

architecture CHEAT SHEET

Conceptual development expertise in a condensed format



Architecture Reviews

Architecture reviews support transparency in software development, they identify major risks and expose important tradeoffs. This cheat sheet offers a well-founded overview.

FIND MORE CHEAT SHEETS ONLINE:

Free pdf-Download of the architecture cheat sheet collection:

www.architektur-spicker.de

4
No.

IN THIS ISSUE

- What's the benefit of architecture reviews?
- Which methods and tools are helpful?
- Who should be included in reviews and to which extent?



What's it about? (challenges)

- ➔ A new development effort is pending and initial solution approaches are being discussed. Are you and your team on the right path?
- ➔ Management has lost confidence in your solution. How do you win them back while radiating a sense of security?
- ➔ Different stakeholders have contradictory goals for your software. How do you specify and prioritise their wishes?
- ➔ Modernization, extension or larger remodelling efforts are planned. How do you find and communicate suitable solution approaches?
- ➔ You want to fuel iterative architecture efforts. How do you get continuous feedback?



Aspects of architecture reviews

Various aspects can be analysed during architectural reviews. In isolation or combined in a wider review effort.

Analysing the Suitability of the Architecture (SA)

- Is there a common understanding of the architecture and its goals?
- Are the architecture concepts suitable for achieving these goals?
- Are architecture decisions subject to considerable risks?
- What are the most important tradeoffs and are they suitable?

Analysing the inner Quality of the Architecture (QA)

- Does the architecture comply with common best practices?
- Is the architecture understandable, comprehensible and simple?
- Is there high conceptual integrity?
- Is the architecture well-communicated and established?

Checking if Implementation and Architecture are in sync (IC)

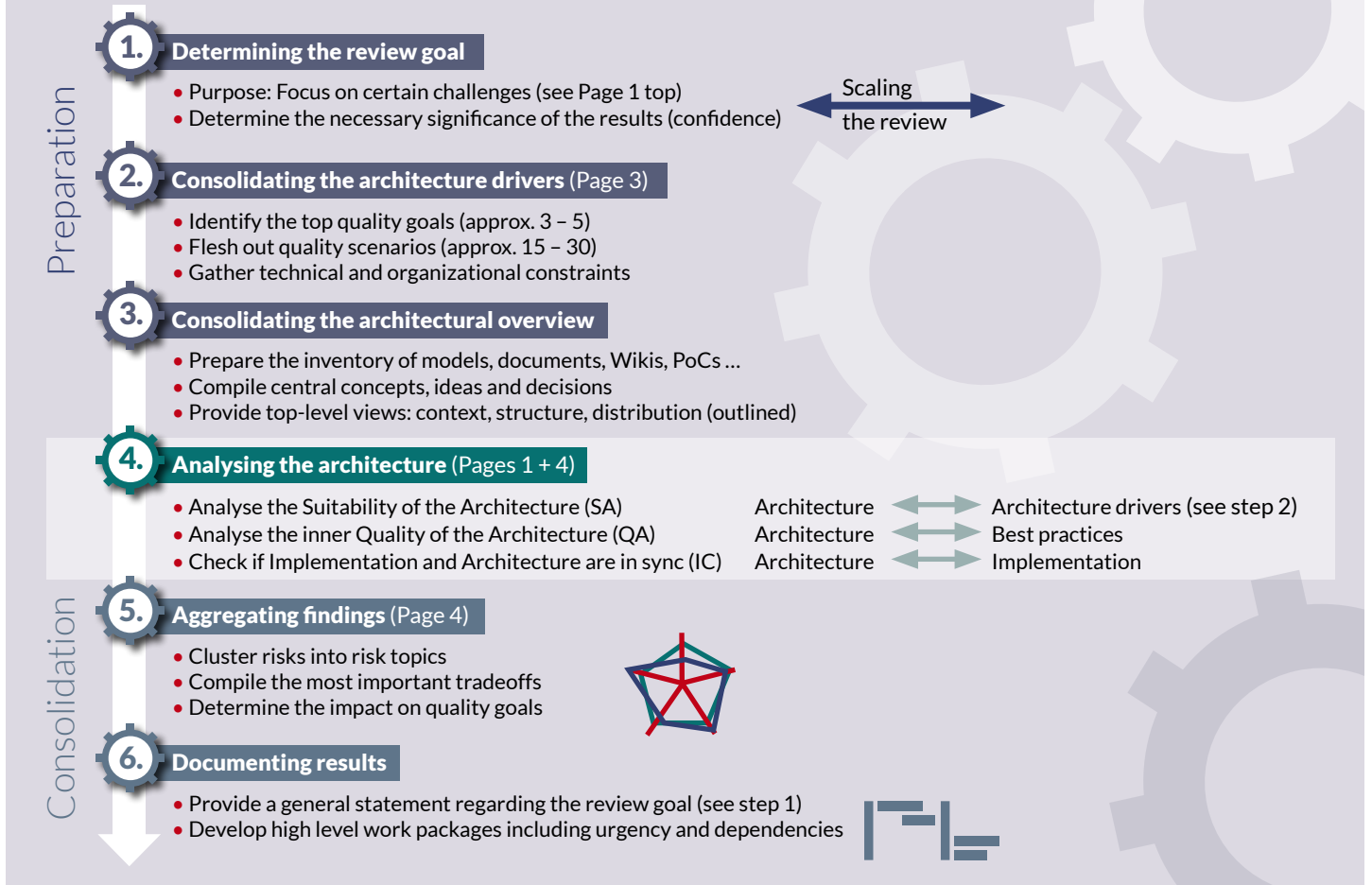
- Are architecture decisions apparent on code level?
- Are there weak spots in the implementation?
- Do code structures correspond to the communicated architecture?

