architecture CHEAT SHEET

Conceptual development expertise in a condensed format

FIND MORE CHEAT SHEETS ONLINE:
Free pdf-Download of the architecture cheat sheet collection:
www.architektur-spicker.de

NO. 1

IN THIS ISSUE

- Which ingredients are part of an Architecture overview?
- Which forms prove successful in which situations?
- How do you prepare an overview?

The Architecture Overview

An Architecture overview illustrates the central solution approaches of your software architecture - in compact form.

Challenges

- Team or project members (e.g. developers) lack a high-level-overview of solution approaches in order to work effectively..
- New team members who want to participate in development are not able to find their way within the architecture.
- Decision-makers and other stakeholders lost confidence or do not fully trust the solution.
- Colleagues outside of the team are interested in solution approaches, but find no or only very detailed information that makes it difficult to provide a quick overview.

Content of an architecture overview

Work in small batches! Produce independent "ingredients" that you can combine into various forms and refine iteratively if needed.

Forms

Various forms are possible for an architecture overview depending on the target group and communication channel.

- **Architecture wall:** Large-format, modular posts on a wall in the project room that are accessible to everyone
- **Architecture flyer or poster:** Small handout, e.g. DIN A4 printed on both sides, folded 2-3x or produced in a larger format (e.g. DIN A1) as a poster for widespread dissemination
- **Architecture portal in Wiki:** Entry page(s) in Wiki that guide interested persons through the content.
- **Concise document:** Structured text, augmented with illustrations, maximum volume 20 pages
- **Slide set:** 10-15 slides for supporting an architecture presentation
- **Video:** Recording of an overview with sound and image, possibly combined with slide set

Ingredients

What should be included? The ingredients of this illustration are described on the next page. Don't worry; normally you don't need all of these.

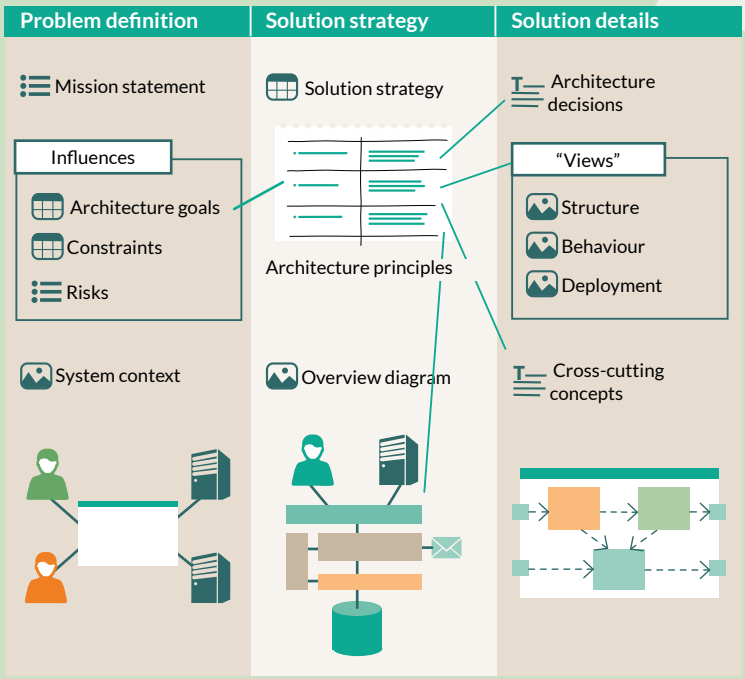

Illustration 1: Overview of important ingredients

architecture CHEAT SHEET 1

Ingredients describing the problem definition

… define the task and describe the goals for your software system as well as the central influential factors for your solution.

Ingredient and typical format	Description
Mission statement	Concise illustration of the task. What is the purpose of the system (or of the component, framework, …)?
System Context	Blackbox-visualisation of the system with its most important external systems and directly interacting users.
Architecture goals (top 3-5)	The most important quality requirements for the system ("-ilities"), including their motivation
Essential constraints	The most important technical or organisational constraints that must (or had to) be kept in the design process
Greatest risks	Possible harmful events that influence (or have influenced) the software architecture

Tool: Guiding questions for the mission statement

The following questions support you in preparing your mission statement:

- What is the purpose of the system?
- What is the central selling point/ usage argument? ("claim", "slogan")
- Who does benefit from it?
- What are the essential system features?
- How does it differ from competitor products or the previous version?

Ingredients describing the solution strategy

… bridge the gap between problem and solution (the "big picture").

