

Inhaltsverzeichnis

1. Einführung und Ziele	2
1.1. Aufgabenstellung	2
2. Was ist "Produktname"?	2
3. Wesentliche Features	2
3.1. Qualitätsziele	2
3.2. Stakeholder	6
3.3. Technischer Kontext	11
3.4. Infrastruktur Ebene 1	15
3.5. Infrastruktur Ebene 2	15
4. Querschnittliche Konzepte	17
4.1. <Konzept 1>	18
4.2. <Konzept 2>	18
4.3. <Konzept n>	18
5. Architekturentscheidungen	19
6. Qualitätsanforderungen	20
6.1. Qualitätsbaum	20
6.2. Qualitätsszenarien	20
7. Risiken und technische Schulden	22

Über arc42

arc42, das Template zur Dokumentation von Software- und Systemarchitekturen.

Template Version 8.1 DE. (basiert auf AsciiDoc Version), Mai 2022

Created, maintained and © by Dr. Peter Hruschka, Dr. Gernot Starke and contributors. Siehe <https://arc42.org>.

HINWEIS

Diese Version des Templates enthält Hilfen und Erläuterungen. Sie dient der Einarbeitung in arc42 sowie dem Verständnis der Konzepte. Für die Dokumentation eigener System verwenden Sie besser die *plain* Version.

1. Einführung und Ziele

In diesem Kapitel wird auf die Aufgabenstellung, die funktionalen Anforderung, dies das Produkt beschreibt sowie auch die Qualitätsziele eingegangen. Ausserdem werden die Stakeholder und ihre Erwartungshaltung definiert. Dies soll einen Überblick über die Rahmenbedingungen des Systems bieten.

1.1. Aufgabenstellung

In der Aufgabenstellung wird dies näher angeschaut und das Produkt erläutert.

2. Was ist "Produktname"?

Wie genau das Stromnetz aufgebaut wissen wenige, mit der steigenden "Awareness" bezüglichem der effizienten Stromnutzung und auch der optimalen Verteilung des Stroms innerhalb des Netzes an die jeweiligen Endnutzer. bzw. zur Abstraktion in unserem Fall Häuser.

Unser Produkt, mit dem Namen "Produktname" soll diese optimale Verteilung, basierend auf dem "Max-flow-Algorithmus" anschaulich dargestellt werden.

- Thematisiert die Komplexität der Verteilung von Strom im Stromnetz
- Interaktiver Umgang mit den einzelnen "Edges" und den "Kapazitäten"
- Die optimale Verteilung wird als anstrebenendes Ergebniss dem Nutzer übergeben. Der Spieler versucht also durch überlegen, selber auf die beste Verteilungs-möglichkeit zu kommen.

3. Wesentliche Features

Die wesentlichen Features sind:

- Strategisches Spiel, bzw. "Quiz".
- Wiederspielbarkeit bzw. "replayability", da Anpassung auf den eigenen Wissenstand möglich, da Levelauswahl vorhanden.
- Wissensübertrag, Verständniss bezüglich Stromnetz wird erhöht.

3.1. Qualitätsziele

Die Qualitätsziele umfassen die wichtigsten Qualitätsanforderung an das Spiel mit dem Namen "Produktname". Sie sind nach Priorität in der folgenden Tabelle geordnet.

Tabelle 1. Qualitätsziele

Qualitätsziel	Motivation und Beschreibung
Benutzbarkeit:	Der Aufwand der Nutzung für die User sollte minimal sein.

Zuverlässigkeit:	Unser System soll die zugewiesene Funktion in einem, von uns noch zu definierenden Zeitfenster erledigen.
Robustheit:	Da mit greifbaren Teilen, Knöpfen und Spielfiguren gespielt werden kann, sollen diese Robust und gleichzeitig "wartbar" sein.
Effizienz:	Da unser Spiel live demonstriert wird, ist es wichtig, dass die Berechnungen schnell erledigt werden..

