

cards42

**Cards for Analyzing
and Reflecting on Doomed
Software**

**Activity cards for
software architects
English edition**

cards42.org/en

INOQ

About INNOQ

INNOQ is a technology consulting company. We provide honest advice, think innovatively, and are passionate about software development. The results are successful software solutions, infrastructures, and business models.

Many ideas for these cards originated from our fruitful collaboration with our customers.

Website: www.innoq.com

Imprint and contact

innoQ Deutschland GmbH
Krischerstraße 100
40789 Monheim am Rhein
Germany
Telephone: +49 2173 3366 0
Email: info@innoq.com

License

This work is licensed under Creative Commons - Attribution-ShareAlike 4.0 International (CC BY-SA 4.0).



INNOQ

ResourceAllocationExpectation

Suppose you are allowed to allocate infinite resources (time, money, people, etc.) to your software project. **What would you do?**



Architecture quick check

How well do your architectural approaches match the most important quality goals?

Top 3 quality goals

1.
2.
3.

Key architectural approaches

- a.
- b.
- c.

Evaluation

For each key architectural approach, assign a number that shows how well it supports the most important quality goals ----->

Rating:

0 = no supported at all
5 = maximum supported

	a.	b.	c.
1.			
2.			
3.			

Architecture Decision Record

Title

Status

- proposed accepted deprecated

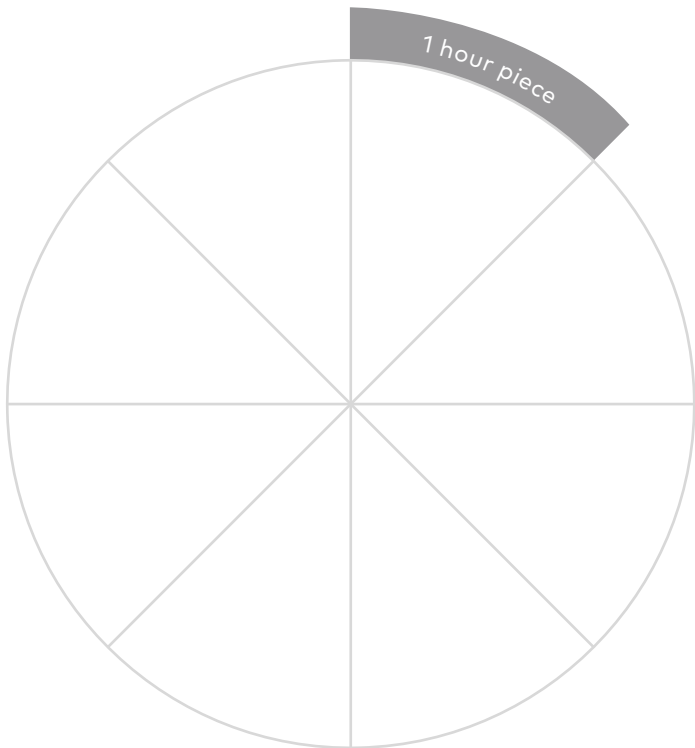
Context

Decision

Consequences

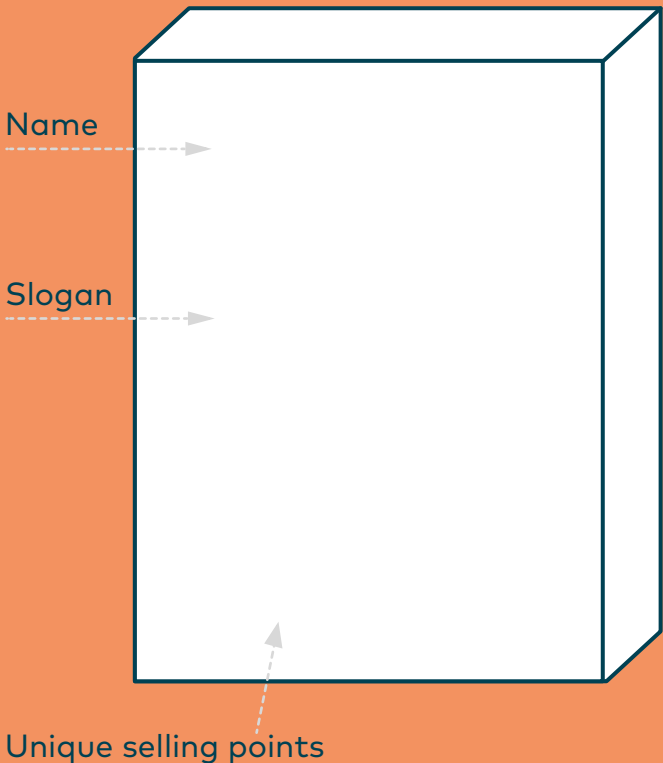