Scenario-based
(Check the architecture against its drivers)

Possible ad-hoc
(Preparation can be very minimal)

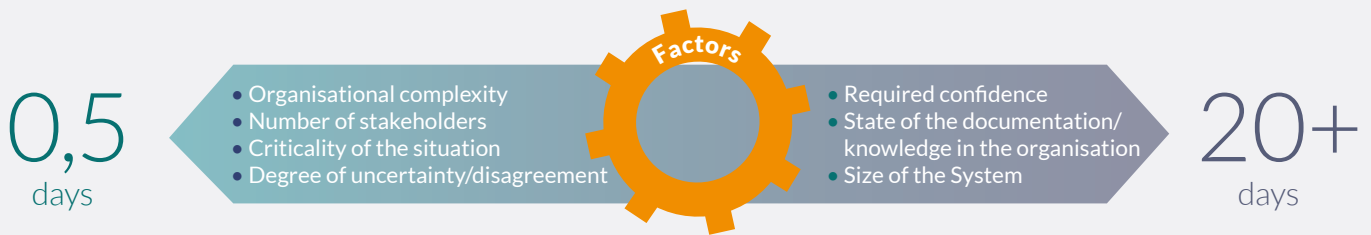
Tool-supported
(see Cheat Sheet #2 – Quantitative Analysis)