Auszug aus den Produktzielen unserer Confluence [Seite](#)

Tabelle 2. Produktziele

ID	Name	Beschreibung	Messung	Soll-Wert	
	<i>Kurzer Name des Ziels</i>	<i>Beschreibung des Zieles im Detail, inkl. in welcher Situation bzw. durch wen das Ziel erreicht werden soll.</i>	<i>Wie in einer Demo/Nutzerte st die Zielerreichung überprüft werden soll</i>	<i>Schwellwert, um auszusagen "Ziel ist erreicht."</i>	

Z01	Bewusstsein & Energienutzung	<p>Die Besucher:innen der Ausstellung der Primeo Energie AG in Münchenstein sollen ein Bewusstsein für die Nutzung von Strom entwickeln. Ziel ist es, dass Strom nicht als Selbstverständlichkeit angeschaut wird und man dadurch sparsamer und bewusster mit Energie/Strom umgeht. Im Allgemeinen betrifft dies nicht nur Primeo Energie AG, sondern jeden Menschen.¹⁴ Jährige haben ein Bewusstsein für eine nachhaltige Energienutzung entwickelt oder erweitert.</p>	<p>Am Schluss eine Frage auf Touch Display stellen, die mit einer Skala von 1-5 bewertet werden kann. Ein Fragebogen auf Papier funktioniert auch</p>	<p>Der Durchschnitt des Fragebogens soll bei 3.5 von 5 Sternen sein.</p>	
-----	------------------------------	---	---	--	--

Z02	Unterhaltung	<p>Wer das Produkt nutzt, soll während des Spielens unterhalten werden und Spass daran haben, das gestellte Problem spielerisch zu lösen. So haben die Besucher:innen der Ausstellung der Primeo Energie AG in Münchenstein eine positive Tätigkeit erlebt - der Besuch der Ausstellung wird somit emotional positiv gewertet.Habe ich Spass? Qualität Vergleich mit Alternativen. Subjektive Wahrnehmung des Spasses/Unterhaltung</p>	<p>Es wird ein Counter eingeführt, der misst, wie viele Levels gespielt wurden.dann dito wie oben</p>	<p>Die Spieler spielen mehr als 1 Level bei unserem Produkt.</p>	
-----	--------------	--	---	--	--

Z03	Max Flow Problem	Das mathematische Problem (Max Flow Problem) soll den Spielern näher gebracht werden. Benötigt ein Ort im Stromnetz mehr Energie, muss ein optimaler Weg für die Energie vom Herstellungsort bis zur Senke berechnet werden. Dabei hilft das Max Flow Problem.	Die Spieler scannen den QR-Code um auf eine Website mit Informationen über das Max Flow Problem zu gelangen	10% der Spieler besuchen die Website	
-----	------------------	--	---	--------------------------------------	--

3.2. Stakeholder

Tabelle 3. Stakeholders des Arc42 Document

Wer?	Interesse, Bezug
Softwarearchitektinnen und -architekten:	Erfahrungen über das Unterhalten und erstellen eines Arc42 Dokuments sammeln.
Entwicklerinnen und Entwickler:	Dokumentieren der Entwicklung eines Projekts
Kunde:	Unterstützung der Kreation eines spannenden Max-flow-basierten Spiel

Tabelle 4. Stakeholders Projekt

Stakeholder	Ansprechpersonen	Charakterisierung und Hintergrund	Bedürfnisse
Art des Stakeholders	Name, e-Mail...	Eigenschaften, Fähigkeiten, Geräte, Wissen, Erfahrungen, etc., die in die Nutzung des Produkts gebracht werden. Evtl. Verweise auf Dokumente/URL.	Mögliche Ziele, Anforderungen, Einschränkungen, die das Produkt attraktiv machen.

Stakeholder	Ansprechpersonen	Charakterisierung und Hintergrund	Bedürfnisse
Kunde	Andreas Vogt, andreas.vogt@fhnw.ch	Vertritt Primeo Energie AG, war in den letzten 3 Jahren bereits Kunde bei der FHNW und hat Board Games und Arcade Games in Auftrag gegeben.	Spielerischer Faktor des Spiels muss gewährleistet sein, erneuerbare Energien müssen eine Rolle spielen. Muss das Max Flow Problem beinhalten
Auftraggeber (FHNW)	Barbara Scheuner, barbara.scheuner@fhnw.ch	nimmt jährlich Aufträge an, welche realisierbar für das Projekt IP12 sind. Eine Jury wird das Produkt am Ende bewerten.	Erfüllung von Deadlines und Abgabeterminen.
Qualitätsmanager	Joshua Brehm, joshua.brehm@student.s.fhnw.ch	Verantwortlich für die Entwicklung und Einhaltung der Qualitätsstandards. Hat bereits in der IT-Branche gearbeitet.	klarer Informationsfluss, Mitarbeit von Teammitgliedern
Projektmanager	Shane Zulauf, shane.zulauf@students.fhnw.ch	Verantwortlich für das initiieren, Planen, Steuern, Kontrollieren und Abschliessen des Projekts. Hat bereits ein Projekt geleitet.	Mitarbeit von Teammitgliedern, Einhaltung von Terminen