Ingredient and typical format	Description
Solution strategy (table)	Two-column table containing architecture goals and their supporting architecture approaches with references to the overview diagram and solution details
Architecture principles	Principles that provide orientation for all decision-making processes (e.g. preferences, "Prefer XY to Z")
Informal overview diagram	Visualisation of the solution with emphasis on central Architecture approaches (e.g. style, pattern, structure, ...) – more likely no UML

Tool: Categories for architecture approaches in the solution strategy table

Typical content in the right column with an example and suitable goal (in parentheses)

Architecture decisions
E.g. usage of an application server cluster (goal: high availability)

Architecture styles
E.g. Microservices (quick adoption of new technological trends)

Architecture patterns
E.g. layered architecture (easy exchangeability of client or simple porting of the solution)

Architecture principles
E.g. prefer standards of proprietary solutions (low maintenance costs)

Concepts
E.g. caching concept (efficiency, quick response times)

Approach
E.g. user-centred design (intuitive usability)

Ingredients describing solution details

… describe solution approaches in detail and make the architecture comprehensible. The overview only shows this content in excerpts.

Ingredient and typical format	Description
Architecture decision	Explain a central, far-reaching decision, e.g. . for technology or framework usage, including alternatives and evaluation criteria
Structure	Technical and/or domain-specific decomposition of the system
Behaviour	Central dynamics within the system (walk-through, fail-over, …)
Deployment	Visualisation of the target environment, system commissioning and operation
Cross-cutting	Description of a comprehensive idea, valid for the whole system (e.g. persistence concept)

Format key

- Enumerated list
- Diagram/Graphic
- Formulated text, if necessary enhanced with images, etc.
- Table

How do you proceed?

Use a matrix to link target groups to ingredients and produce architecture overviews in different forms if necessary.

