THE CRAFT OF SOFTWARE ARCHITECTURE

# Architecture Reviews

Christa Schwanninger, Frank Buschmann,
Siemens AG, Corporate Technology

Christa.Schwanninger@siemens.com
Frank.Buschmann@siemens.com

THE CRAFT OF SOFTWARE ARCHITECTURE

## Architecture reviews

**Learning objectives**

- Become familiar with architecture reviews from the perspective of the architect of a reviewed architecture, a reviewer, and a review initiator
- Get to know when and why to conduct an architecture review
- Become familiar with techniques for architecture reviews

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

## Architecture review

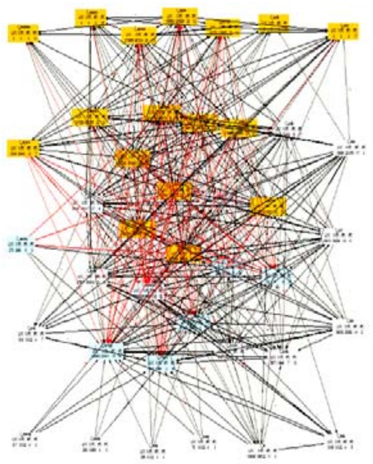
Agenda

- **Architecture reviews as feedback measure**
- **Core structure of architecture reviews**
- **Techniques for architecture reviews**
- **Summary**

© Christa Schwanninger, Frank Buschmann, all rights reserved

## A motivating thought

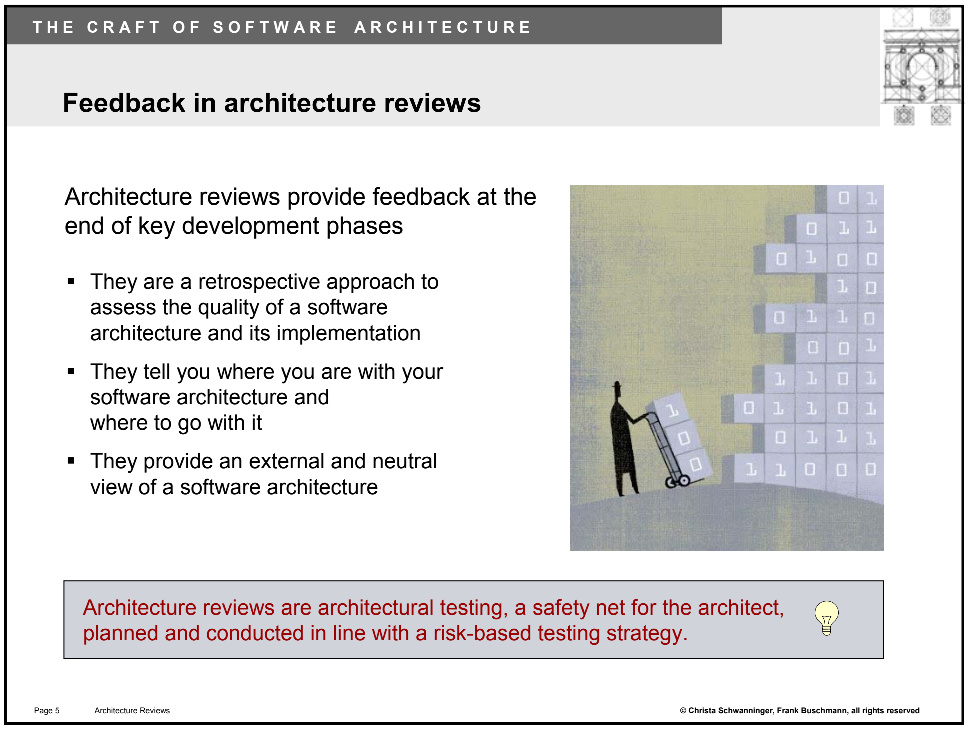
**… software architecture has a profound influence on project organizations' functioning and structure. Poor architecture can reflect poorly defined organizations with inherent project inefficiencies, poor communication, and poor decision making.***



* *Architecture Reviews: Practice and Experience*, J.Maranzano, S. Rozsypal, G. Zimmermann, G. Warnken, P. Wirth, D. Weiss, IEEE Software, 2005

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved


THE CRAFT OF SOFTWARE ARCHITECTURE

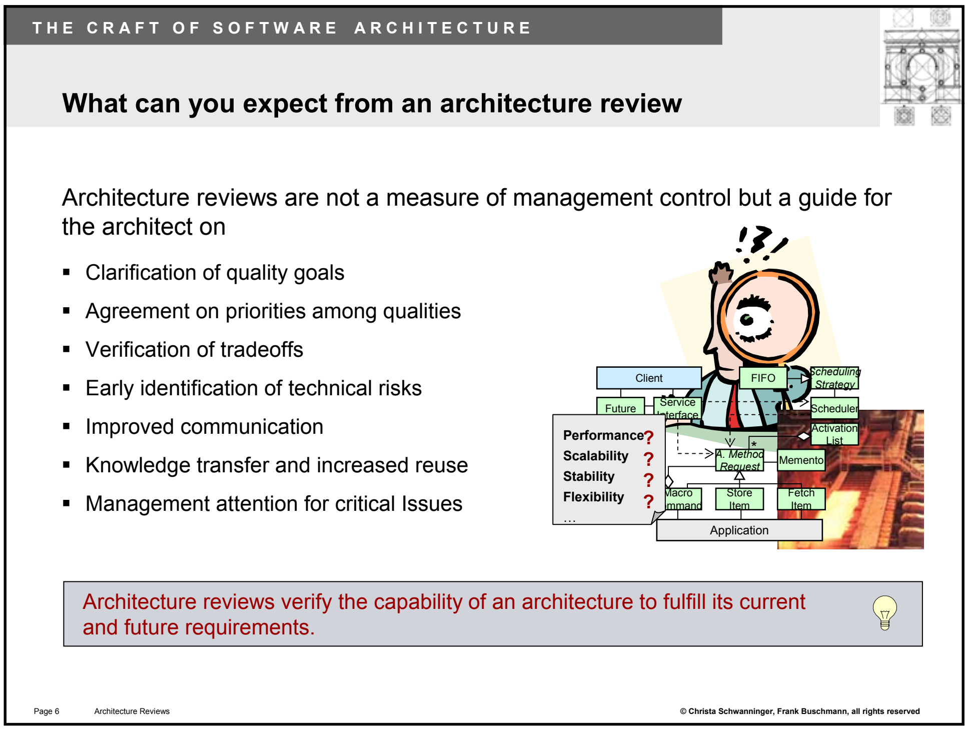
## Feedback in architecture reviews

Architecture reviews provide feedback at the end of key development phases

- They are a retrospective approach to assess the quality of a software architecture and its implementation
- They tell you where you are with your software architecture and where to go with it
- They provide an external and neutral view of a software architecture

Architecture reviews are architectural testing, a safety net for the architect, planned and conducted in line with a risk-based testing strategy.