Features

Tabelle 5. Features

ID	Name	Wichtigkeit	Aufwand	Kurzbeschreibung	Ziele	Implementierung
F101	RFID Daten auslesen	high	high	Das System sollte die vom User gelegten Bausteine auslesen können.	Spieler können ihren Spielzug auf dem Spielfeld realisieren	RFID reading in Java

ID	Name	Wichtigkeit	Aufwand	Kurzbeschreibung	Ziele	Implementierung
F101.1	Daten aus mehreren RFID Readern gleichzeitig auslesen	high	high	Das System sollte nach jetzigen Plan mindestens 20 individuelle ansteuerbare RFID Reader besitzen	Spieler können ihren Spielzug auf dem Spielfeld realisieren	RFID reading in Java
F102	Kontrollleuchte neben den Gebäuden	high	low	Knotenpunkte besitzen eine Kontrollleuchte, damit der Spieler erkennen kann, ob seine Lösung erlaubt ist. Auch kann mit der Kontrollleuchte der Startpunkt und Zwischen-/Endpunkt angezeigt werden.	System hat die Möglichkeit dem User seinen Fehler darzustellen.	LED auf dem Spielfeld sind alle einzeln Ansteuerbar.

ID	Name	Wichtigkeit	Aufwand	Kurzbeschr eibung	Ziele	Implementi erung
F103	Levelauswahl	high	high	Auf einem Touch Bildschirm werden Buttons für verschiedene Levels dargestellt. Mit drücken auf den Bildschirm auf einen Button wird ein Level ausgewählt. Das System weiss somit welches vordefinierte Level dem User dargestellt werden soll.	System gibt dem User die Möglichkeit das Spiel, an seinen gewollten Schwierigkeitsgrad anzupassen.	Touch-Bildschirm
F04	Physische Spielanleitung	medium	low	Auf dem Holz des Board Games ist die Spielanleitung zum Spiel geschrieben.	Der Spieler muss nicht den Bildschirm bedienen um die Spielanleitung zu lesen und kann während des Spielens jederzeit die Spielanleitung lesen.	Laserdruck auf Holz, Klebstoff auf Holz

ID	Name	Wichtigkeit	Aufwand	Kurzbeschreibung	Ziele	Implementierung
F05	Touch Screen	medium	high	Mithilfe eines Touch Screens kann die Levelauswahl betätigt werden, der Highscore kann angezeigt werden und bei einem Fehler des Spielers kann dies auch angezeigt werden.	Der Spieler tippt auf den Bildschirm, um weitere Level spielen zu können	Verkabelung mit RPi und Programmierung auf RPi, dann Einsetzung in Holzrahmen

Inhalt

Randbedingungen und Vorgaben, die ihre Freiheiten bezüglich Entwurf, Implementierung oder Ihres Entwicklungsprozesses einschränken. Diese Randbedingungen gelten manchmal organisations- oder firmenweit über die Grenzen einzelner Systeme hinweg.

Motivation

Für eine tragfähige Architektur sollten Sie genau wissen, wo Ihre Freiheitsgrade bezüglich der Entwurfsentscheidungen liegen und wo Sie Randbedingungen beachten müssen. Sie können Randbedingungen vielleicht noch verhandeln, zunächst sind sie aber da.

Form

Einfache Tabellen der Randbedingungen mit Erläuterungen. Bei Bedarf unterscheiden Sie technische, organisatorische und politische Randbedingungen oder übergreifende Konventionen (beispielsweise Programmier- oder Versionierungsrichtlinien, Dokumentations- oder Namenskonvention).

