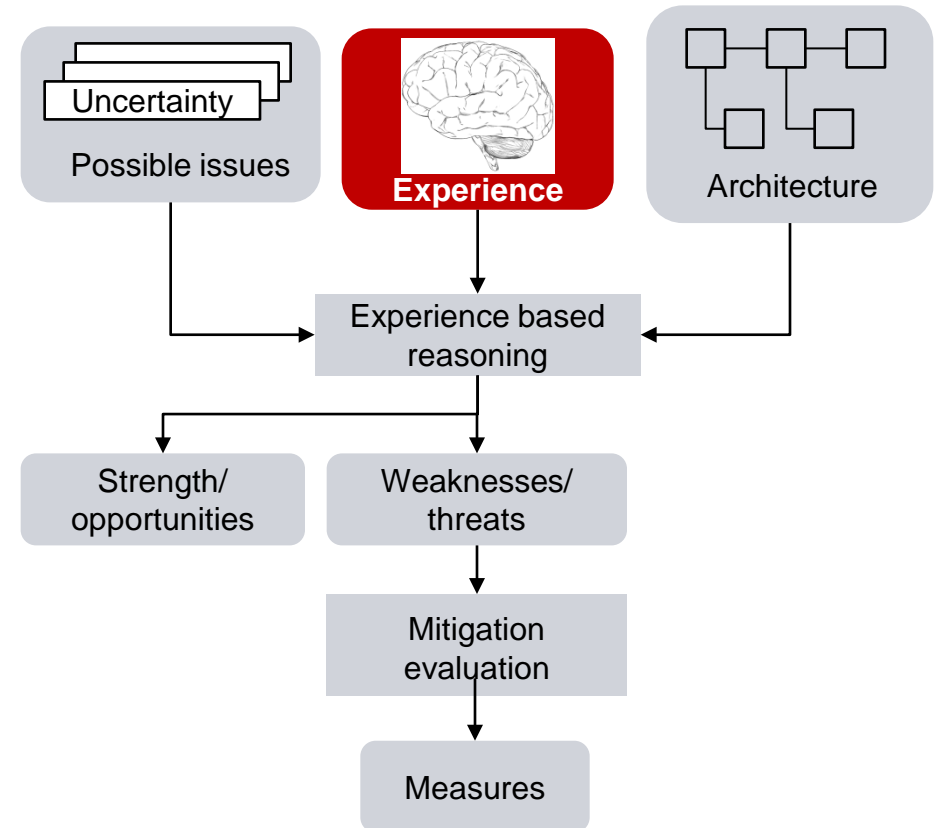
- Relies mainly on documents and statements from personally involved stakeholders
- Experienced reviewers required
- No "hard facts" (unless supported by quantitative reviews)

