architecture CHEAT SHEET

Essential knowledge for software developers squeezed into a few pages

FIND MORE CHEAT SHEETS ONLINE:
All architecture cheat sheets are available as pdf for free:
www.architektur-spicker.de

NR. 12

Lightweight Software Reviews with LASR

Reviews uncover weaknesses in software solutions and confirm technical and architectural ideas. This cheat sheet describes a scalable evaluation approach that delivers results quickly.

IN THIS EDITION

What's it all about? (challenges)

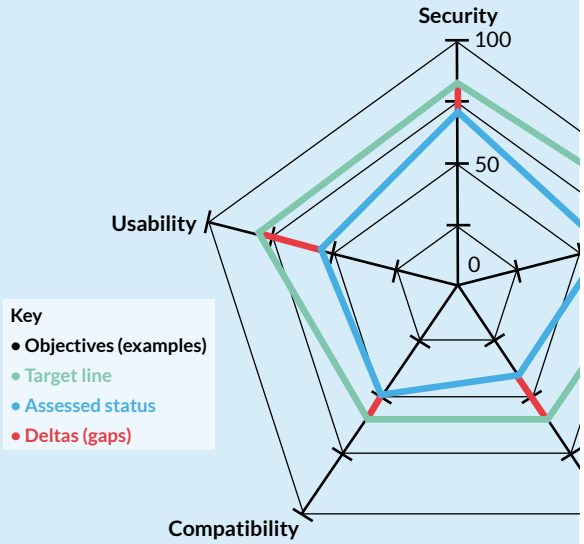
- Conventional evaluation methods for software are often difficult to learn and apply. Are there simpler approaches that are easily adopted?
- Profound analysis methods often require many participants. Can reviews be carried out in smaller groups or even alone?
- Many evaluation methods only deliver raw results that subsequently need to be analyzed, summarized and linked back to goals. Is it possible to get Management-ready results early on?
- The more critical the situation, the more complex the review. How can you adjust the scope and depth of a review appropriately?

A lightweight approach

LASR (Lightweight Approach for Software Reviews) is a structured method for efficient software reviews.

Characteristic features

- Leaner and quicker than ATAM, but still goal-oriented
- Can be carried out within your team or even by yourself
- Provides a quick first result (in a few hours)
- Allows for optional deep dives to increase the results maturity step-by-step
- Summarizes findings in a spider chart for easy communication (see illustration on the right)
- Offers extensive support material to make the application straightforward



LASR is roughly divided into two modes (I. and II.), the second of which is optional:

I. LASR Review

Provides a quick first review result, that can be refined in the last step 4.

→ *This mode is detailed on pages 2 and 3.*

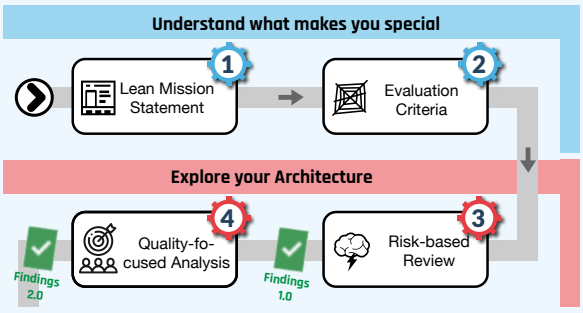
II. Optional Deep Dive with LASR+

Provides maturity and depth to the review result if needed. Includes code level checks and an organizational perspective.

→ *This mode is detailed on page 4.*

What are the key tasks? 1 2

- Condense the system's vision into specific claims that work as a mission statement
- Identify the top 3-5 quality attributes of the system
- Determine the respective target levels



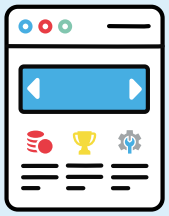
What are the key tasks? 3 4

- Identify the most significant risks to the system's success
- Find the deltas to the defined target levels by quantifying said risks ('gaps')
- Analyze controversial gaps (or their absence) in a quality focused evaluation

Understand what makes you special

1 Lean Mission Statement

The first step of the review condenses the product's vision into a lean mission statement. It consists of a set of 'claims' that make the system's goals easy to grasp.



Goal of this step: A concise and convincing description of the system's purpose

The landing pages of well-known software products provide good inspiration for this step.

Checklist for the mission statement

- ○ Is the name of the system featured early on?
- ○ Is it clear who benefits from the software?
- ○ Do you name the key features?
- ○ Do you state the unique selling points?
- ○ Do you use active speech?
- ○ Do you describe the system like it already reached its objectives? (no 'will', 'should', 'could' or 'perhaps'...)
- ○ Is the content still readable when printed on a DIN A5 sheet?

2 Evaluation Criteria

Quality requirements are essential in serious software reviews. Therefore, the most important quality attributes and assotiated objectives form the benchmark by which the software is measured. LASR sets its focus on the top 3-5 quality attributes.