Weiterführende Informationen

Siehe [Randbedingungen](#) in der online-Dokumentation (auf Englisch!).

Inhalt

Die Kontextabgrenzung grenzt das System gegen alle Kommunikationspartner (Nachbarsysteme und Benutzerrollen) ab. Sie legt damit die externen Schnittstellen fest und zeigt damit auch die Verantwortlichkeit (scope) Ihres Systems: Welche Verantwortung trägt das System und welche Verantwortung übernehmen die Nachbarsysteme?

Differenzieren Sie fachlichen (Ein- und Ausgaben) und technischen Kontext (Kanäle, Protokolle,

Hardware), falls nötig.

Motivation

Die fachlichen und technischen Schnittstellen zur Kommunikation gehören zu den kritischsten Aspekten eines Systems. Stellen Sie sicher, dass Sie diese komplett verstanden haben.

Form

Verschiedene Optionen:

- Diverse Kontextdiagramme
- Listen von Kommunikationsbeziehungen mit deren Schnittstellen

Weiterführende Informationen

Siehe [Kontextabgrenzung](#) in der online-Dokumentation (auf Englisch!).

Inhalt

Festlegung **aller** Kommunikationsbeziehungen (Nutzer, IT-Systeme, ...) mit Erklärung der fachlichen Ein- und Ausgabedaten oder Schnittstellen. Zusätzlich (bei Bedarf) fachliche Datenformate oder Protokolle der Kommunikation mit den Nachbarsystemen.

Motivation

Alle Beteiligten müssen verstehen, welche fachlichen Informationen mit der Umwelt ausgetauscht werden.

Form

Alle Diagrammarten, die das System als Blackbox darstellen und die fachlichen Schnittstellen zu den Nachbarsystemen beschreiben.

Alternativ oder ergänzend können Sie eine Tabelle verwenden. Der Titel gibt den Namen Ihres Systems wieder; die drei Spalten sind: Kommunikationsbeziehung, Eingabe, Ausgabe.

<Diagramm und/oder Tabelle>

<optional: Erläuterung der externen fachlichen Schnittstellen>

3.3. Technischer Kontext

Die technischen Schnittstellen des Projekts und der umliegenden Technik wird in diesem Teilbereich genauer untersucht. Hier werden die technischen Schnittstellen auf die fachlichen spezifikationen "gemappt".Deploymentdiagramm technischer Kontext

Inhalt

Technische Schnittstellen (Kanäle, Übertragungsmedien) zwischen dem System und seiner Umwelt. Zusätzlich eine Erklärung (*mapping*), welche fachlichen Ein- und Ausgaben über welche technischen Kanäle fließen.

Motivation

Viele Stakeholder treffen Architekturentscheidungen auf Basis der technischen Schnittstellen des Systems zu seinem Kontext.

Insbesondere bei der Entwicklung von Infrastruktur oder Hardware sind diese technischen Schnittstellen durchaus entscheidend.

Form

Beispielsweise UML Deployment-Diagramme mit den Kanälen zu Nachbarsystemen, begleitet von einer Tabelle, die Kanäle auf Ein-/Ausgaben abbildet.

<Diagramm oder Tabelle>

<optional: Erläuterung der externen technischen Schnittstellen>

<Mapping fachliche auf technische Schnittstellen>

Inhalt

Kurzer Überblick über die grundlegenden Entscheidungen und Lösungsansätze, die Entwurf und Implementierung des Systems prägen. Hierzu gehören:

- Technologieentscheidungen
- Entscheidungen über die Top-Level-Zerlegung des Systems, beispielsweise die Verwendung gesamthaft prägender Entwurfs- oder Architekturmuster,
- Entscheidungen zur Erreichung der wichtigsten Qualitätsanforderungen sowie
- relevante organisatorische Entscheidungen, beispielsweise für bestimmte Entwicklungsprozesse oder Delegation bestimmter Aufgaben an andere Stakeholder.

Motivation

Diese wichtigen Entscheidungen bilden wesentliche „Eckpfeiler“ der Architektur. Von ihnen hängen viele weitere Entscheidungen oder Implementierungsregeln ab.