# Types of qualitative reviews

## Scenario-based

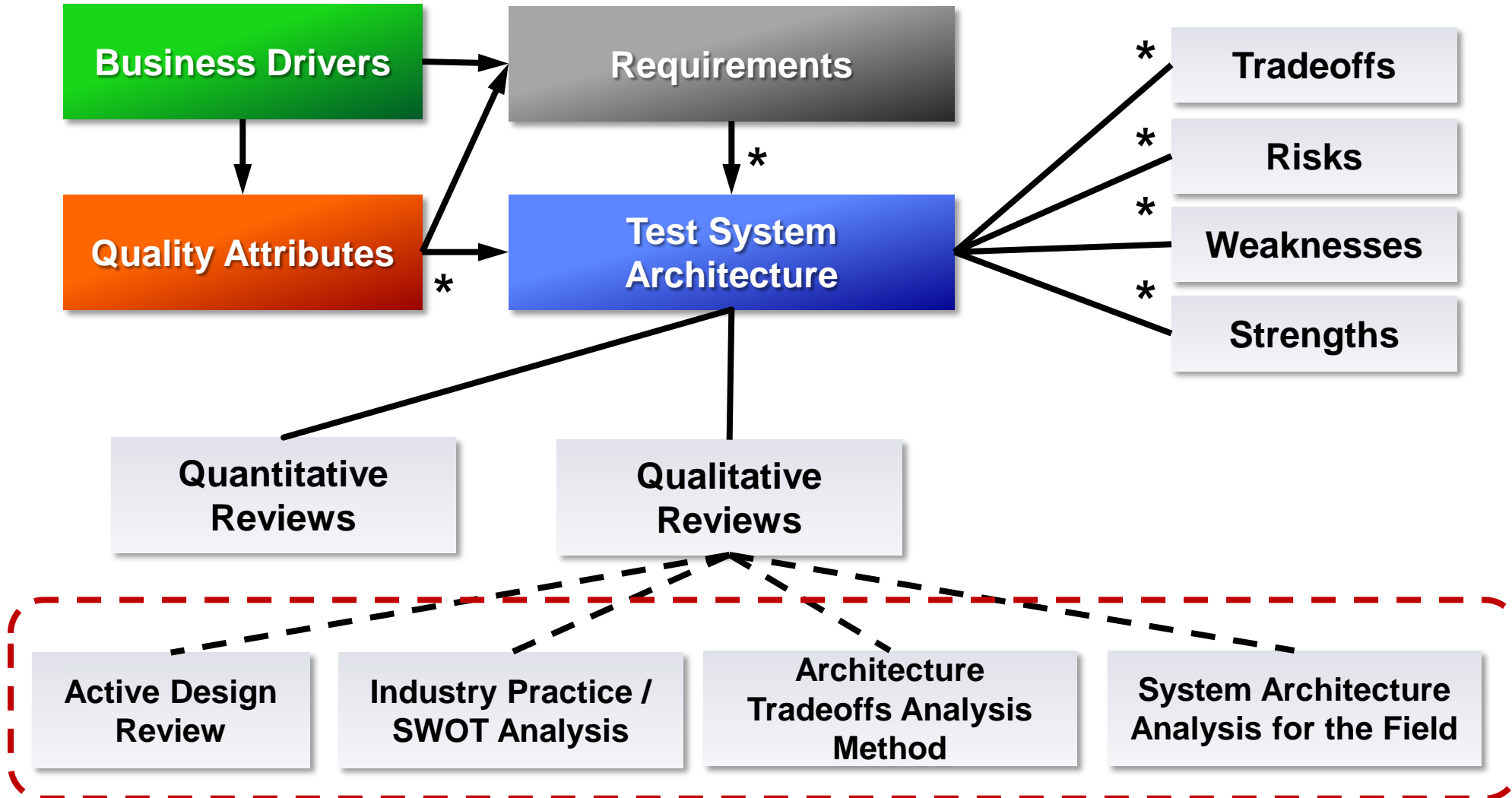


## Experience-based



Bosch, 2000

## Qualitative Review Toolbox





# Qualitative Review Toolbox

	Active design review	Industry practice	Architecture Tradeoff Analysis Method	System Architecture Analysis for the Field
Type	Experience-based, scenario-based	Experience-based	Scenario-based	Scenario-based
Intention	Improve design, find errors	SWOT analysis, identify measures	Clarify and prioritize requirements, evaluate suitability of architecture for change scenarios	Improve design, find errors, identify measures for mitigating typical lifecycle issues

## Typical phases of reviews

1. **Preparation** – reviewer and customer – discussion
  - Clarification of concrete, project-specific review goals and questions
2. **Collection** – reviewer and interviewees – interview
  - Interviews with architects, developers and stakeholders
  - Analysis of documents and source code
  - Demonstration of the running software, ...
3. **Elaboration** – reviewer – documentation and analysis
  - SWOT analysis of the software architecture: Strengths, weaknesses, opportunities, threats
  - Measures for dealing with weaknesses and threats
4. **Consolidation** – reviewer and customer – clarification
  - Clarification and consolidation of the final report with key stakeholders
5. **Presentation** – reviewer – presentation
  - Presentation of review results to stakeholders;  
constructive view on the potential for improvement
6. **Optional: workshop** – all stakeholders – discussion
  - Joint discussion of results and measures with stakeholders;  
development of concrete improvement scenarios

## Error classification

Error type	Examples
<b>Inconsistencies</b>	<ul style="list-style-type: none"> <li>▪ Places where the design will not function properly</li> <li>▪ Contradicting behavior</li> </ul>
<b>Inefficiencies</b>	<ul style="list-style-type: none"> <li>▪ Unnecessary complexity</li> <li>▪ Inefficient use of resources, channels, etc</li> </ul>
<b>Ambiguities</b>	<ul style="list-style-type: none"> <li>▪ Unclear requirements</li> <li>▪ Undocumented assumptions</li> <li>▪ Elements of the design specification which can be understood in different ways</li> </ul>
<b>Inflexibility</b>	<ul style="list-style-type: none"> <li>▪ Elements making change requests difficult</li> <li>▪ Obstacles for dealing with lifecycle issues (e.g. configuration, service, upgrade, etc.)</li> </ul>

# Overview of qualitative architecture reviews

	SAAM	ATAM	ADR	Industry practice
Type	Scenario-based	Scenario-based	Experience-based, scenario-based	Experience-based
Intention	Clarify and prioritize requirements, evaluate suitability of architecture for change scenarios	Clarify and prioritize requirements, find risks, sensitivity points, tradeoffs	Improve design, find errors	SWOT analysis, identify measures
Interaction	Workshop	Workshop	Designer, reviewer	Interviews
Phase	Architecture design complete enough for walkthroughs	Architecture design complete enough for walkthroughs	Detailed component / module design ready	After architecture has been designed
Strength	Bring stakeholders together, requirement prioritization	Like SAAM, but deeper architectural evaluation	Focused on finding defects in design	Concrete measures
Key restriction	No risks, no measures	No measures	Small scale	No common understanding of requirement priorities
Duration	2–3 days	Two weeks	2 days / reviewer	Four weeks regular 1 day flash