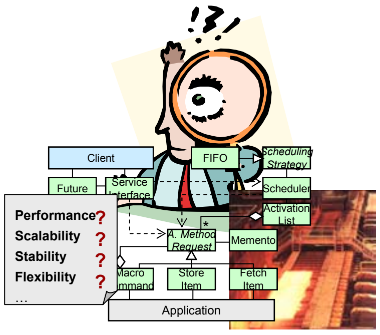
Page 5 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved


THE CRAFT OF SOFTWARE ARCHITECTURE

## What can you expect from an architecture review

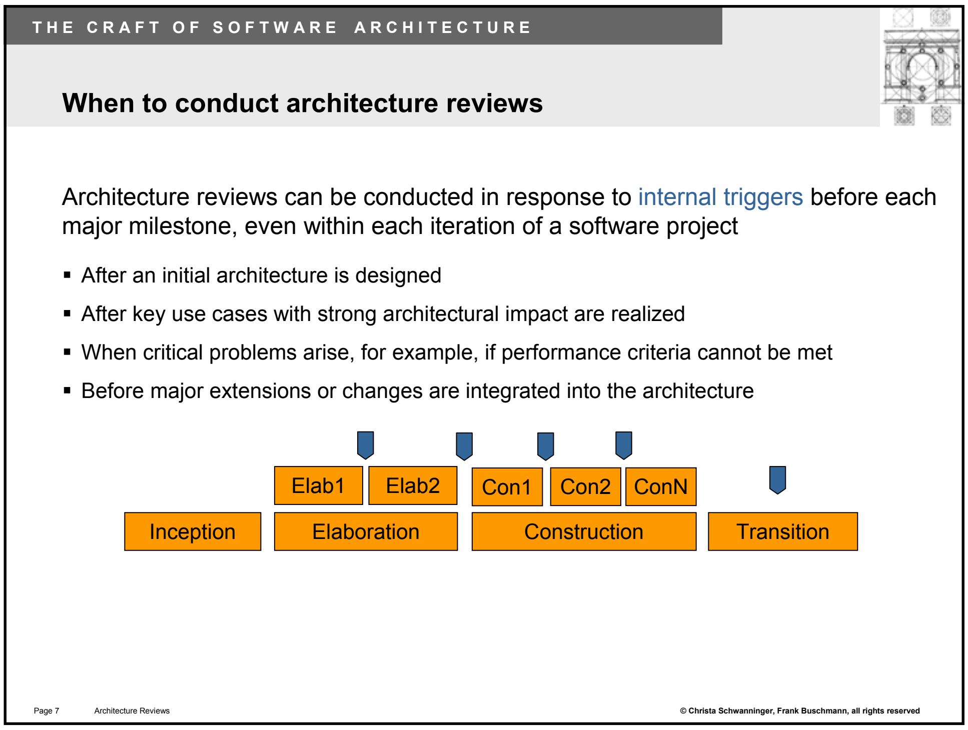
Architecture reviews are not a measure of management control but a guide for the architect on

- Clarification of quality goals
- Agreement on priorities among qualities
- Verification of tradeoffs
- Early identification of technical risks
- Improved communication
- Knowledge transfer and increased reuse
- Management attention for critical Issues



Architecture reviews verify the capability of an architecture to fulfill its current and future requirements.

Page 6 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved

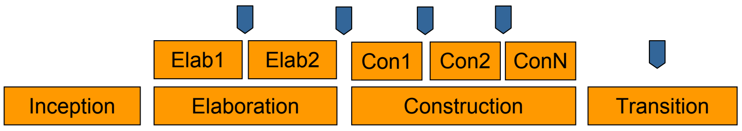


THE CRAFT OF SOFTWARE ARCHITECTURE

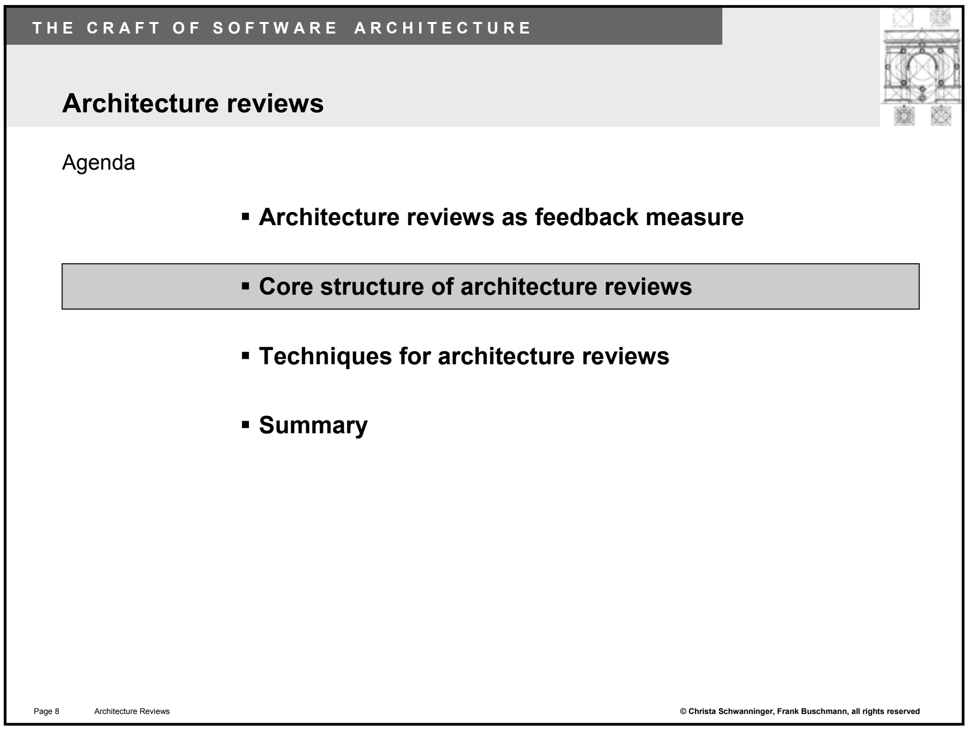
## When to conduct architecture reviews

Architecture reviews can be conducted in response to internal triggers before each major milestone, even within each iteration of a software project

- After an initial architecture is designed
- After key use cases with strong architectural impact are realized
- When critical problems arise, for example, if performance criteria cannot be met
- Before major extensions or changes are integrated into the architecture



Page 7 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved



THE CRAFT OF SOFTWARE ARCHITECTURE

## Architecture reviews

Agenda

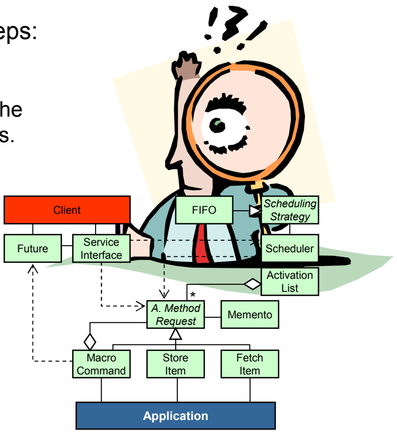
- **Architecture reviews as feedback measure**
- **Core structure of architecture reviews**
- **Techniques for architecture reviews**
- **Summary**

Page 8 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved

## Review structure: overview

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.