Form

Fassen Sie die zentralen Entwurfsentscheidungen **kurz** zusammen. Motivieren Sie, ausgehend von Aufgabenstellung, Qualitätszielen und Randbedingungen, was Sie entschieden haben und warum Sie so entschieden haben. Vermeiden Sie redundante Beschreibungen und verweisen Sie eher auf weitere Ausführungen in Folgeabschnitten.

Weiterführende Informationen

Siehe [Lösungsstrategie](#) in der online-Dokumentation (auf Englisch!).

Inhalt

Die Bausteinsicht zeigt die statische Zerlegung des Systems in Bausteine (Module, Komponenten, Subsysteme, Klassen, Schnittstellen, Pakete, Bibliotheken, Frameworks, Schichten, Partitionen, Tiers, Funktionen, Makros, Operationen, Datenstrukturen, ...) sowie deren Abhängigkeiten

(Beziehungen, Assoziationen, ...)

Diese Sicht sollte in jeder Architekturdokumentation vorhanden sein. In der Analogie zum Hausbau bildet die Bausteinsicht den *Grundrissplan*.

Motivation

Behalten Sie den Überblick über den Quellcode, indem Sie die statische Struktur des Systems durch Abstraktion verständlich machen.

Damit ermöglichen Sie Kommunikation auf abstrakterer Ebene, ohne zu viele Implementierungsdetails offenlegen zu müssen.

Form

Die Bausteinsicht ist eine hierarchische Sammlung von Blackboxen und Whiteboxen (siehe Abbildung unten) und deren Beschreibungen.

Inhalt

Diese Sicht erklärt konkrete Abläufe und Beziehungen zwischen Bausteinen in Form von Szenarien aus den folgenden Bereichen:

- Wichtige Abläufe oder *Features*: Wie führen die Bausteine der Architektur die wichtigsten Abläufe durch?
- Interaktionen an kritischen externen Schnittstellen: Wie arbeiten Bausteine mit Nutzern und Nachbarsystemen zusammen?
- Betrieb und Administration: Inbetriebnahme, Start, Stop.
- Fehler- und Ausnahmeszenarien

Anmerkung: Das Kriterium für die Auswahl der möglichen Szenarien (d.h. Abläufe) des Systems ist deren Architekturelevanz. Es geht nicht darum, möglichst viele Abläufe darzustellen, sondern eine angemessene Auswahl zu dokumentieren.

Motivation

Sie sollten verstehen, wie (Instanzen von) Bausteine(n) Ihres Systems ihre jeweiligen Aufgaben erfüllen und zur Laufzeit miteinander kommunizieren.

Nutzen Sie diese Szenarien in der Dokumentation hauptsächlich für eine verständlichere Kommunikation mit denjenigen Stakeholdern, die die statischen Modelle (z.B. Bausteinsicht, Verteilungssicht) weniger verständlich finden.

Form

Für die Beschreibung von Szenarien gibt es zahlreiche Ausdrucksmöglichkeiten. Nutzen Sie beispielsweise:

- Nummerierte Schrittfolgen oder Aufzählungen in Umgangssprache
- Aktivitäts- oder Flussdiagramme
- Sequenzdiagramme
- BPMN (Geschäftsprozessmodell und -notation) oder EPKs (Ereignis-Prozessketten)

- Zustandsautomaten
- ...

Weiterführende Informationen

Siehe [Laufzeitsicht](#) in der online-Dokumentation (auf Englisch!).

=== <Bezeichnung Laufzeitszenario 1>

- <hier Laufzeitdiagramm oder Ablaufbeschreibung einfügen>
- <hier Besonderheiten bei dem Zusammenspiel der Bausteine in diesem Szenario erläutern>

=== <Bezeichnung Laufzeitszenario 2>

...

=== <Bezeichnung Laufzeitszenario n>

...

== Verteilungssicht

#TODO - Create Diagrams that describe the process.

3.4. Infrastruktur Ebene 1

An dieser Stelle beschreiben Sie (als Kombination von Diagrammen mit Tabellen oder Texten):

- die Verteilung des Gesamtsystems auf mehrere Standorte, Umgebungen, Rechner, Prozessoren o. Ä., sowie die physischen Verbindungskanäle zwischen diesen,
- wichtige Begründungen für diese Verteilungsstruktur,
- Qualitäts- und/oder Leistungsmerkmale dieser Infrastruktur,
- Zuordnung von Softwareartefakten zu Bestandteilen der Infrastruktur

Für mehrere Umgebungen oder alternative Deployments kopieren Sie diesen Teil von arc42 für alle wichtigen Umgebungen/Varianten.