# Architecture Quality and Architecture Reviews

## Agenda

Architecture Quality

Quantitative Reviews

Qualitative Reviews

**Experience-based Reviews**

Scenario-based Reviews

Summary

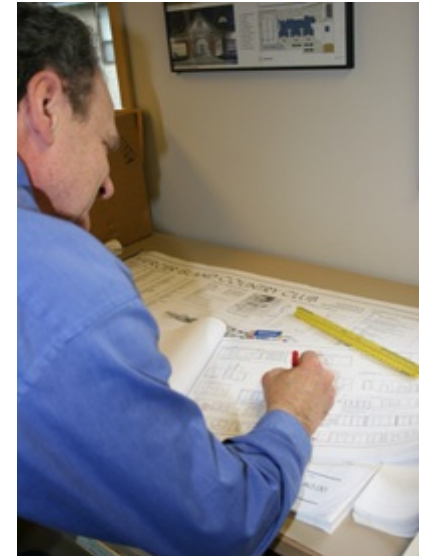
## Active Design Review (ADR)

**Purpose:** Test design and design documentation  
Review detailed designs for components / modules  
Scenario-based, designer asks reviewer to solve a concrete tasks  
Experience-based, designer and reviewer involved  
Tests the design and the documentation of the design  
Different reviewers for different fields of expertise

**Effort:** 2 days for each reviewer,  
1 day for designer per reviewer

**Results:** List of errors, improved design and design documentation

**Benefits:** Efficient, deep analysis, improved documentation, improved understanding



## ADR: Process steps

### ***Prepare design and documentation for review*** – designer

- Documentation should be sufficient to answer detailed questions:
  - Software: interfaces, data types, exceptions thrown, dynamic behavior
  - Systems: structure, interfaces, channels (e.g. information/material/energy flow), diagnostics, dynamic behavior

### ***Identify for specialized reviews*** – designer

- Identify critical design issues (interfaces to other (test) system parts) to be addressed

### ***Identify the reviewers needed*** – designer

- Specialists for different technical domains
- Clients of the design

### ***Design the questionnaires & checklists*** – designer

- Ask the reviewer to use the design in exercises that test their understanding

### ***Conduct the review*** – reviewer, designer

- Reviewers fill out the questionnaires using the design documentation
- Debriefing together with reviewer



## Industry Practice review method

**Purpose:** Confirm strength, find challenges and identify measures

Reviewers are experienced architects

System description by project externals

- Elaboration of the key requirements
- Elaboration of the key design elements

Analysis and documentation of strengths, weaknesses, opportunities, and threats

**Effort:** Regular review: Reviewer team 20–60 person days, project team 8–16 p.-days  
Flash review: Review team 2–3 days, project team 2–3 hours

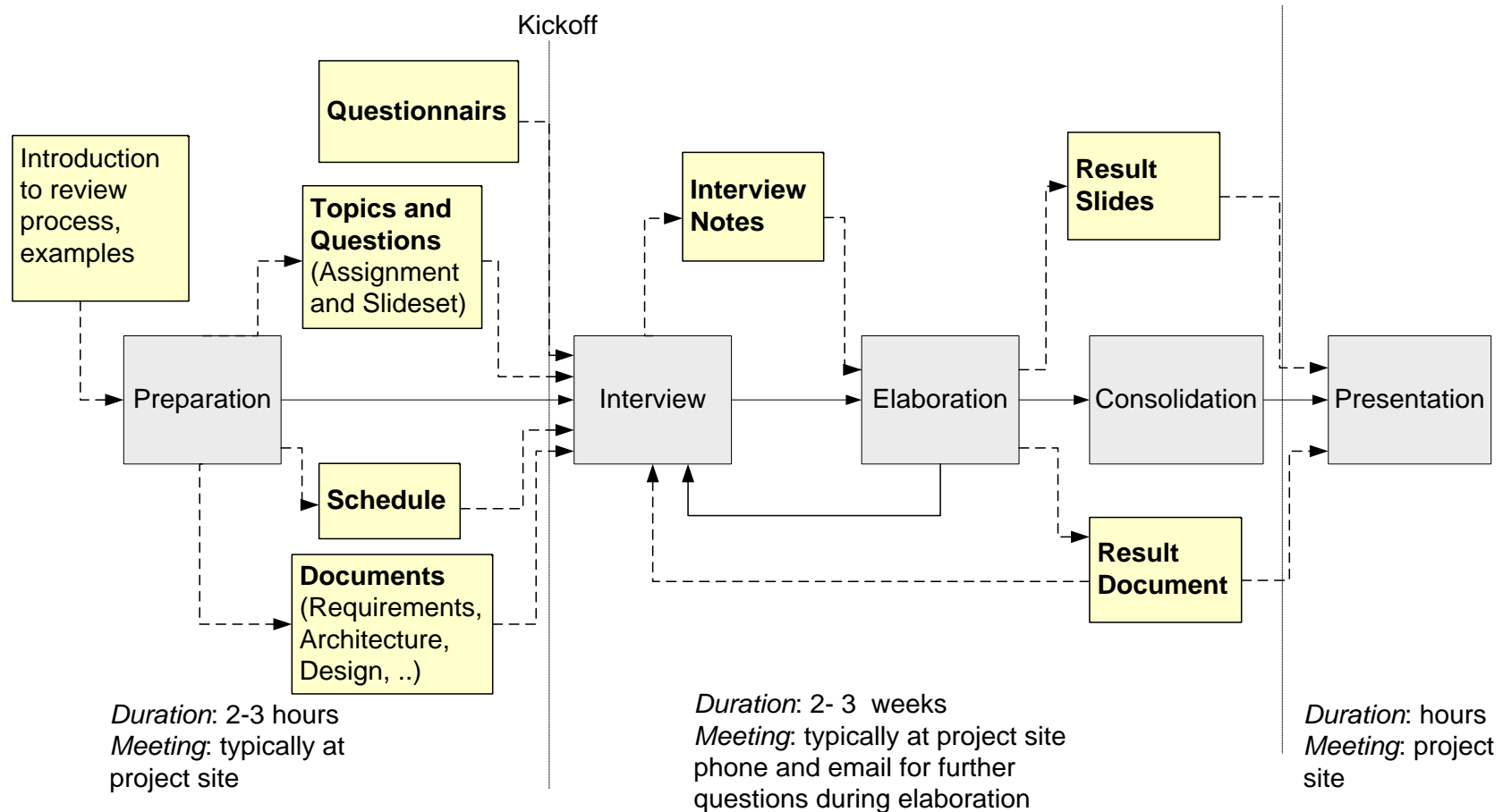
**Results:** Detailed report including architecture description, SWOT analysis, measures