The typical review process



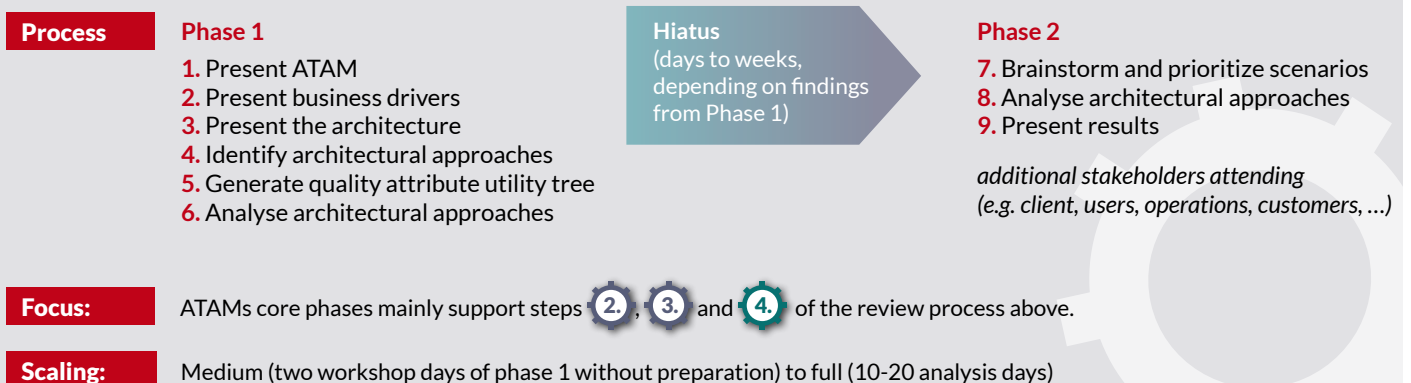
Scaling of Reviews

Many factors determine the size of a reviewing effort. A few important ones are:



ATAM – Architecture Tradeoff Analysis Method

ATAM is a theoretical basis on which many architectural reviews are built, if the Suitability of the Architecture (SA) is of concern. The architectural evaluation method was developed at the Software Engineering Institute (SEI) of the Carnegie-Mellon University and was originally published in 2000. The architecture is broken down and analysed in two workshops (the 'core-phases'). Due to its complexity, smaller reviews tend to use only parts of ATAM (see scaling).



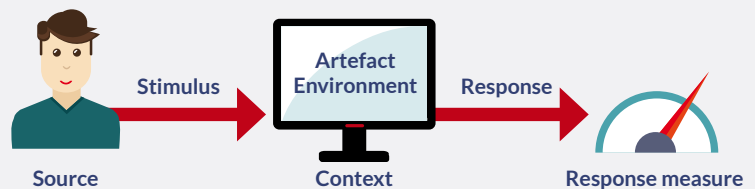
Roles in Architecture Reviews

Role	Possible filling	Tasks	Relevant in analysis type
Reviewer	Developers or architects (that are not part of the project/product-team), external experts	Define the review focus, pose questions, identify risks, contribute own experiences	SA, QA, IC
Architecture representative	Developers, senior developers, architects, testers	Present decisions and solutions, clarify problems, illustrate tradeoffs	SA, QA, IC
Stakeholder	Product owner, customer, specialist department, (project) managers, enterprise architect, operations	Representing drivers and constraints, validate tradeoffs, appreciate topic relevance	SA, (IC)
Moderator	Employees who are not part of the project, external evaluators	Organisation, preparation, moderation, ...	SA (workshop parts)

2. Architecture drivers

Quality Scenarios

Quality Scenarios specify quality requirements and have a typical structure that is quite similar to User Stories:



Types of scenarios

Various types of scenarios are available for specifying quality characteristics:

Use case scenarios

Characteristic: "Normal" usage of the system
Typical sources: User action (as defined in the requirements)
Often used for: usability, efficiency, functionality

Growth scenarios

Characteristic: Something is added or changed.
Typical sources: Greater load, new features, technical migration, ...
Often used for: Maintainability, scalability, portability, compatibility

Exploratory scenarios

Characteristic: Something "unforeseen" occurs.
Typical sources: (External/Partial) system failure, overload, incorrect use, ...
Often used for: Reliability, security

A case worker records the damage. With one click he opens the according screen and is finished within 3 minutes.

The number of users increases by 100% in one year. The system remains efficient and robust without personnel efforts.

A technical breakdown shuts down parts of the system landscape (up to 30%). Customers do not experience downtimes.