<Übersichtsdiagramm>

Begründung

<Erläuternder Text>

Qualitäts- und/oder Leistungsmerkmale

<Erläuternder Text>

Zuordnung von Bausteinen zu Infrastruktur

<Beschreibung der Zuordnung>

3.5. Infrastruktur Ebene 2

An dieser Stelle können Sie den inneren Aufbau (einiger) Infrastrukturelemente aus Ebene 1 beschreiben.

Für jedes Infrastrukturelement kopieren Sie die Struktur aus Ebene 1.

3.5.1. <Infrastrukturelement 1>

<Diagramm + Erläuterungen>

3.5.2. <Infrastrukturelement 2>

<Diagramm + Erläuterungen>

...

3.5.3. <Infrastrukturelement *n*>

<Diagramm + Erläuterungen>

4. Querschnittliche Konzepte

Inhalt

Dieser Abschnitt beschreibt übergreifende, prinzipielle Regelungen und Lösungsansätze, die an mehreren Stellen (=querschnittlich) relevant sind.

Solche Konzepte betreffen oft mehrere Bausteine. Dazu können vielerlei Themen gehören, beispielsweise:

- Modelle, insbesondere fachliche Modelle
- Architektur- oder Entwurfsmuster
- Regeln für den konkreten Einsatz von Technologien
- prinzipielle — meist technische — Festlegungen übergreifender Art
- Implementierungsregeln

Motivation

Konzepte bilden die Grundlage für *konzeptionelle Integrität* (Konsistenz, Homogenität) der Architektur und damit eine wesentliche Grundlage für die innere Qualität Ihrer Systeme.

Manche dieser Themen lassen sich nur schwer als Baustein in der Architektur unterbringen (z.B. das Thema „Sicherheit“).

Form

Kann vielfältig sein:

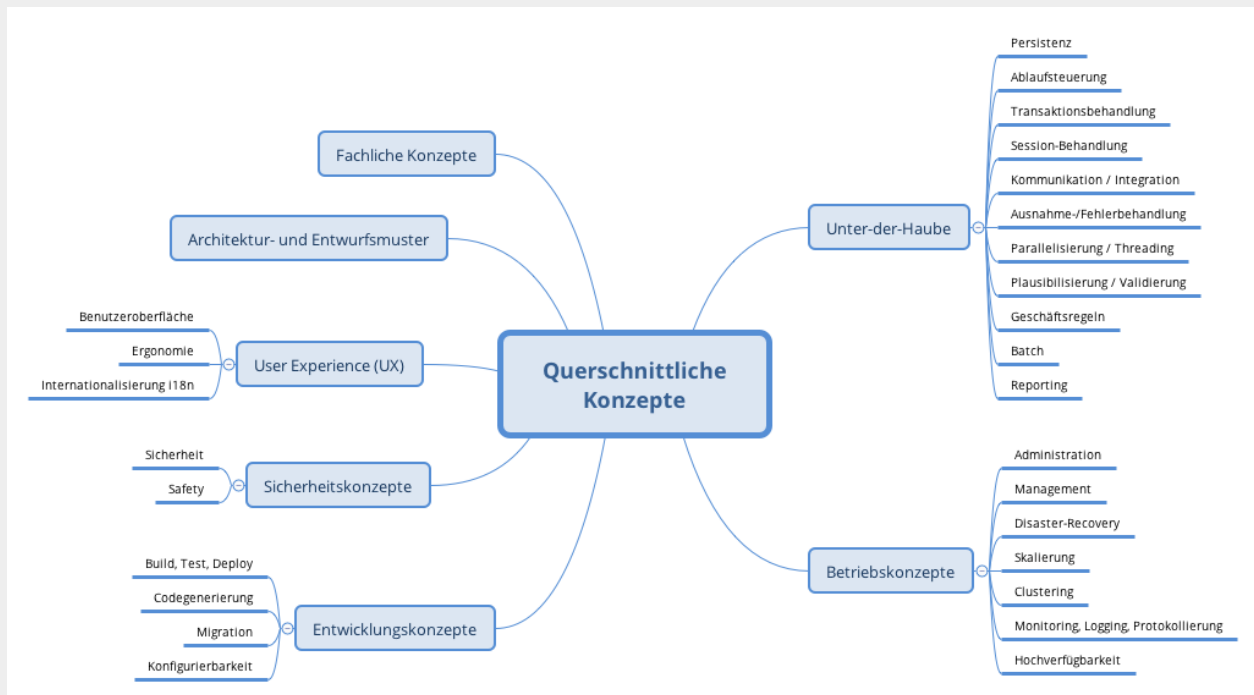
- Konzeptpapiere mit beliebiger Gliederung,
- übergreifende Modelle/Szenarien mit Notationen, die Sie auch in den Architektursichten nutzen,
- beispielhafte Implementierung speziell für technische Konzepte,
- Verweise auf „übliche“ Nutzung von Standard-Frameworks (beispielsweise die Nutzung von Hibernate als Object/Relational Mapper).