**Benefits:** Rating of a (test) system architecture regarding compliance to its requirements, **dedicated** measures; minimal effort for project team

## SWOT ANALYSIS



# Review process for industry practice reviews



# Architecture Quality and Architecture Reviews

## Agenda

Architecture Quality

Quantitative Reviews

Qualitative Reviews

Experience-based Reviews

**Scenario-based Reviews**

Summary

# Architecture Tradeoff Analysis Method (ATAM)\*

## Purpose:

Identify risks, sensitivity points and tradeoffs.

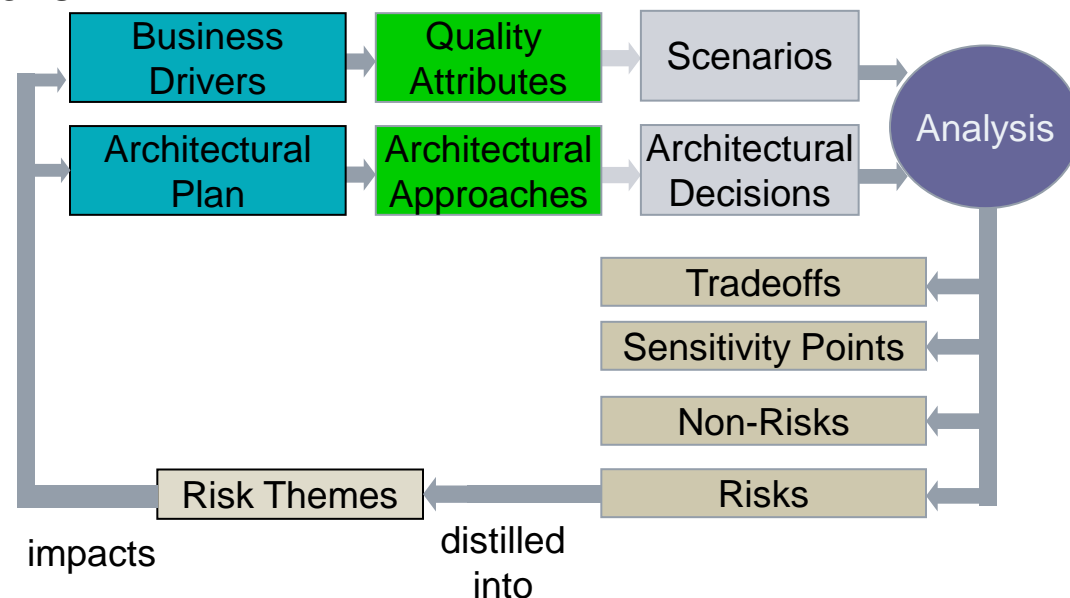
## Workshop steps:

- Present the ATAM
- Present business drivers
- Present architecture
- Identify architectural approaches
- Generate quality attribute utility tree
- Analyze architectural approaches
- Brainstorm and prioritize scenarios
- Analyze architectural decisions
- Present results

**Effort:** 3–4 day workshops, evaluation team 30–40 person days, project team 30–40 person-days (In practice, a lot less, because of previous experiences and result reuse)

**Results:** Prioritized list of scenarios regarding business drivers, risks and tradeoff points related to architectural decisions