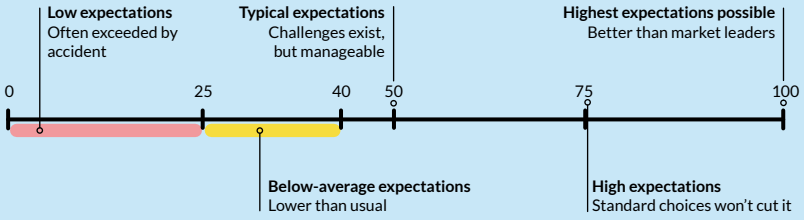
Usage perspective Who works with the software. Primarily: (end) users	Organizational perspective Who commissions and is responsible for the software. Primarily: Company	Implementation perspective Who builds and runs the software. Primarily: Development and operations
Usability	Cost Efficiency	Maintainability
Reliability	Sustainability	Scalability
Performance Efficiency	Auditability	Operability
Functional Suitability	Safety	Compatibility
Security		Portability

→ LASR defines 14 quality attributes and 3 primary stakeholder perspectives. The 9 standard attributes of ISO 25010 are included with minor modification.

→ See page 4 'Additional information' for support materials that help you finding your top 3-5 .

Quantify quality objectives

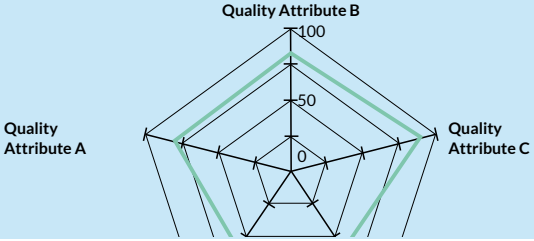
Each of the top quality attributes is assigned an axis in a spider chart – the LASR result diagram. The expectations are plotted on these axes.



The LASR result diagram quantifies review results and serves as a driver for in-depth analysis.

A clear benchmark for the review after step 2

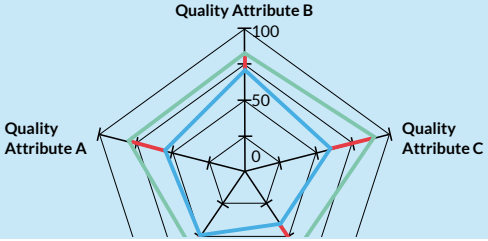
The quantified objectives per quality attribute result in a green line that illustrates the review benchmark.



Risk-based Review

System status and gaps after steps 3 and 4

The evaluation steps of the core review add a system assessment and identify gaps:



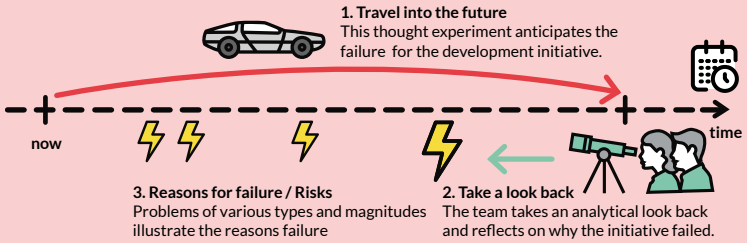
Explore your Architecture

3 Risk-based Review

Use a pre-mortem like approach to identify risks

What is a pre-mortem?

Pre-mortems take a critical look at systems. They envision a hypothetical future in which your development initiative has failed. From this perspective they ask: What are the 'reasons for failure'? From today's perspective these reasons represent possible risks.



'Pre-Mortem on Steroids'

- LASR uses 8 risk areas with 30+ standard risks to direct the brainstorming of potential problems. → *See box on the right*
- Identified problems are mapped to quality attributes and quantified on the corresponding axis.

Workshop activities

1. Identify relevant risks (brainstorm with the help of LASR standard risks)
2. Link the identified risks to your quality attributes
3. Assess the risks in terms of likelihood and impact
4. Quantify the deviation from your review benchmark determine the 'gaps'
5. Plot your assessed system status line (blue line)
   → *See page 2, bottom right*

The 8 LASR risk areas

Software solution: Unsuitable, complex or immature solution aspects

Expertise and experience: Missing or isolated knowledge (technical/domain-specific)

Objectives and expectations: Unrealistic or inconsistent goals or expectations

Platforms and external systems: Unstable, error-prone, unsuitable external systems, licensing issues, ...

Existing solutions or solution parts: Fragility, lack of tests, documentation or understanding

Deployment and production: Slowing processes or roles, organizational boundaries, obstructive policies or standards

Deployment and production: Blocking build or delivery processes, low automation, little feedback

Soft factors: disagreements, conflicts, lack of discipline or communication, inappropriate culture

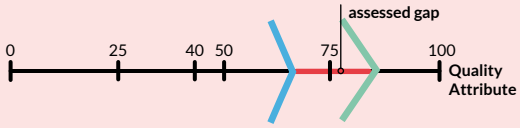
Support Material:
The LASR card set contains 32 standard risks for the 8 risk areas and adds 14 quality attribute cards.