Struktur

Eine mögliche (nicht aber notwendige!) Untergliederung dieses Abschnittes könnte wie folgt aussehen (wobei die Zuordnung von Themen zu den Gruppen nicht immer eindeutig ist):

- Fachliche Konzepte
- User Experience (UX)
- Sicherheitskonzepte (Safety und Security)
- Architektur- und Entwurfsmuster
- Unter-der-Haube
- Entwicklungskonzepte

- Betriebskonzepte



Weiterführende Informationen

Siehe [Querschnittliche Konzepte](#) in der online-Dokumentation (auf Englisch).

4.1. <Konzept 1>

<Erklärung>

4.2. <Konzept 2>

<Erklärung>

...

4.3. <Konzept n>

<Erklärung>

5. Architekturentscheidungen

Inhalt

Wichtige, teure, große oder riskante Architektur- oder Entwurfsentscheidungen inklusive der jeweiligen Begründungen. Mit "Entscheidungen" meinen wir hier die Auswahl einer von mehreren Alternativen unter vorgegebenen Kriterien.

Wägen Sie ab, inwiefern Sie Entscheidungen hier zentral beschreiben, oder wo eine lokale Beschreibung (z.B. in der Whitebox-Sicht von Bausteinen) sinnvoller ist. Vermeiden Sie Redundanz. Verweisen Sie evtl. auf Abschnitt 4, wo schon grundlegende strategische Entscheidungen beschrieben wurden.

Motivation

Stakeholder des Systems sollten wichtige Entscheidungen verstehen und nachvollziehen können.

Form

Verschiedene Möglichkeiten:

- ADR ([Documenting Architecture Decisions](#)) für jede wichtige Entscheidung
- Liste oder Tabelle, nach Wichtigkeit und Tragweite der Entscheidungen geordnet
- ausführlicher in Form einzelner Unterkapitel je Entscheidung

Weiterführende Informationen

Siehe [Architekturentscheidungen](#) in der arc42 Dokumentation (auf Englisch!). Dort finden Sie Links und Beispiele zum Thema ADR.

6. Qualitätsanforderungen

Inhalt

Dieser Abschnitt enthält möglichst alle Qualitätsanforderungen als Qualitätsbaum mit Szenarien. Die wichtigsten davon haben Sie bereits in Abschnitt 1.2 (Qualitätsziele) hervorgehoben.

Nehmen Sie hier auch Qualitätsanforderungen geringerer Priorität auf, deren Nichteinhaltung oder -erreichung geringe Risiken birgt.

Motivation

Weil Qualitätsanforderungen die Architekturentscheidungen oft maßgeblich beeinflussen, sollten Sie die für Ihre Stakeholder relevanten Qualitätsanforderungen kennen, möglichst konkret und operationalisiert.

Weiterführende Informationen

Siehe [Qualitätsanforderungen](#) in der online-Dokumentation (auf Englisch!).

6.1. Qualitätsbaum

Inhalt

Der Qualitätsbaum (à la ATAM) mit Qualitätsszenarien an den Blättern.

Motivation

Die mit Prioritäten versehene Baumstruktur gibt Überblick über die — oftmals zahlreichen — Qualitätsanforderungen.