**Benefits:** Identified risk, documented basis for architectural decisions



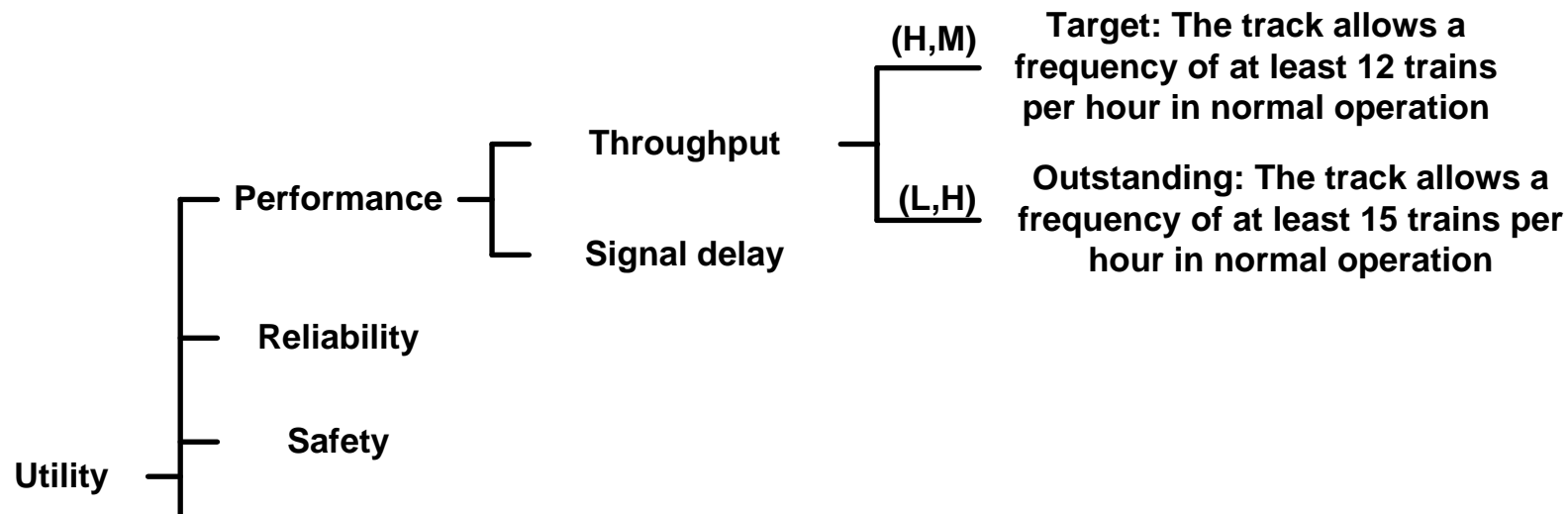
\*Source: Software Engineering Institute, Carnegie Mellon University

# Structuring quality attributes using a "Utility Tree"

Quality attributes are not atomic. They usually have multiple facets.

## A Utility Tree

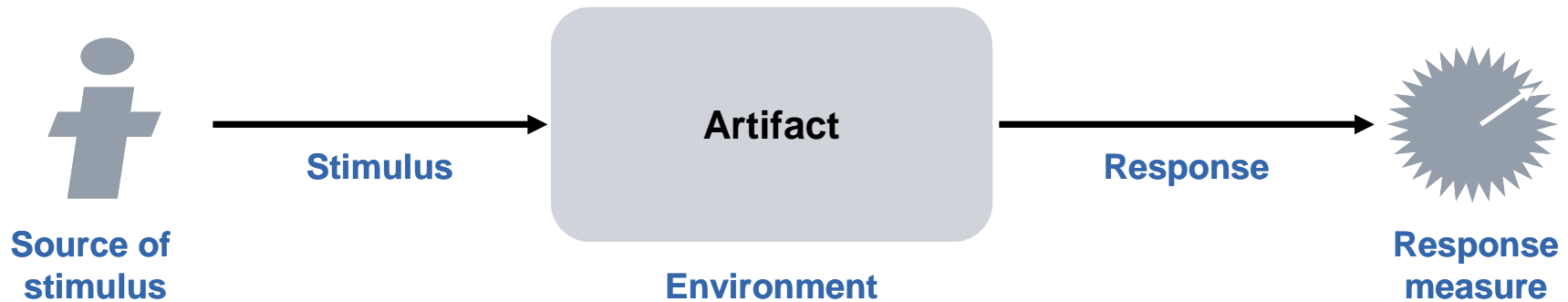
- describes the most important quality attributes (in a project)
- details each quality attribute in relevant facets
- relates scenarios, described in requirements, to quality attributes or their facets and thus allows to prioritize them



# Scenario description Examples

Remember WS2

**SIEMENS**  
*Ingenuity for life*



Stimulus for the event that concerns the quality attribute, e.g. function invoked, failure, modification	Relevant assumptions about the environment and the relevant conditions	Precise statement of quality attribute response, e.g. response time, difficulty of modification
One of the CPUs fails	Normal operation in a redundant system	hand-over to mate in <0.1s
X-Ray tube fails	Official lifetime end not yet reached	remote alarm to service center; tube replaced in < 24h
Database is changed from MySQL to Oracle	SW modification during development	Change implemented in 20 work days

## System Architecture Analysis for the Field

**Purpose:** Evaluates (test) system lifecycle scenarios.

Checklist/Templates to focus on cross-cutting aspects:

- **Robustness** against all (including invalid) external signal inputs
- **System upgrades** under all (including fatal) external circumstances
- **Service and Failure resolution** during abnormal conditions
- **Diagnostic capabilities** especially under commercial conditions of the (test) system
- **(Test) System identification**, including hardware and software components
- Other relevant aspects for the given domain
  - Examples: fire safety, tamper resistance, security against sabotage,

**Initial Effort:** 2–3 day workshop, evaluation team 10–20 person days, project team 15–25 person days (for creating the checklists/templates).