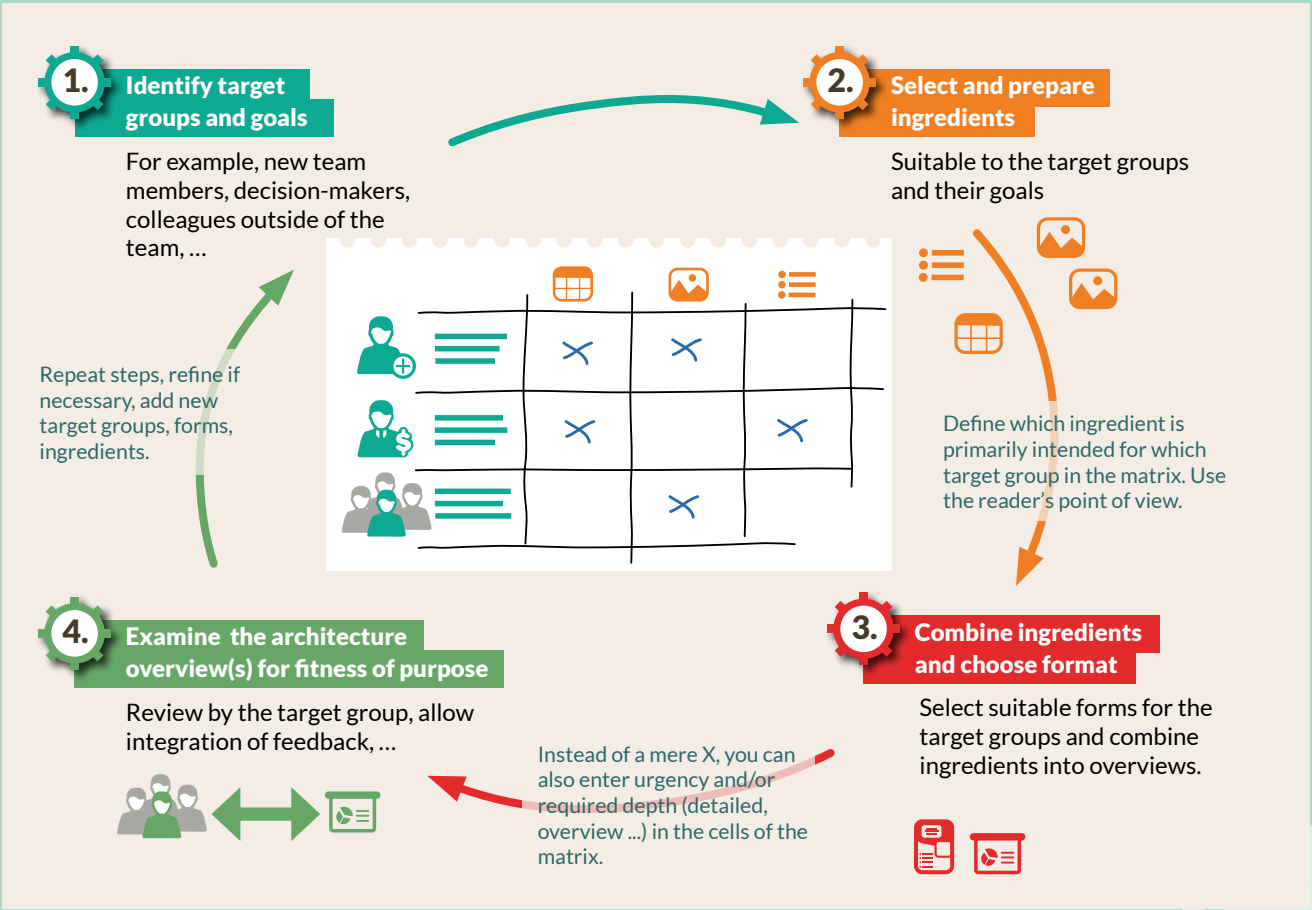

Illustration 2: Creating an overview step by step

Tool: Exemplary structure for a slide set

Illustration 1 (also) provides a possible structure for your architecture wall. This table suggests a structure for a slide set that supports you in the presentation of your architecture.

Section	Possible content
I. Task definition	**Mission statement** **architecture goals** **System context** Challenges, pain points Essential constraints
II. The "big picture"	**Solution strategy** (table) Informal overview diagram Architecture principles
III. The solution in detail	**Architecture decisions** Diagrams (structure, deployment) Cross-cutting concepts Demo, walk-through
IV. Conclusion and outlook	**Open items** **Next steps** Discussion Further information What questions do you have?

(Bold content is especially important.)

Tool: Strengths and weaknesses of different forms

The following table states exemplary criteria with a rough assessment for various forms. The selection always depends on your specific context!

	Document	Slide set	Wiki	Architecture wall	Poster/Flyer
Initial effort	Low, few ingredients and good structure in the beginning	Low, few slides and good structure in the beginning	Medium, product selection/configuration may be necessary.	Medium, suitable free wall and fitting culture is necessary.	Expectations for the first edition often already very high
Subsequent change and expansion	Can be easily modified and supplemented with ingredients, suitable tooling assumed	Can be easily modified and supplemented, versioning possibly difficult	Easy, versions and tracking of changes, but depends on the product	Possible at any time, but tracking changes difficult	Difficult since space is limited and production elaborate
Developer acceptance	Low, OK for reading if concise and appropriate for the target group	By itself only medium, higher in presentations/ discussions	Rather high, at the same time prejudices due to frequent "rotting" and disorganization	High, since the wall is appealing for feedback and cooperation.	High if executed well; exciting and unfamiliar format
Manager acceptance	High, often meets expectations	By itself only medium, higher in support of presentations	Low, somewhat better for entry pages according to the target group	Medium, rather seen as work equipment of development	High if graphically appealing and clearly designed
Communication in spatially distributed teams	Easy distribution, but does not promote sharing by itself ("one-way street")	Medium, slides by themselves often not convincing, additional presentations are elaborate.	Comparably well-suited for collaborations in distributed teams	Poor, distribution via photos may be possible	Can be distributed easily to different locations, flyers also, but feedback difficult

The forms do not exclude each other. For example, start with an Architecture wall and derive other items subsequently.

Colour key: Positive Neutral Negative

Further information

Links and literature

- Simon Brown: Software Architecture for Developers - Volume 2 (Visualise, document and explore your software architecture), Leanpub 2017, **https://leanpub.com/visualising-software-architecture**
- Paul Clements et al: Documenting Software Architectures, Addison Wesley, 2nd revised Edition 2010
- arc42, Template for architecture descriptions, **https://arc42.org**
- 139 tips how to use the arc42 template, **https://docs.arc42.org/home/**

Author of this cheat sheet

- Stefan Zörner is a software developer and architect at embarc in Hamburg. He has written a renowned German book about documenting software architecture.
  Contact: stefan.zoerner@embarc.de

Examples

- arc42 by Example - Software Architecture Documentation in Practice, Leanpub, 2023, **https://leanpub.com/arc42byexample**

- Architectural description of a chess engine, structured according to arc42, **https://www.dokchess.de/en/**