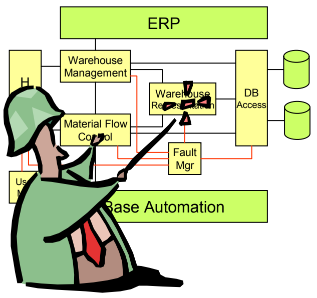
© Christa Schwanninger, Frank Buschmann, all rights reserved

## Review structure: 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, source code, a demo, test reports, and interviews



© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

## Review structure: information collection

Retrieving the relevant information about the architecture requires to „access“ multiple sources!

- **Documents** describe the „desired“ architecture, but not necessarily the implemented architecture.
- **Code, demos, and test reports** help to uncover the real architecture, its strengths and weaknesses, but do not tell whether particular deficiencies already get tackled 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

© Christa Schwanninger, Frank Buschmann, all rights reserved

## Review structure: 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 was the information retrieved and assessed
- **Description and Assessment**: A description of the software 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

Be politically correct but honest – decisions on how to proceed with the architecture or even the entire project will be made on the review results

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

## Procedure: 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 examined key areas to set the right scope
- Not only mention the major weaknesses of the reviewed architecture, but also its key strengths
- Spend most time on the suggestions for improvements, this is the information that is most important for the customer
- Inform the architects / project team of the reviewed system before the results get presented to the customer – this avoids unpleasant surprises

© Christa Schwanninger, Frank Buschmann, all rights reserved

## Architecture reviews

Agenda

- **Architecture reviews as feedback measure**
- **Core structure of architecture reviews**
- **Techniques for architecture reviews**
- **Summary**

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

## Skills needed to conduct a review

| Technical knowledge | Soft skills | Methodological knowledge |
|---|---|---|
| ▪ Design qualities<br>▪ Design tactics<br>▪ Technology<br>▪ Processes<br>▪ Methodology (test, CM ..) | ▪ Conflict management<br>▪ Listen<br>▪ Accept feedback<br>▪ Initiative<br>▪ Change orientation<br>▪ Learning<br>▪ Strategic judgment and risk management | ▪ Review techniques<br>▪ Feedback techniques<br>▪ Moderation<br>▪ Presentation<br>▪ Architectural views |

© Christa Schwanninger, Frank Buschmann, all rights reserved

## Techniques for architecture review

**Quantitative**

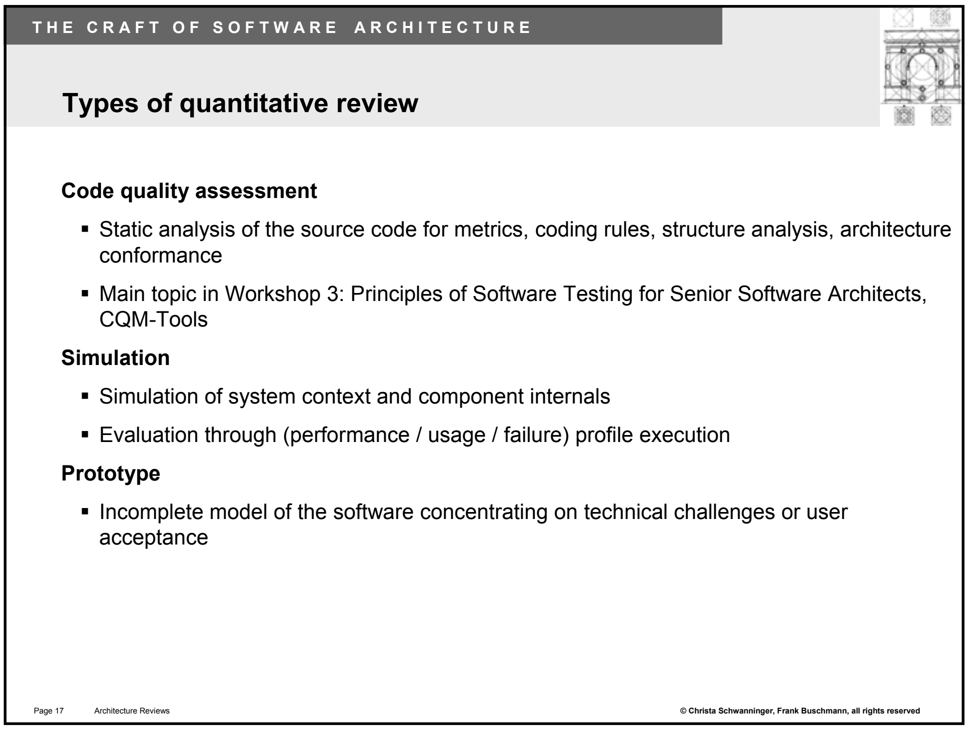
- Code quality assessment
- Simulations
- Prototypes

**Qualitative**

- Scenario-based approaches
- Experience-based approaches

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved



THE CRAFT OF SOFTWARE ARCHITECTURE

## Types of quantitative review

**Code quality assessment**

- Static analysis of the source code for metrics, coding rules, structure analysis, architecture conformance
- Main topic in Workshop 3: Principles of Software Testing for Senior Software Architects, CQM-Tools

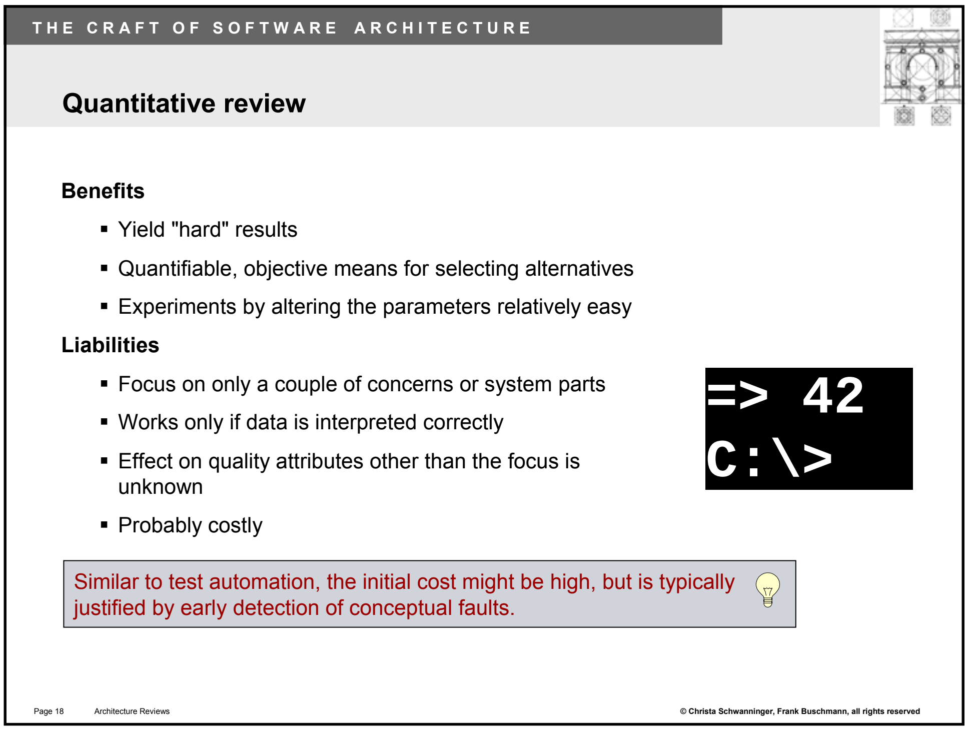
**Simulation**

- Simulation of system context and component internals
- Evaluation through (performance / usage / failure) profile execution