**Results:** Checklists, Prioritized scenarios, mapping of scenarios to the architecture with associated cost

**Benefits:** Clarification of quality goals, improved documentation, improved communication

Source: Jürgen Salecker, C. Morra

RAMSS=Reliability, Availability, Maintainability, Safety, Security



## Comparison of qualitative reviews

	Active Design Review	Industry Practice	Architecture Tradeoff Analysis Method	System Architecture Analysis for the Field
<b>Interaction</b>	Designer, reviewer	Interviews	Workshop	Workshop
<b>Phase</b>	Detailed component / module design ready	After architecture has been designed	Architecture design complete enough for walkthroughs	Detailed component / module design ready
<b>Strength</b>	Focused on finding defects in design	Concrete measures	Bring stakeholders together, requirement prioritization	Concrete measures
<b>Key restriction</b>	Small scale	No common understanding of requirement priorities	No measures	Focus on system in operational conditions
<b>Duration</b>	2 days / reviewer	Four weeks regular 1 day flash	Two weeks	Two weeks

# Architecture Quality and Architecture Reviews

## Agenda

Architecture Quality

Quantitative Reviews

Qualitative Reviews

Experience-based Reviews

Scenario-based Reviews

**Summary**


## What we learned ...

- Review techniques are essential in an architect's toolbox
- Architecture reviews help improve quality
- Different types of quantitative and qualitative architecture reviews
- How to make use of architecture reviews in various situations and from different perspectives

## Further readings

Use the SSA Wiki :  
<https://wiki.ct.siemens.de/x/fReTBQ>

and check the “Reading recommendations”:  
<https://wiki.ct.siemens.de/x/-pRgBg>

- 
- **Architect's Resources:**
    - Competence related content
    - Technology related content
    - Design Essays
    - Collection of How-To articles
    - Tools and Templates
    - Reading recommendations
    - Job Profiles for architects
    - External Trainings
    - ... more resources

## Experience-based review: Scoping

Every architecture review needs a focus! Otherwise it is impossible to provide a valuable result back to the project team.

The initial step of an architecture review is therefore dedicated to identifying:

- **The review topic:** What is the overall goal and what are the 3 to 5 key areas that contribute to this goal?
- **Requirements to evaluate against:** What are the concrete measures regarding the goal and the key areas that the architecture under review should fulfill?
- **Sources from which the required information could be retrieved:**

Documents, models, design files (e.g. schematics, technical drawings, layouts), source code, a demo, prototypes, test reports, and interviews

**SIEMENS**

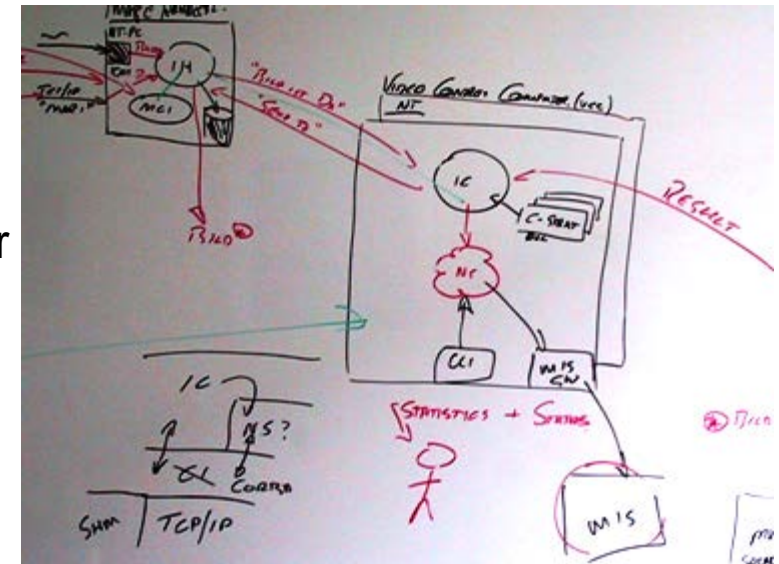
**Review Objectives**

1. Evaluate document quality, identify strength, shortcomings and potential for improvement (COHERENCE PRS, COHERENCE SDS, RTT Workspace FS).
2. We have significant differences between the LANTIS and DICOM data models (Fraction counting, manual recording, relative setups, printing, ...).  
Evaluate the risk of the current concept. What went wrong in concept and realization phase?
3. We will finally have 5 databases in the system, a weak data transfer concept, and various different deployment scenarios.  
Evaluate the risk of central data management.

## Experience-based review: Information collection

Retrieving the relevant information about the architecture requires "accessing" multiple sources!

- **Documents** describe the "desired" architecture, but not necessarily the implemented architecture
- **Design files, code, demos, and test reports** help to uncover the real architecture, its strengths and weaknesses, but do not tell whether particular deficiencies are already addressed and by what measures
- **Interviews** with all stakeholders of the architecture will tell you how the architecture under review is received, assessed, and what the next development steps are



Collecting information is neutral: No assessments of the retrieved information must be made

## Experience-based review: Conducting interviews

### Preparation

Always have two reviewers (one to talk, the other one to write)

Keep the groups to be interviewed at a time small (usually 1 to 3 persons)

### Build trust

Explain the process to all interviewees (kick-off meeting)

Restate values at the beginning of each interview: Anonymous, no judgment during the interview

### Structure

Have a set of starting questions: What is your role? How long have you been part of this organization?