The 8-hour cake

What parts does your workday consist of?
Which ones do you like? Which ones would
you like to get rid of?



Your product box

Design the retail package for your software!



Do you work according to common development standards?

Good practice



Version control system

Build system

Continuous integration

High test coverage

Code reviews

Static code analysis

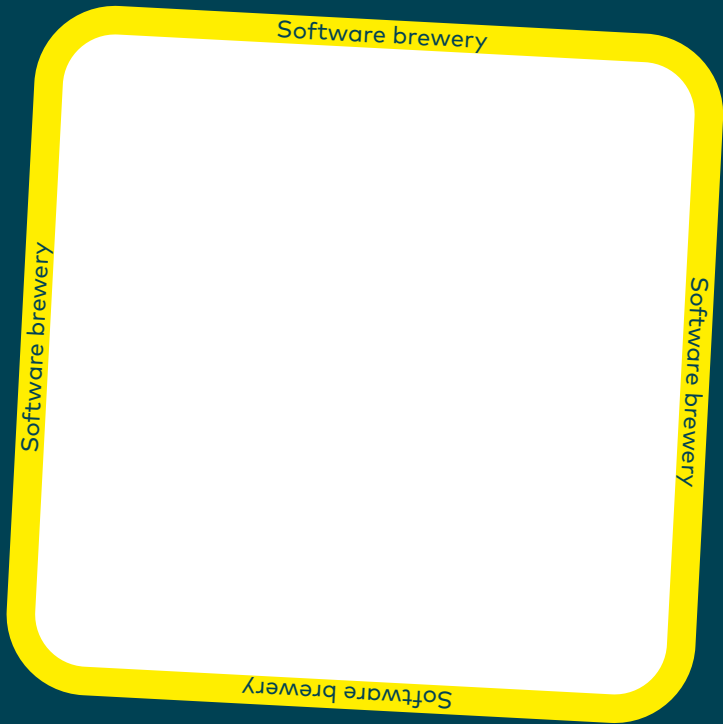
Architecture evaluation

Retrospectives

Architecture documentation

Architecture on a beer coaster

Sketch your architecture in the small area of the coaster! How well does that work?



The unknown familiar

Find a stakeholder of your software system whose behavior you understand the least.

Slip into their role/mind!

That's me:

What I want to achieve with the system:

.....

.....

.....

.....

That's important to me

1.

2.

3.

My most painful problem

Name

Short description

Pain scale



Problem cost estimation

X

 [;]

Frequency of
occurrence

Costs (max)

Costs (min)

Root cause analysis

Ask "Why?" five times, starting from your initial problem (= may be a symptom)!

Symptom	<input type="text"/>
1. Why?	<input type="text"/>
2. Why?	<input type="text"/>
3. Why?	<input type="text"/>
4. Why?	<input type="text"/>
5. Why?	<input type="text"/>

Success in the future

Imagine your software system is extremely successful in the future. Why did it become a success? **What will you do for that today?**



2.

3.

1.

How bad are you talking about documentation to yourself?

Lazy excuse

Yes!