**Prototype**

- Incomplete model of the software concentrating on technical challenges or user acceptance

Page 17 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved



THE CRAFT OF SOFTWARE ARCHITECTURE

## Quantitative review

**Benefits**

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

**Liabilities**

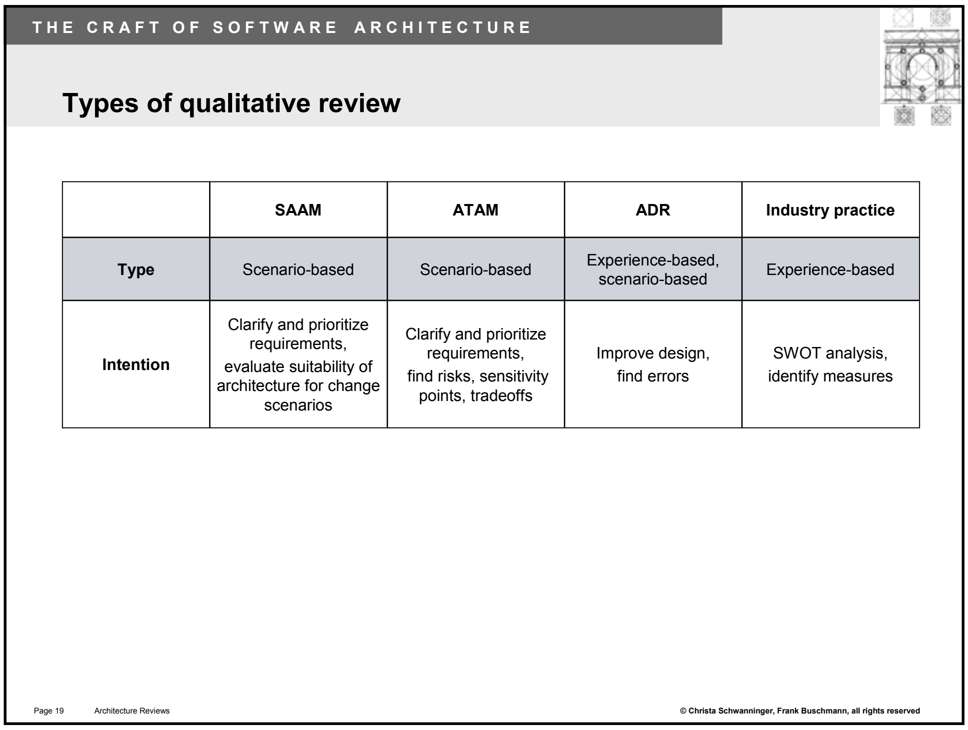
- Focus on only a couple of concerns or system parts
- Works only if data is interpreted correctly
- Effect on quality attributes other than the focus is unknown
- Probably costly

```
=> 42
C:\>
```

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

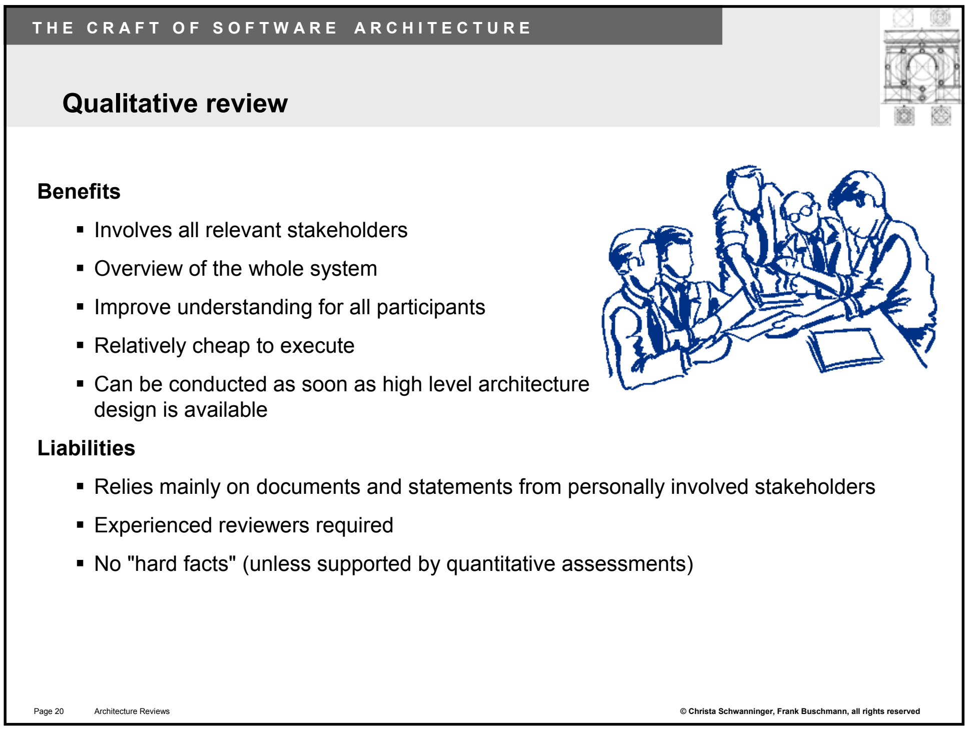
Page 18 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved



## Types of qualitative review

| | 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 |



## Qualitative review

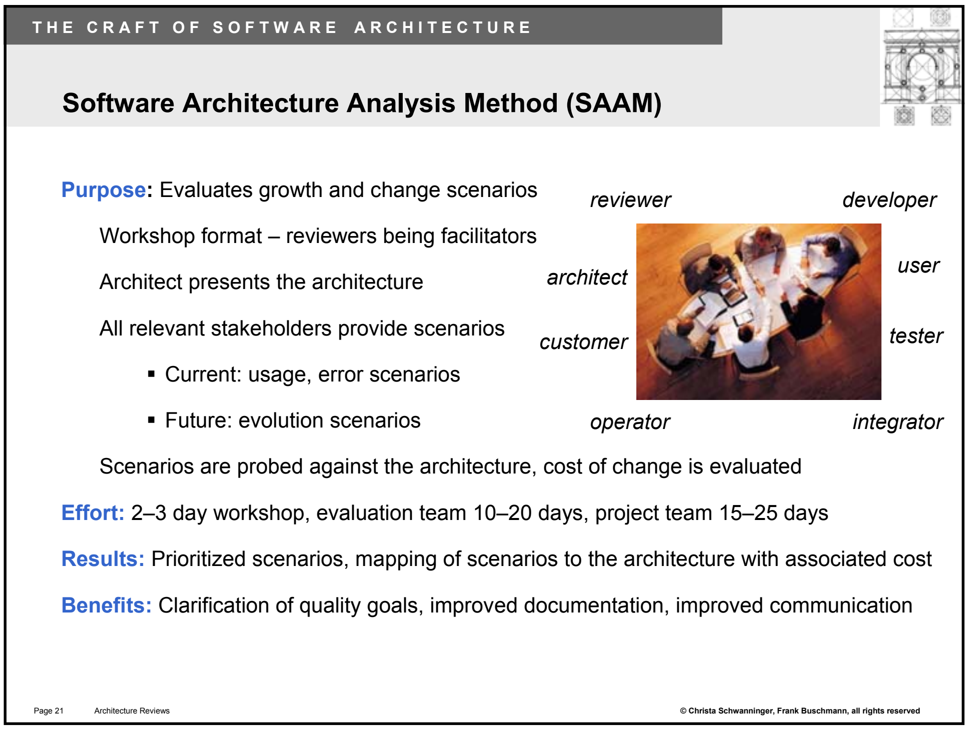
**Benefits**

- Involves all relevant stakeholders
- Overview of the whole 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 assessments)

© Frank Buschmann, all rights reserved



## Software Architecture Analysis Method (SAAM)

**Purpose:** Evaluates growth and change scenarios

Workshop format – reviewers being facilitators

Architect presents the architecture

All relevant stakeholders provide scenarios

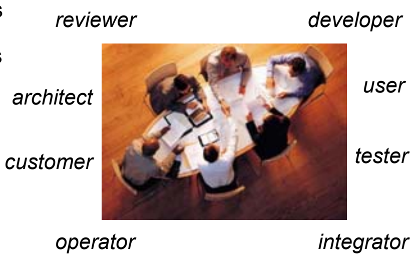
- Current: usage, error scenarios
- Future: evolution scenarios

Scenarios are probed against the architecture, cost of change is evaluated

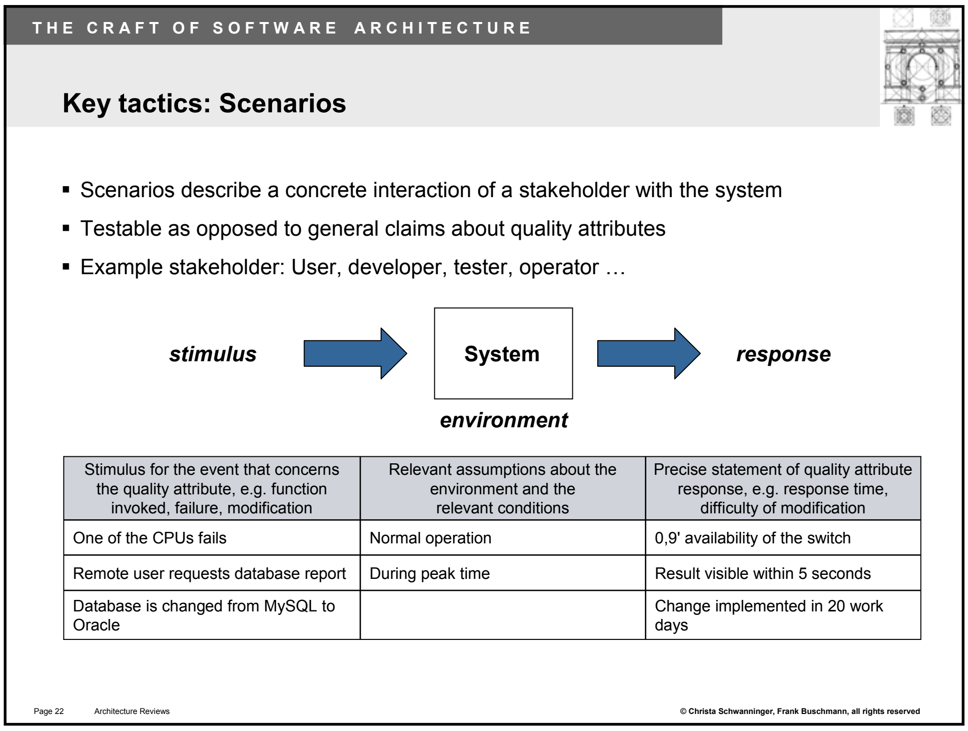
**Effort:** 2–3 day workshop, evaluation team 10–20 days, project team 15–25 days

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

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

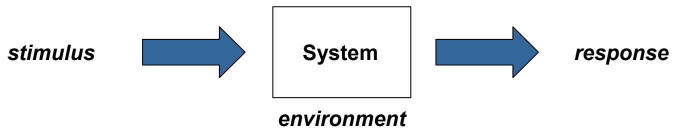


© Christa Schwanninger, Frank Buschmann, all rights reserved



## Key tactics: Scenarios

- Scenarios describe a concrete interaction of a stakeholder with the system
- Testable as opposed to general claims about quality attributes
- Example stakeholder: User, developer, tester, operator …



| 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 | 0,9' availability of the switch |
| Remote user requests database report | During peak time | Result visible within 5 seconds |
| Database is changed from MySQL to Oracle | | Change implemented in 20 work days |

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

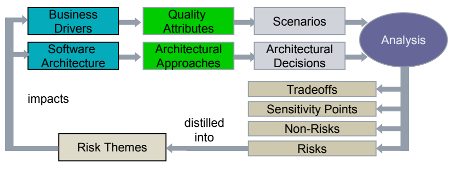
THE CRAFT OF SOFTWARE ARCHITECTURE

## Architecture Tradeoff Analysis Method (ATAM )

**Purpose:** Identify risks, sensitivity points and tradeoffs

Enhancement of SAAM, additional measures for

- Aligning the qualities with the business drivers
- Relating architectural decisions with quality goals, identifying risks and tradeoffs



Iteration with different stakeholder groups

**Effort:** 3–4 day workshops, evaluation team 30–40, project team 30–40 person days

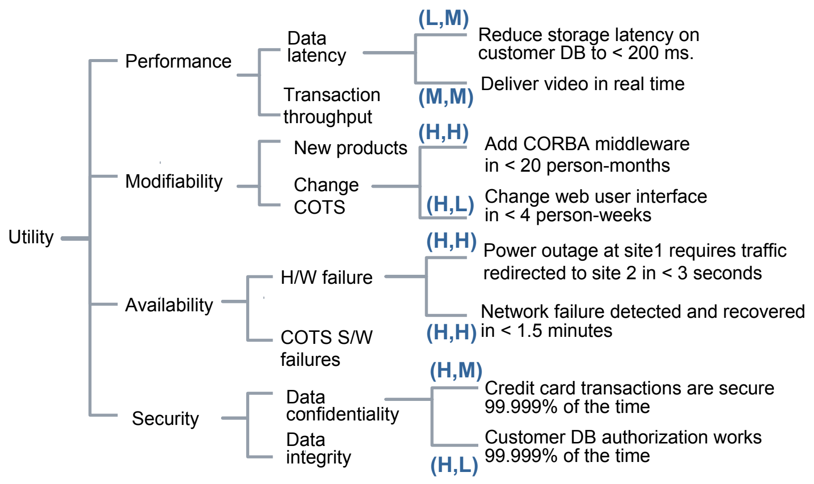
**Results:** Prioritized list of scenarios with relation to business drivers, risks and tradeoff points related to architectural decisions