Have a set of finishing questions: What do you like, and what would you like to have changed? Anything you want to mention?

Do not accept slides at the interview (with exceptions)

Have people draw the architecture on a white board (use a photo camera)

## Experience-based review: Information evaluation

Assessing the information gathered during the collection step and drawing conclusions from it is the review's core activity.

The result of the evaluation step is a review report with the following structure:

- **Goals:** A description of the review goal and the 3 to 5 key areas that were addressed, including the requirements for these key areas
- **Procedure:** How the information was retrieved and assessed
- **Description and assessment:** A description of the system architecture from the perspective of each relevant key area, and the assessment of its quality with respect to the requirements for these areas
- **Recommendations:** Measures for improvement, if certain parts of the reviewed architecture show deficiencies





<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Objectives .....	4
1.2	Intended Audience .....	5
1.3	Approach .....	5
1.4	Overview .....	6
<b>2</b>	<b>Management Summary .....</b>	<b>7</b>
<b>3</b>	<b>System Overview .....</b>	<b>8</b>
3.1	Business Model .....	8
3.2	Current Situation .....	8
3.3	Requirements .....	9
3.3.1	Functional Requirements .....	9
3.3.2	Non-Functional Requirements .....	10
3.4	System Architecture .....	10
3.4.1	Hardware Architecture .....	12
3.4.2	Software Architecture .....	13
<b>4</b>	<b>Review Results .....</b>	<b>15</b>
4.1	Results for Key Questions .....	15
4.1.1	Application-Hosting in EAGLE .....	15
4.1.1.1	Architectural Considerations for Splitting the Operating Responsibility .....	16
4.1.2	Extensibility of the Finavigate Software Architecture .....	17
4.1.2.1	Quality of current architecture .....	17
4.1.2.1.1	Strengths of the architecture .....	18
4.1.2.1.2	Weaknesses of the architecture .....	18
4.1.2.2	Development Process .....	19
4.1.2.2.1	Strengths of the Process .....	20
4.1.2.2.2	Weaknesses of the Process .....	20
4.1.2.3	Technology .....	21
4.1.2.3.1	Strengths of the Technology Decision .....	22
4.1.2.3.2	Weaknesses of the Technology Decision .....	22
4.1.2.4	Integration of New User Languages .....	22
4.1.3	Scalability of the Finavigate Software Architecture .....	23
4.1.3.1	Scalability Evaluation .....	23
4.1.3.1.1	Strengths Concerning Scalability .....	24
4.1.3.1.2	Weaknesses Concerning Scalability .....	24
4.1.3.2	Removal of Data from Online Storage .....	26
4.2	Side Findings .....	27
4.2.1	Organizational weaknesses .....	27
4.3	Opportunities .....	27
4.4	Risks .....	27
<b>5</b>	<b>Proposed Measures .....</b>	<b>27</b>

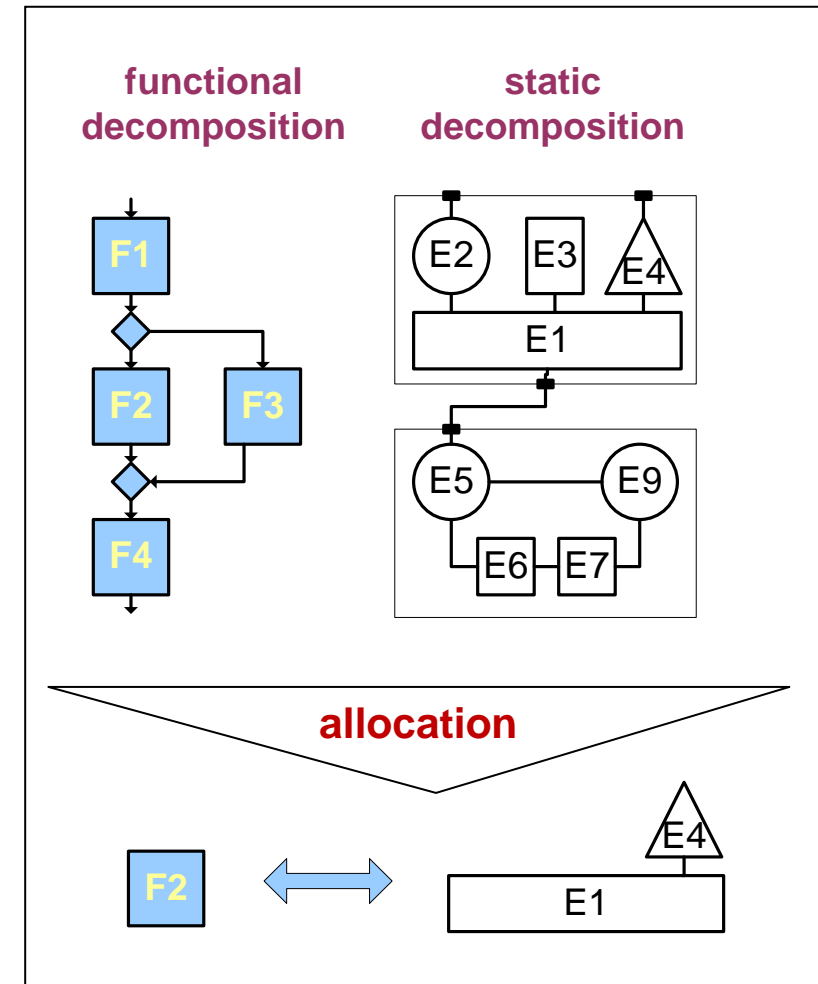
**Example**

## Architecture description

Describe the part of the architecture relevant for the review as you understood it, ideally in diagrams.