- // Documentation is always totally outdated!
- // Documentation is always too complex!
- // It won't help anyway because nobody reads it!
- // A huge documentation is not manageable!
- // I don't know where I can find all those documents!
- // I don't have the tools to write documentation!
- // I need UML for documenting, which I don't know!
- // Nobody understands what I'm writing down!
- // The documentation is too superficial for me!

Your waste case scenario

What needs to happen that you chuck your software system in the **dustbin** ...



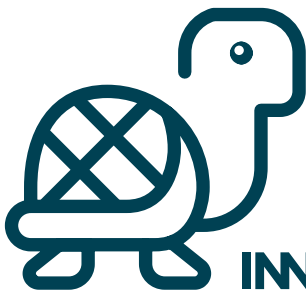
Positioning

What are your software system's strengths, weaknesses, opportunities, and (un-)conscious risks?



Technical debt

Draw a circle on the back of the tortoise, symbolizing the burden of your **technical debt** that slows you down. Write down the reasons for it in the circle!



Avoid the #shitstorm!

Find out which people are involved in your software project. How much do they influence your system? How can you involve them? How can you change negative views?

Big influencer,
negative views

Big influencer,
positive views

Small influencer,
negative views

Small influencer,
positive views

Code smell monster

You've caught a code smell monster! What is its name? Draw a picture of the monster!



Checklist for a better documentation

Questions

OK

Is the audience of the documentation defined?

Are the goals of the documentation clear?

Is it defined which aspects should be described?

Is it clear how detailed the documentation should be?

Do the documentation's content and the current state of the software system match?

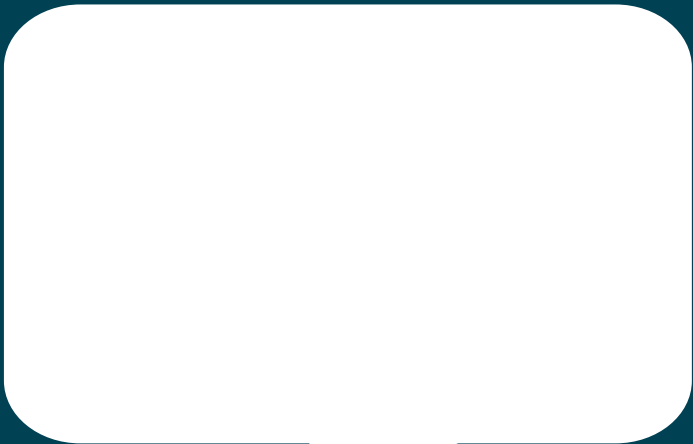
Is the audience familiar with the used terms, notations and concepts?

Are the persons known that will later be responsible for maintaining the documentation?

Can the future maintenance team keep the documentation current without big efforts?

What would _____ do?

Are you stuck on a problem? What would one of your idols do in your situation to move on?

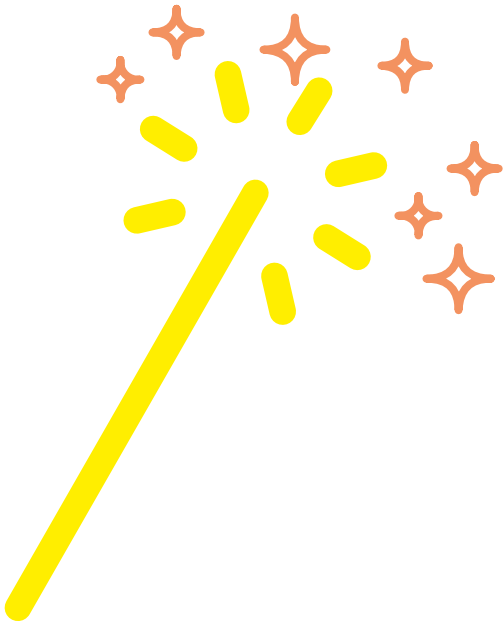


My idol:



The conjurer's wand

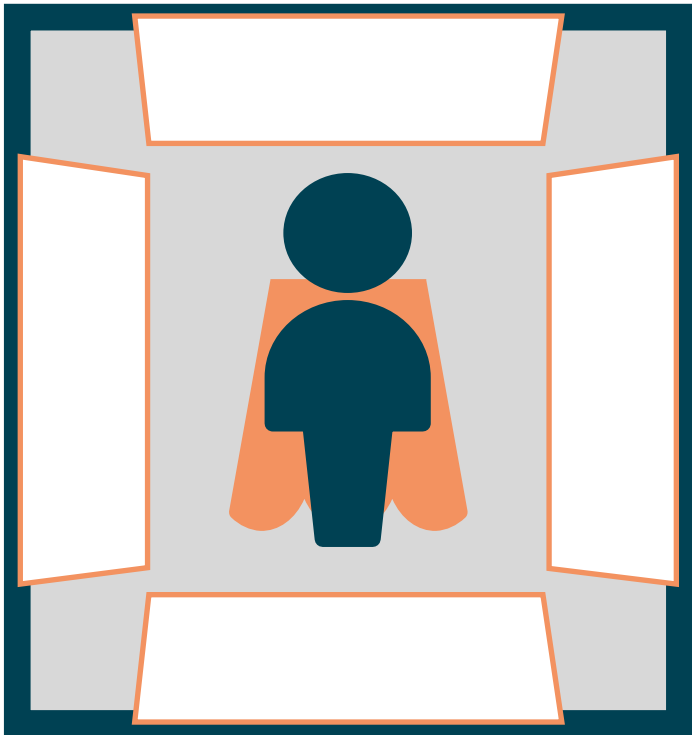
This magic wand solves all your problems!
You just have to swing it hard enough!



Sorry, there are no golden magic wands.
Problems have to be solved with hard work.
But swinging around the card did feel good, didn't it?

The Superarchitectodon

What are the four superpowers a software architect needs in your software project?



Dream project's checklist

To-do

Done

Problems to be solved are clearly defined



An easy-to-understand vision exists



Business goals and their value are clear



Most valuable features are known



Key stakeholders are involved



Software architecture is outlined



Main risks are identified



Scope and costs are known



Trade-offs and expectations are clarified

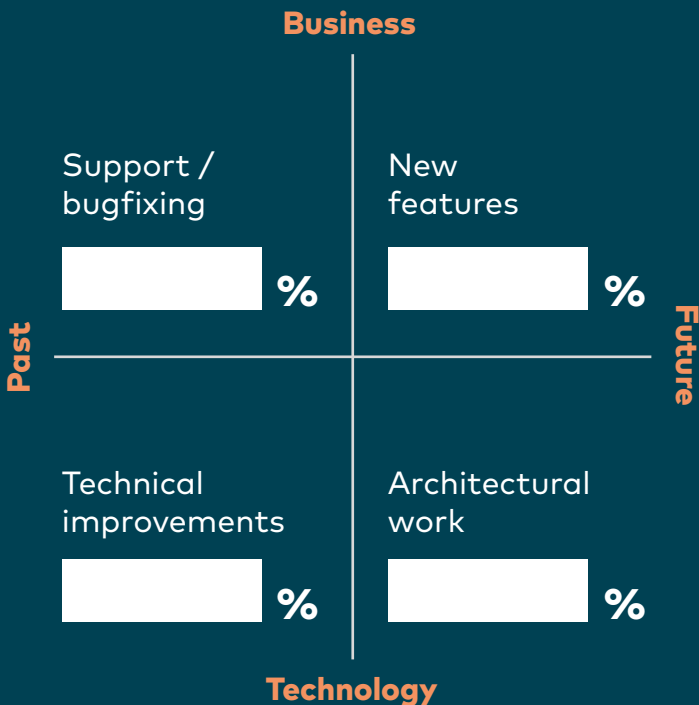


Time horizon is communicated



Investment sanity checker

How balanced are the investments in your software system? Is this reasonable?



Mini architecture documentation

Write down the most important aspects of your software architecture!