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

© Christa Schwanninger, Frank Buschmann, all rights reserved

THE CRAFT OF SOFTWARE ARCHITECTURE

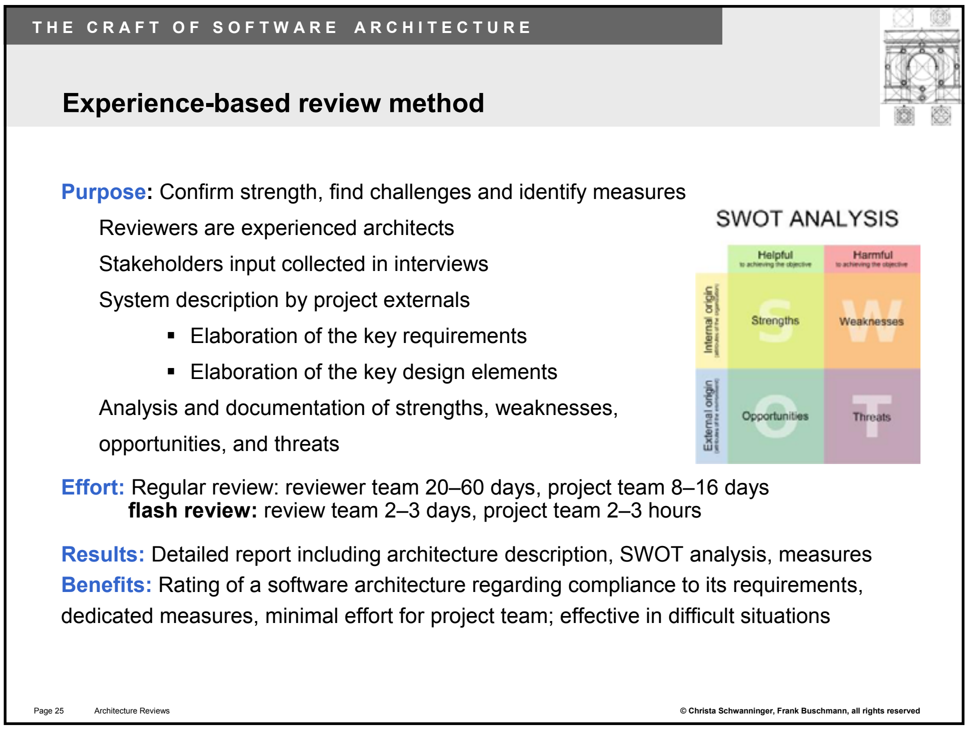
## Key tactics: Utility tree



Utility
Performance
Data latency
Transaction throughput
(L,M)
Reduce storage latency on customer DB to < 200 ms.
(M,M)
Deliver video in real time
Modifiability
New products
Change COTS
(H,H)
Add CORBA middleware in < 20 person-months
(H,L)
Change web user interface in < 4 person-weeks
Availability
H/W failure
COTS S/W failures
(H,H)
Power outage at site1 requires traffic redirected to site 2 in < 3 seconds
(H,H)
Network failure detected and recovered in < 1.5 minutes
Security
Data confidentiality
Data integrity
(H,M)
Credit card transactions are secure 99.999% of the time
(H,L)
Customer DB authorization works 99.999% of the time

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved



## Experience-based review method

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

Reviewers are experienced architects

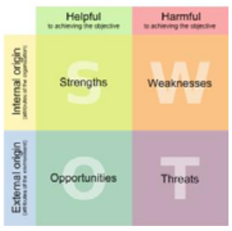
Stakeholders input collected in interviews

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

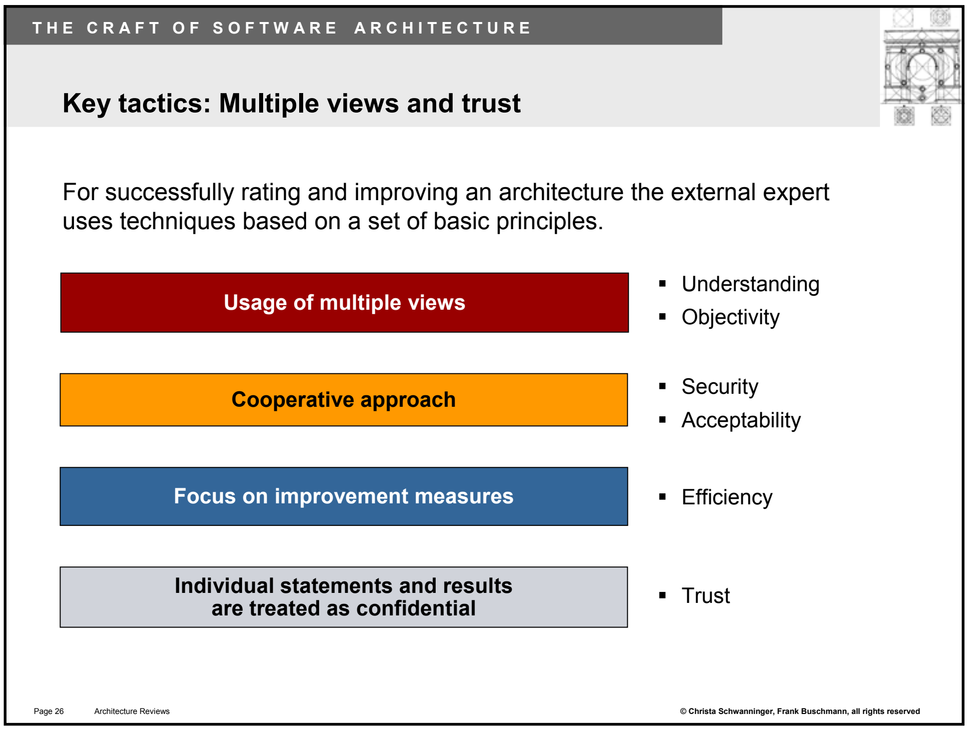
SWOT ANALYSIS



**Effort:** Regular review: reviewer team 20–60 days, project team 8–16 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 software architecture regarding compliance to its requirements, dedicated measures, minimal effort for project team; effective in difficult situations

© Christa Schwanninger, Frank Buschmann, all rights reserved



## 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.

| | |
|---|---|
| **Usage of multiple views** | ▪ Understanding<br>▪ Objectivity |
| **Cooperative approach** | ▪ Security<br>▪ Acceptability |
| **Focus on improvement measures** | ▪ Efficiency |
| **Individual statements and results are treated as confidential** | ▪ Trust |

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

## Active Design Review (ADR)

**Purpose:** Test design and design documentation

Review detailed designs for components / modules

Scenario-based, designer asks reviewer to solve 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