Form

- Baumartige Verfeinerung des Begriffes „Qualität“, mit „Qualität“ oder „Nützlichkeit“ als Wurzel.
- Mindmap mit Qualitätsoverbegriffen als Hauptzweige

In jedem Fall sollten Sie hier Verweise auf die Qualitätsszenarien des folgenden Abschnittes aufnehmen.

6.2. Qualitätsszenarien

Inhalt

Konkretisierung der (in der Praxis oftmals vagen oder impliziten) Qualitätsanforderungen durch (Qualitäts-)Szenarien.

Diese Szenarien beschreiben, was beim Eintreffen eines Stimulus auf ein System in

bestimmten Situationen geschieht.

Wesentlich sind zwei Arten von Szenarien:

- Nutzungsszenarien (auch bekannt als Anwendungs- oder Anwendungsfallszenarien) beschreiben, wie das System zur Laufzeit auf einen bestimmten Auslöser reagieren soll. Hierunter fallen auch Szenarien zur Beschreibung von Effizienz oder Performance. Beispiel: Das System beantwortet eine Benutzeranfrage innerhalb einer Sekunde.
- Änderungsszenarien beschreiben eine Modifikation des Systems oder seiner unmittelbaren Umgebung. Beispiel: Eine zusätzliche Funktionalität wird implementiert oder die Anforderung an ein Qualitätsmerkmal ändert sich.

Motivation

Szenarien operationalisieren Qualitätsanforderungen und machen deren Erfüllung mess- oder entscheidbar.

Insbesondere wenn Sie die Qualität Ihrer Architektur mit Methoden wie ATAM überprüfen wollen, bedürfen die in Abschnitt 1.2 genannten Qualitätsziele einer weiteren Präzisierung bis auf die Ebene von diskutierbaren und nachprüfbaren Szenarien.

Form

Entweder tabellarisch oder als Freitext.

7. Risiken und technische Schulden

ID	Risiken	Risiko "LEVEL"
I001	RFID interference, Da wir vor haben, mehrere RFID Reader zu nutzen, besteht das Risiko von grosser interference. Die Praktische anwendung haben wir noch nicht untersucht (nur small scale testing).	High, lösbar mit viel Testing
I002	Wissensaneignung in Bezug auf allgemeine Lösungen mit der Library Pi4j. Speziell in Bezug auf die RFID Chips und deren ansteuerung (bei grösserer Anzahl)	High, Zeitintensiv, Backup plan notwendig.
I002.1	Wissensaneignung im Bezug auf i2c, SPI, UART, ansteuerung von mehrfachen RFID Chips unter Verwendung der Pi4j library.	High, Zeitintensiv, Backup plan notwendig.
I002.2	Vordefinierter Backup plan: Einen, zentralen RFID Chip für Erkennung der "Capacity", bestätigung dieser jedoch mithilfe von Knöpfen an den diversen Stellen.	HIGH Technisch leichter umsetzbar, da nur ein RFID anzusteuern ist und nicht 20.
I003	Erste Erfahrung mit Implementation eines Maxflow problems, da implementierung und Verständniss kompliziert, hoher aufwand.	High, viel know-how noch nachzuholen
I004	Verbindung technischer Komponenten (Verkabelung, Löten), Touchscreen, LED's	High, da know-how noch zu erarbeiten ist.
I005	Absturz aufgrund von Memory-Leaks.	medium, Problem sollte bis zur Live-demonstration, nach genug Testing "low" sein.
I006	Bezüglich Qualitätsziele: Sinnvolle Generation des Schwierigkeitsgrads vom Maxflowproblem. Abschätzung der Stärke	medium

ID	Risiken	Risiko "LEVEL"
I007	Implementierung der Wetter API, basierend auf Qualitätszielen: Sollte Offline verfügbar sein. (Caching der Daten)	low
I008	Vordefinierter Backup plan: Einen, zentralen RFID Chip für Erkennung der "Capacity", bestätigung dieser jedoch mithilfe von Knöpfen an den diversen Stellen.	HIGH Technisch leichter umsetzbar, da nur ein RFID anzusteuern ist und nicht 20.
I009	-	-
I010	-	-
I011	-	-