Relevant system parts or mechanisms are described separately.

If suitable architecture diagrams exist, it is OK to copy and cite them.



## Feedback structure

### **Context:**

Where is the problem?

What is the necessary background for understanding the problem?

### **Problem:**

Point at the painful subject.

### **Solution:**

Make solution proposals;

reference other sections as appropriate for sound argumentation.

## Feedback structure example – Technical advice

### Application redundancy and context sensitivity

**Context:** The applications have to consider the redundancy concept and the context sensitivity concept. This means application developers have to think about when to synchronize application data between active and standby node or between different contexts.

**Problem:** This is a high burden on the developers and a high risk for errors. Every application will have to be changed when the applications later on have to adopt a common strategy.

**Solution:** Think of a synchronization strategy, provide guidelines and support developers in following the guidelines.

## Feedback structure example – Process advice

### Development cycles

**Context:** The development cycle for a new release of the product software takes at least one year. Many features are added.

**Problem:** The long planning, scheduling and testing phases mean that a running system is only achieved late in the cycle. This moves the risk of integration and quality problems to the end of the project.

**Solution:** Shorten planning, scheduling and testing, e.g. by making more use of automated tests, to achieve shorter and incremental development cycles. Shorter development cycles allow for smaller features sets, and vice versa.

## Feedback structure example – Strategic advice

### Business goals

**Context:** Individual developers and marketing have a different understanding of the priorities the project should observe.

**Problem:** The commitment to the project strategy, e.g. division L versus division S, or Features versus Architecture, is smaller than it could be.

**Solution:** Make a clear decision and communicate it inside the project about the business goals. Keep those persistent and consistent.

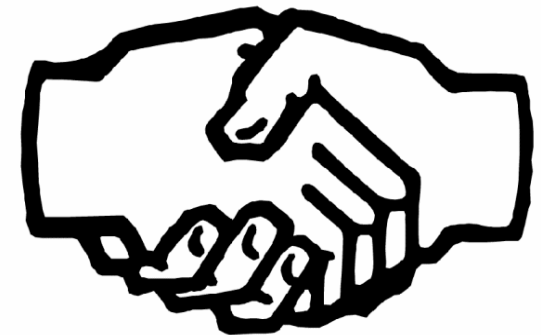
## Experience-based review: Consolidation

Have the customer review the document (findings and measures).

Have the architect review the outline of the architecture.

**Negotiate** content if necessary

- Pay attention to political issues
- But don't change the review results simply because a stakeholder doesn't like them



**Prepare** to defend your results

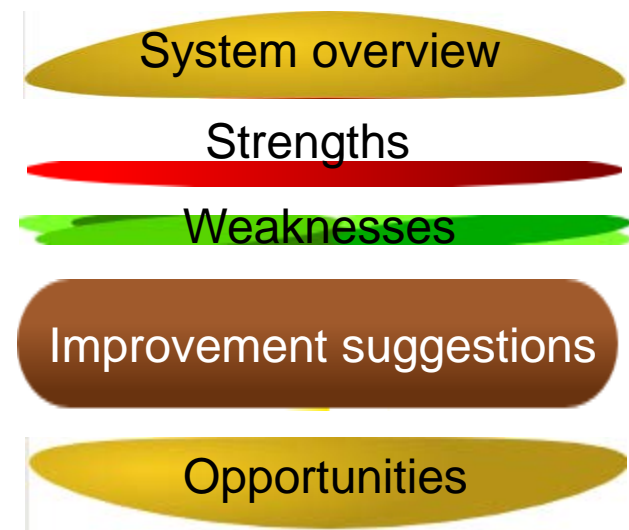
- Make sure you have understood the architecture and its boundary conditions
- Get a first impression of the feasibility of your recommendations

## Experience-based review: Feedback

Workshops communicate the review results to the customer!  
Focus on key issues; do not run through the entire review report.

Begin with the review goals and key areas examined to set the right scope

- Not only mention the major weaknesses of the reviewed architecture, but also its key strengths
- Spend the most time on suggestions for improvements; this is the information that is most important for the customer
- Wrap up with the opportunities when measures are taken





# What are the typical issues found in a qualitative review?

**Problem definition:** The requirements are not clearly or sufficiently defined

**Architecture:** The proposed solution does not adequately solve the problem

**Lifecycle:** The proposed solution is not optimized for lifecycle costs

**Technology:** The chosen technologies are not adequate

**Domain knowledge:** The necessary domain knowledge is not present in the organization

**Process:** The processes are not adequate for developing solutions in the environment

**Organization:** Management monitoring, staffing, controlling and decision-making mechanisms are not adequate

## Key tactics: Multiple views and trust

For successfully rating and improving an architecture, the external expert uses techniques based on a set of basic principles.