Order your LASR card set here:
https://www.embarc.de/lasr-kartenset-bestellen/

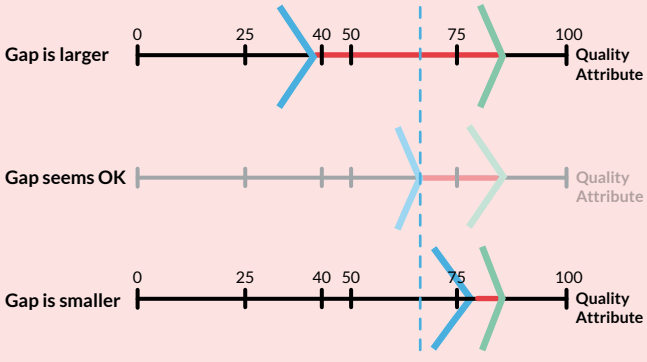
4 Quality-focused Analysis

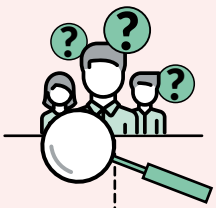
Refine the first result of the risk-based review (step 3) by using qualitative analysis tools. Limit your efforts to quality attributes where the LASR result diagram seems off.

3 Result of the Risk-based review



4 Possible results of the Quality-focused analysis:





Top down: Break down & evaluate quality attributes

- Find important quality aspects for the attribute in focus
- Prioritize the found quality aspects (looking for high technical risks)
- Find relevant solution aspects and discuss their impact on quality aspects

Bottom-up: Find & map problems

- Brainstorm existing weaknesses of the solution
- Find relevant test results and code smells
- Map the identified problems to quality attributes and specific aspects

Optional Deep Dive with LASR+

5 Maturity Boost

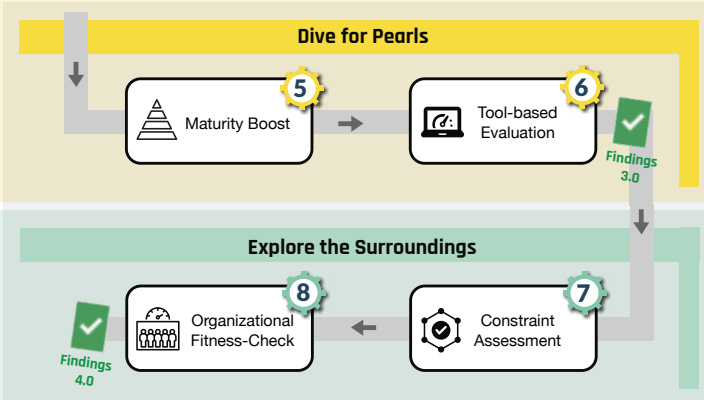
Analytically assess and expand on previous results, like quality goals, solution approaches or risks.

What are the key tasks?

- Organize and consolidate the results of the previous LASR steps
- Identify and prioritize gaps in the analysis so far

Tools and practices

Informal architecture overview, solution strategy table, simple utility tree



6 Tool-based Evaluation

Ensure that architectural ideas are apperent on source code level. Replace 'gut feeling' with reliable metrics.

What are the key tasks?

- Match source code constructs with your structuring ideas
- Perform specific measurements (e.g. response times, availability)

Tools and practices

Perform structure and dependency checks, static code analysis, dynamic checks (e.g. of load behavior)

When to apply?

- If the results from the core review appear too generic or disorganized.
- If large gaps have been identified where measurements really help to evaluate (e.g. performance).

When to apply?

- If a violation of guidelines is suspected in earlier steps.
- If you assume that the solution does not fit the organization's structure or culture.

8 Organizational Fitness-Check

Link the software solution to the structure, processes and culture of the organization.

What are the key tasks?

- Align software structure and team structure ('Conway's Law')
- Check the suitability of the software architecture for the existing development approach, including roles and processes
- Clarify whether the market/domain dynamics match the development dynamics

Tools and practices

Frameworks for organizational development and dynamic scaling approaches such as Team Topologies, unfix or LeSS

7 Constraint Assessment

Ensure that the solution complies with all relevant constraints.

What are the key tasks?

- Find and evaluate relevant constraints on business and technical side
- Check compliance with the constraints found: Mark as OK or as violation.
- Challenge violated constraints

Tools and practices

Stakeholder interviews, review of relevant guidelines, checklists (see further information)

Further information

Support material

→ **Designed specifically for the LASR method** to support the generation of results and assist facilitation. Downloadable card decks, checklists, ...
→ lasr-reviews.org/community/

E-Book

→ **Stefan Toth, Stefan Zörner Reviewing Software Systems** with the Lightweight Approach for Software Reviews – LASR Leanpub, June 2025, English
→ https://leanpub.com/reviewing-software-systems

The authors of this cheat sheet

→ Stefan Toth | Contact: stefan.toth@embarc.de
→ Stefan Zörner | Contact: stefan.zoerner@embarc.de

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

https://architektur-spicker.de

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

SIGS DATACOM FACHINFORMATIONEN FÜR IT-PROFESSIONALS https://www.sigs-datacom.de info@sigs-datacom.de