## Key tactics: Concrete and narrow focus

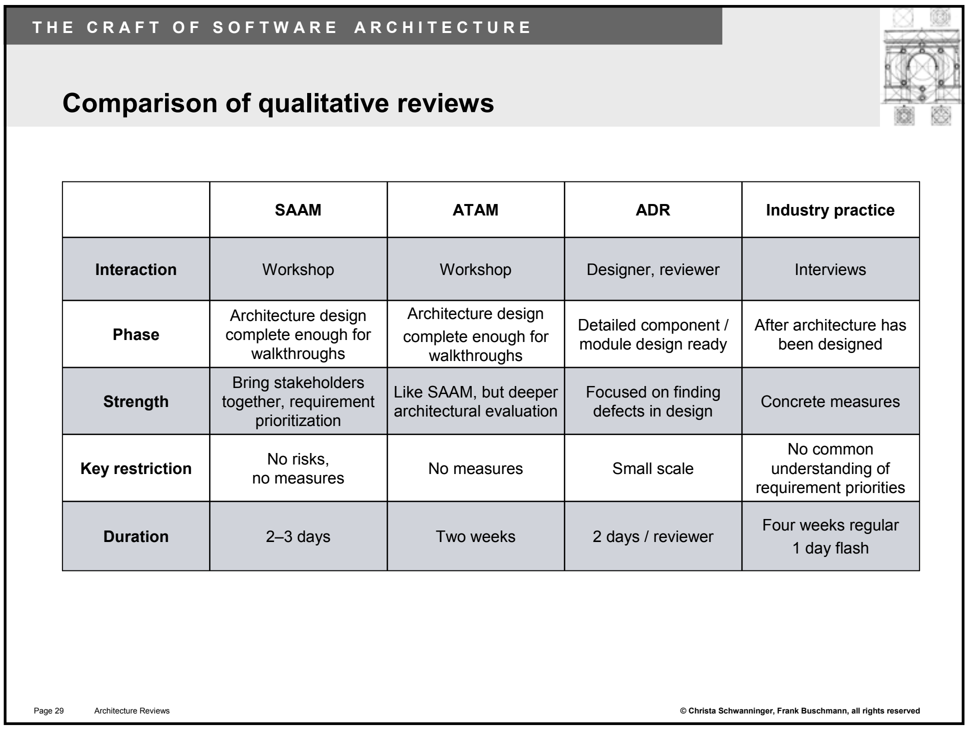
**The designer provides a questionnaire with *concrete* assignments for each reviewer**

- "*Write a short pseudocode program that uses the design to accomplish …*"
- *"Write down the exceptions that can occur"*
- *"For each access function provided, write down the specific requirements from the requirements list that you believe the function was designed to meet."*

**The effect is**

- Very focused review
- Test completeness and understandability of documentation, including requirements documentation
- There is a chance to find design errors
- The reviewer is not bored by the reviewing task

© Frank Buschmann, all rights reserved

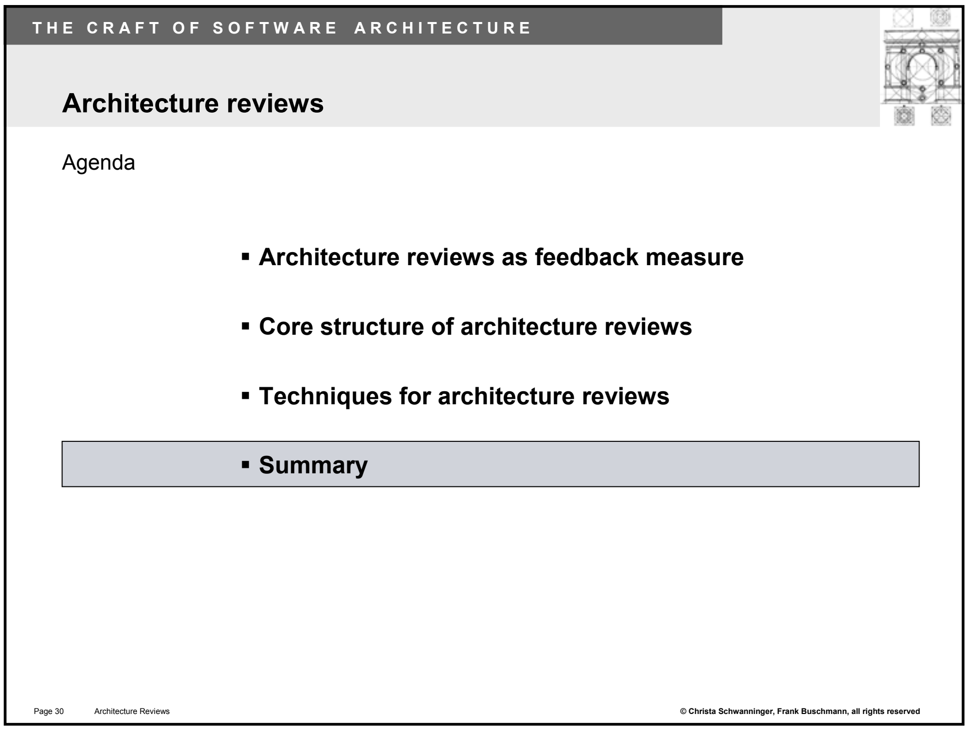


THE CRAFT OF SOFTWARE ARCHITECTURE

## Comparison of qualitative reviews

| | SAAM | ATAM | ADR | Industry practice |
|---|---|---|---|---|
| **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 |

Page 29 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved



THE CRAFT OF SOFTWARE ARCHITECTURE

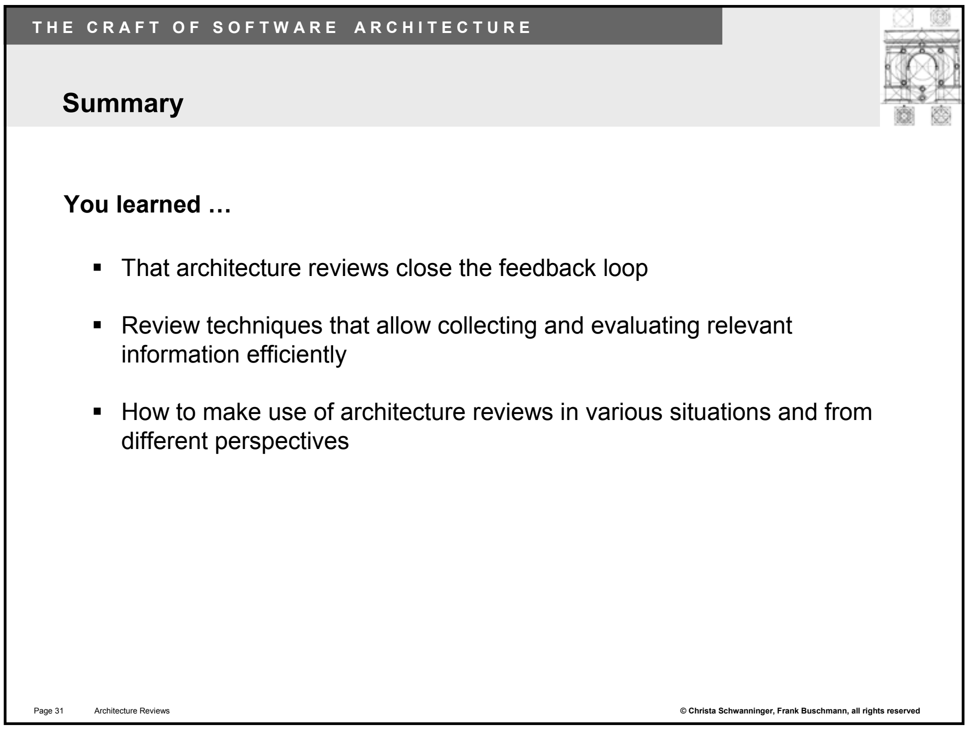
## Architecture reviews

Agenda

- **Architecture reviews as feedback measure**
- **Core structure of architecture reviews**
- **Techniques for architecture reviews**
- **Summary**