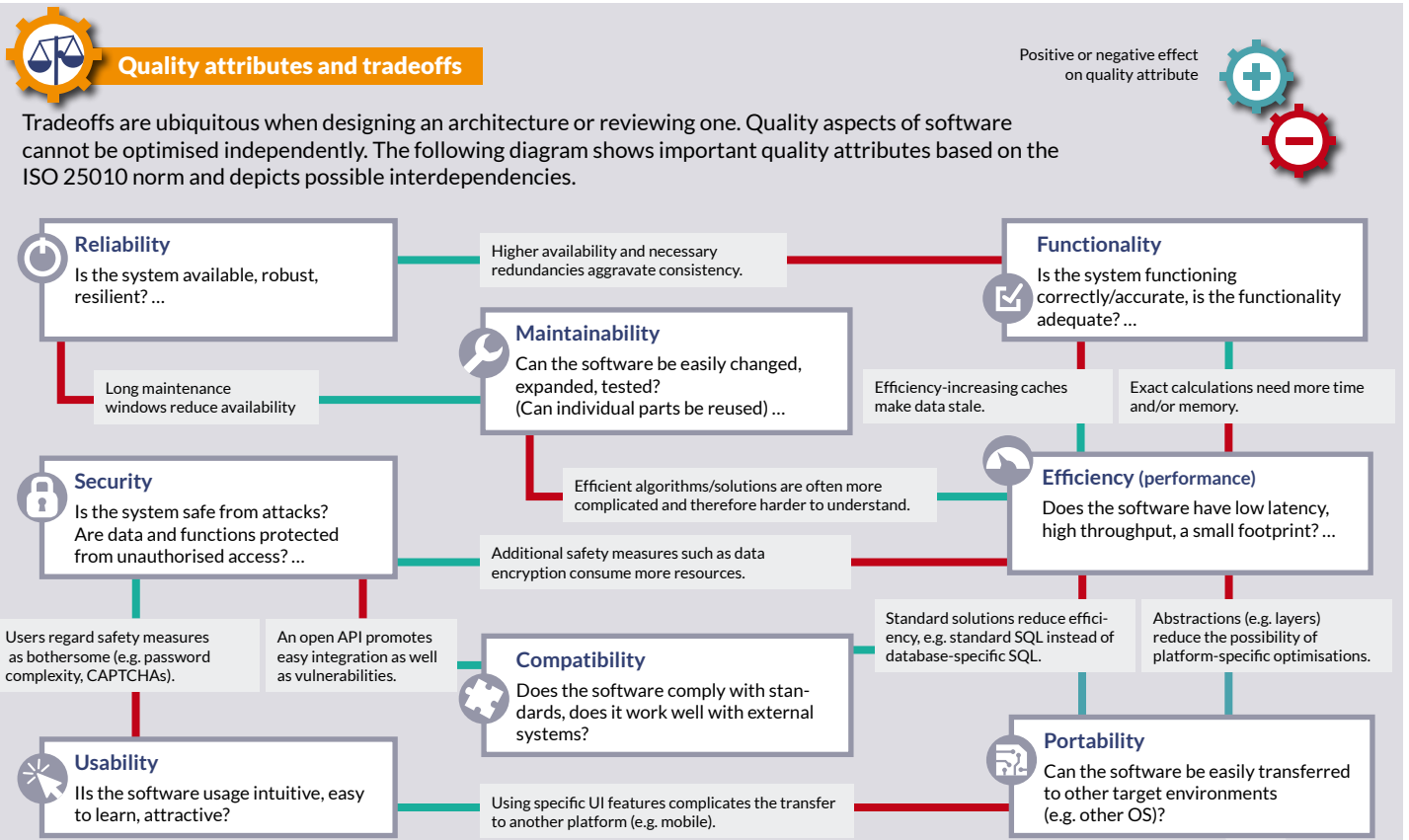
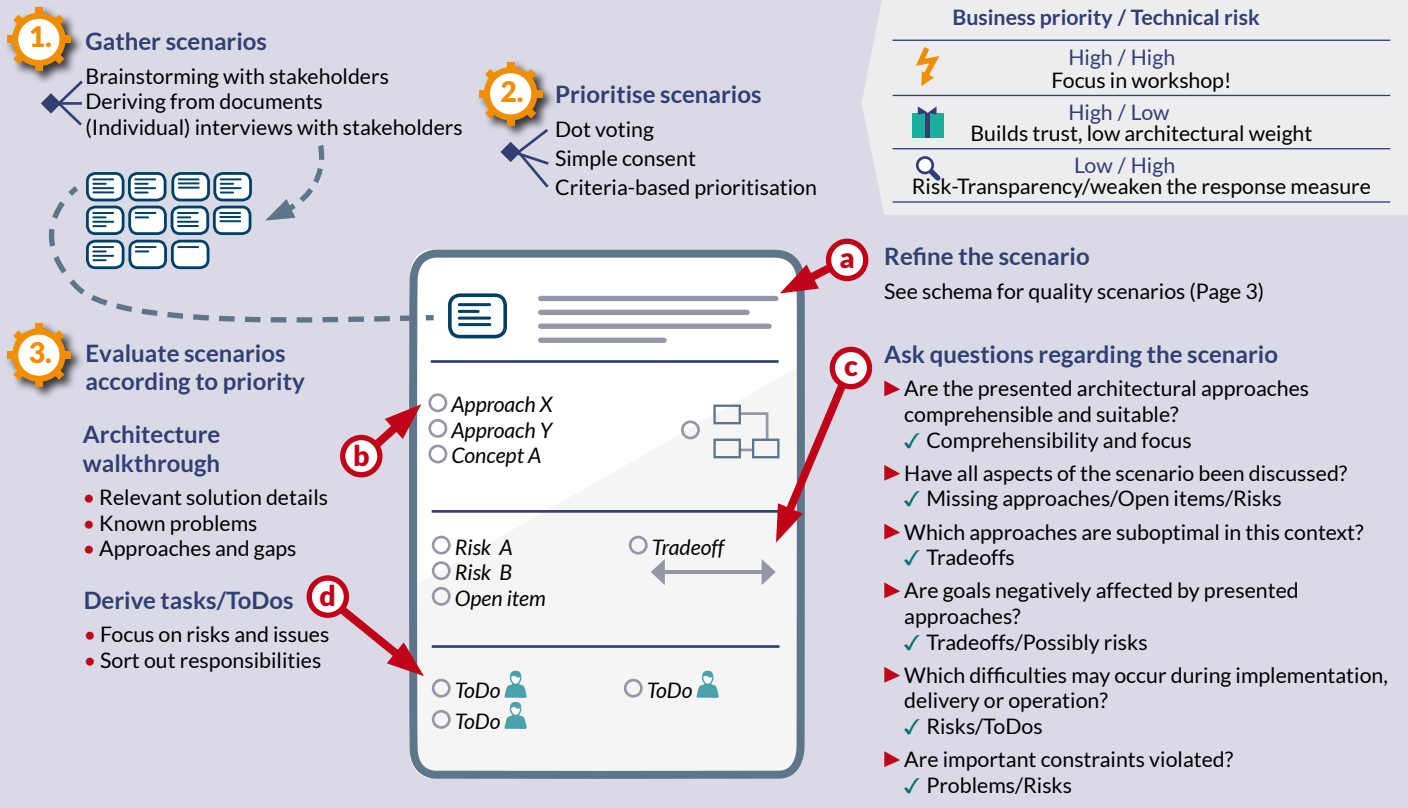
Constraints

In addition to quality requirements, constraints are the most important drivers of architectural design. It is important to determine which constraints have been overlooked, violated or are prospectively in danger of being violated.

Technical constraints	Organisational constraints	Legal constraints
<ul style="list-style-type: none"> • Hardware restrictions • Platform restrictions • Base frameworks • Operations guidelines • Programming guidelines • Other technical restrictions 	<ul style="list-style-type: none"> • Organisational structure • Project size • Methodology • Budget • Time • Organisational standards • Available Know-how 	<ul style="list-style-type: none"> • Liability issues • Data protection • Verification obligations • Audit security • Regulatory specifications • International legal matters

During evaluation workshops

The Suitability of the Architecture (SA) is often evaluated in workshops as a scenario-based walkthrough (as seen in ATAM) – architectural approaches are discussed and analysed in the light of the most important drivers.



We look forward to your feedback: spicker@embarc.de

<http://architektur-spicker.de>

embarc
Software Consulting GmbH

<https://www.embarc.de/>
info@embarc.de

SIGS DATACOM
FACHINFORMATIONEN FÜR IT-PROFESSIONALS

<https://www.sigs-datacom.de>
info@sigs-datacom.de