

Design Matrix

Description of design challenge		Result of design decision	
Name:	<input type="text"/>	Date:	<input type="text"/> Status: <input type="checkbox"/> Approved <input type="checkbox"/> Rejected
Topic overview & Solution details:	<div></div> <small>If useful, link relevant documents</small>	Decided by:	<input type="text"/>
		Stakeholder:	<input type="text"/>
		Summary:	<div></div>

Simple	<ul style="list-style-type: none"> <input type="checkbox"/> Is the solution easy to understand (even in the future)? Is there a solution that is easier? <input type="checkbox"/> Does it avoid „clever“ magic and overly generic approaches? <input type="checkbox"/> Is the solution explicit so there is less room for misinterpretation or for ugly surprises? <input type="checkbox"/> Are there unrequested features we can omit? 	<div></div> notes	Powerful	<ul style="list-style-type: none"> <input type="checkbox"/> Is the solution foresighted enough? <input type="checkbox"/> Does it take non-functional requirements into account? <input type="checkbox"/> Is the solution generic and reusable? <input type="checkbox"/> Which parts will change in the near future? Which ones continuously? What should stay stable, what flexible? 				
	Abstract	<ul style="list-style-type: none"> <input type="checkbox"/> Does the solution fit into the big picture (e.g. uniformity, structure and architectural constraints)? <input type="checkbox"/> Is each part of the solution understandable on its own? <input type="checkbox"/> Are modules cohesive and is coupling low? 		<div></div> notes	Concrete	<ul style="list-style-type: none"> <input type="checkbox"/> Does the solution avoid duplication of functionality (also think about other modules and systems)? <input type="checkbox"/> Are there places we can refactor/optimize in the same breath? <input type="checkbox"/> Can the solution grow naturally over time? (e.g. allow further changes/refactorings) 		
		Pragmatic		<ul style="list-style-type: none"> <input type="checkbox"/> Does the solution provide value early on? <input type="checkbox"/> Does the solution really address the customer's goals/use cases? <input type="checkbox"/> Does the solution really fit to the timeline? <input type="checkbox"/> Can we use already existing Code (snippets, libraries, services)? 		<div></div> notes	Idealistic	<ul style="list-style-type: none"> <input type="checkbox"/> Is this the right solution? <input type="checkbox"/> Is it consistent with the rest of the system? <input type="checkbox"/> Is ensured that there are no workarounds or bad decisions that will produce serious problems later?
				Robust		<ul style="list-style-type: none"> <input type="checkbox"/> Is the solution hard to misuse? <input type="checkbox"/> Are the chances for something to go wrong minimized? <input type="checkbox"/> Are standards used and adhered to? <input type="checkbox"/> Are used technologies/libraries stable? <input type="checkbox"/> Do all involved people have the necessary knowledge? 		<div></div> notes



Thank you for using the Design Matrix. We hope it will help you getting better and well-balanced solutions.

Goals

The Design Matrix is an instrument to...

- ...reflect design decisions from different perspectives
- ...avoid neglecting relevant points.

Overview

The Design Types System describes four dimensions of design:

- simple vs. powerful** (green)
- abstract vs. concrete** (blue)
- pragmatic vs. idealistic** (red)
- robust vs. technologic** (yellow)

Each dimension represents two contrary but related perspectives that need to be considered when making a design decision. The Design Matrix provides a set of starting questions that help you to consider a design decision from all these perspectives. An overview:

simple stands for simple solutions, no magic, nothing sophisticated just easy to read/maintain (symbol: spear).
powerful stands for foresighted solutions, generic and flexible (symbol: halberd).
abstract stands for having the big picture in mind and keeping the bird's eye view (symbol: eagle).
concrete stands for knowing the details, being able to breathe code like a fish can breathe water (symbol: fish).
pragmatic stands for creating value with a very customer focused perspective (symbol: money bag).
idealistic stands for focusing on quality and professionalism, for avoiding dirty hacks and 80 percent solutions (symbol: cross).
robust stands for stability and reduction of risks (symbol: tower).
technologic stands for the potential new technology offers (symbol: cannon).

How to Use the Matrix

You typically use the Design Matrix for preparing a design decision. It helps you identify relevant questions to ask so you can check whether you've missed an important point.

Basic Concepts

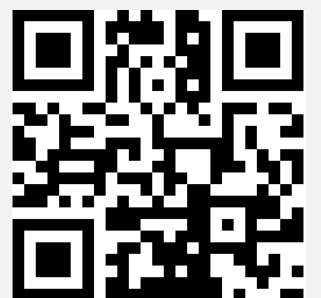
- For each dimension and each perspective examine your intended solutions using the questions provided by the Matrix.
- Often the questions lead to further questions that need to be considered.
- Use the notes boxes to write down answers, findings, or further questions.
- If you use the Matrix as a means to document your decision, use the header area to note down relevant meta data.

Advanced Usage

- The note boxes can be used to write down the arguments why and how this perspective is relevant and why not.
- The Matrix can also be used to compare several solutions for the same problem. Use a separate matrix for each solution and use the header area to note down which solution this matrix belongs to.
- You can use the matrix as documentation for important design decisions.
- You may establish team rules that certain important design decisions should be checked and/or documented using the Design Matrix.
- Not every aspect is equally important in a particular project. Mark important aspects in the matrix or even assign weights. In this way you can discuss and communicate the focus of further design decisions. This is especially helpful as a preparation for a discussion using the Design Cards.

Further Information

Like to get to know more?
Have a look at our free design type questionnaire and learn more about yourself, your colleagues and how to make discussions even more productive. Also try out our Design Cards. Have a look at **design-types.net**



The Authors



Matthias Wittum



Christian Rehn

Have fun with the Design Matrix!