Page 30 Architecture Reviews © Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved

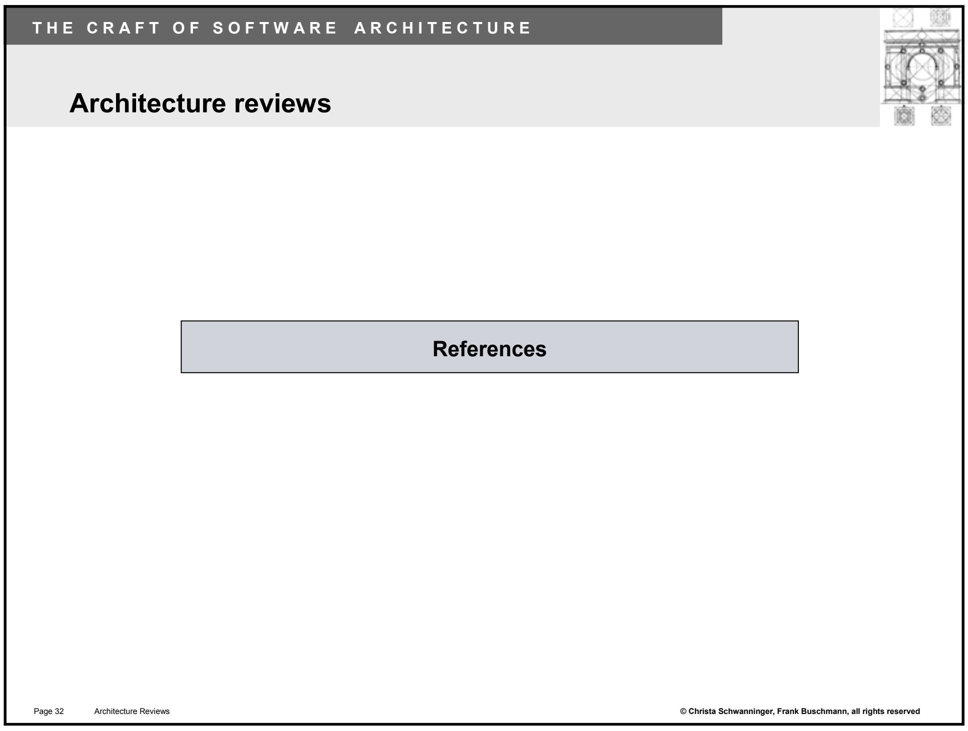


## Summary

**You learned …**

- That architecture reviews close the feedback loop
- Review techniques that allow collecting and evaluating relevant information efficiently
- How to make use of architecture reviews in various situations and from different perspectives

© Christa Schwanninger, Frank Buschmann, all rights reserved

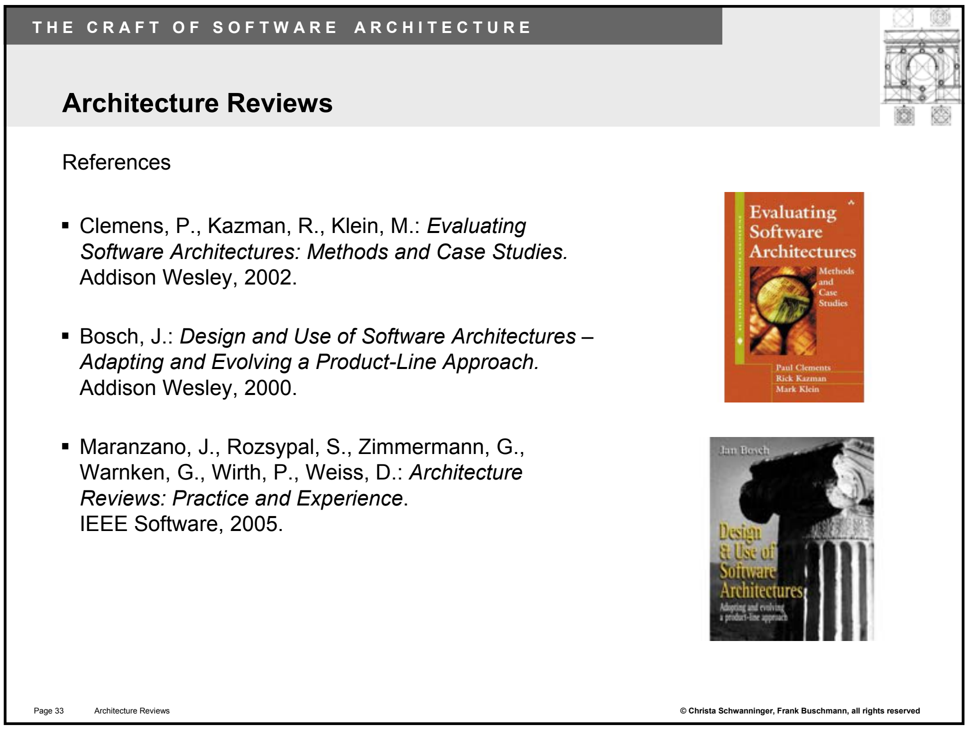


## Architecture reviews

**References**

© Christa Schwanninger, Frank Buschmann, all rights reserved

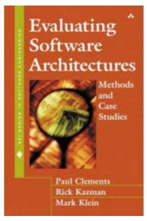
© Frank Buschmann, all rights reserved



## Architecture Reviews

References

- Clemens, P., Kazman, R., Klein, M.: *Evaluating Software Architectures: Methods and Case Studies.* Addison Wesley, 2002.
- Bosch, J.: *Design and Use of Software Architectures – Adapting and Evolving a Product-Line Approach*. Addison Wesley, 2000.
- Maranzano, J., Rozsypal, S., Zimmermann, G., Warnken, G., Wirth, P., Weiss, D.: *Architecture Reviews: Practice and Experience*. IEEE Software, 2005.



Jan Bosch
Design & Use of Software Architectures
Adopting and evolving a product-line approach

© Christa Schwanninger, Frank Buschmann, all rights reserved

© Frank Buschmann, all rights reserved