Business goals to be achieved

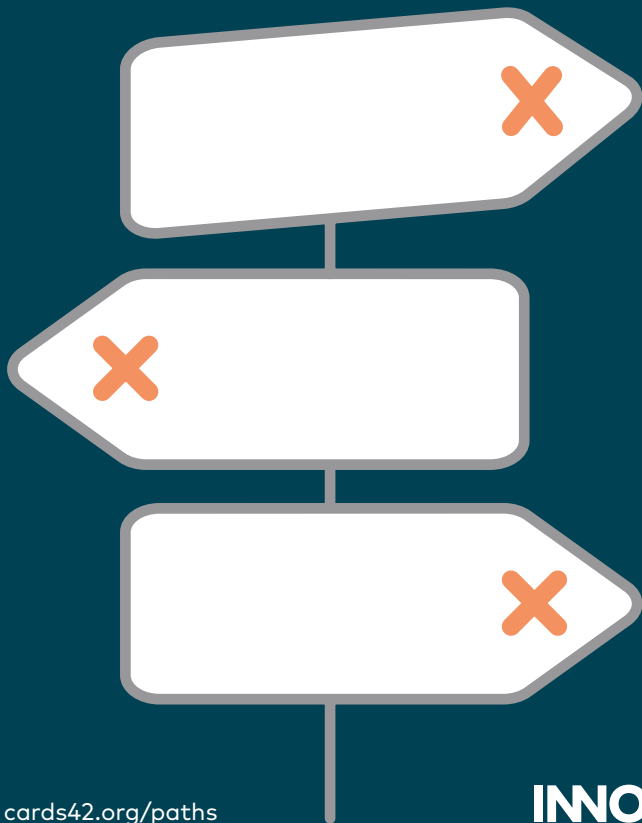
Top 3 quality goals (ordered by importance):

..... > >

Significant architectural decisions:

Paths not taken

Which options were deliberately not pursued? Write them into the signposts!
What were the reasons?



How to learn X?

You must have great expertise in at least one field. What is your special topic?

Your area of expertise:

What learning recommendations do you have for newcomers that want to become experts in this area, too?

My top 5 recommendations:

1.

2.

3.

4.

5.

Risk minimization

To avoid

risk



we take

measures



from which we hope to gain

effect



Businessability

You have to decide in favor of a certain technology and you are undecided between solution **A** and **B**? How does the comparison look from a business's point of view?

Rating 0 (worst) – 5 (best)	A	B
Popularity How well does the technology pay off positively on the employees' careers?		
Internal know-how To what extent can existing knowledge of the employees be used?		
External expertise How good is the availability of external consultants or service providers?		
Self-study How good are available literature and online learning resources?		
Training What training opportunities or classes are available?		
Start-up costs How are the costs for new introductions and re-training?		
Ramp-up time How long does it take before employees can work productively with the new technology?		
Total		

Technology roadmap

Clarify and communicate for your new or existing technologies in what stage of adoption they currently are.



Where is the value for the user?

Are you stuck in a technical discussion?
How much do the points you're talking
about help the users of your software?
Write down the main points ordered by
the value they provide to the users!



Directly
visible



Value for the users



invisible

A vertical orange arrow points upwards from the 'invisible' eye icon to the 'Directly visible' eye icon. To the right of the arrow are seven horizontal dashed lines for writing.



Feedback?

Did you like the cards? Can we improve something? Do you have new ideas?

Feel free to give us some feedback at cards42.org/en/#feedback!

Contributors

Markus Harrer (creation, card ideas, design, texts)

Sonja Scheungrab (redesign)

Tobias Erdle (website layout, texts)

Ben Wolf (card ideas, reviews)

Lisa Maria Moritz (texts)

Simon Harrer (card ideas)

cards42 online

Website: cards42.org/en

Twitter: [@cards42org](https://twitter.com/cards42org) [#cards42](https://twitter.com/cards42)

GitHub: github.com/innoq/cards42org



What is cards42 about?

We created cards42 to support the daily work with software systems. The cards provide food for thought for deadlocked situations and help shed new light on difficult challenges.

More information: cards42.org/en