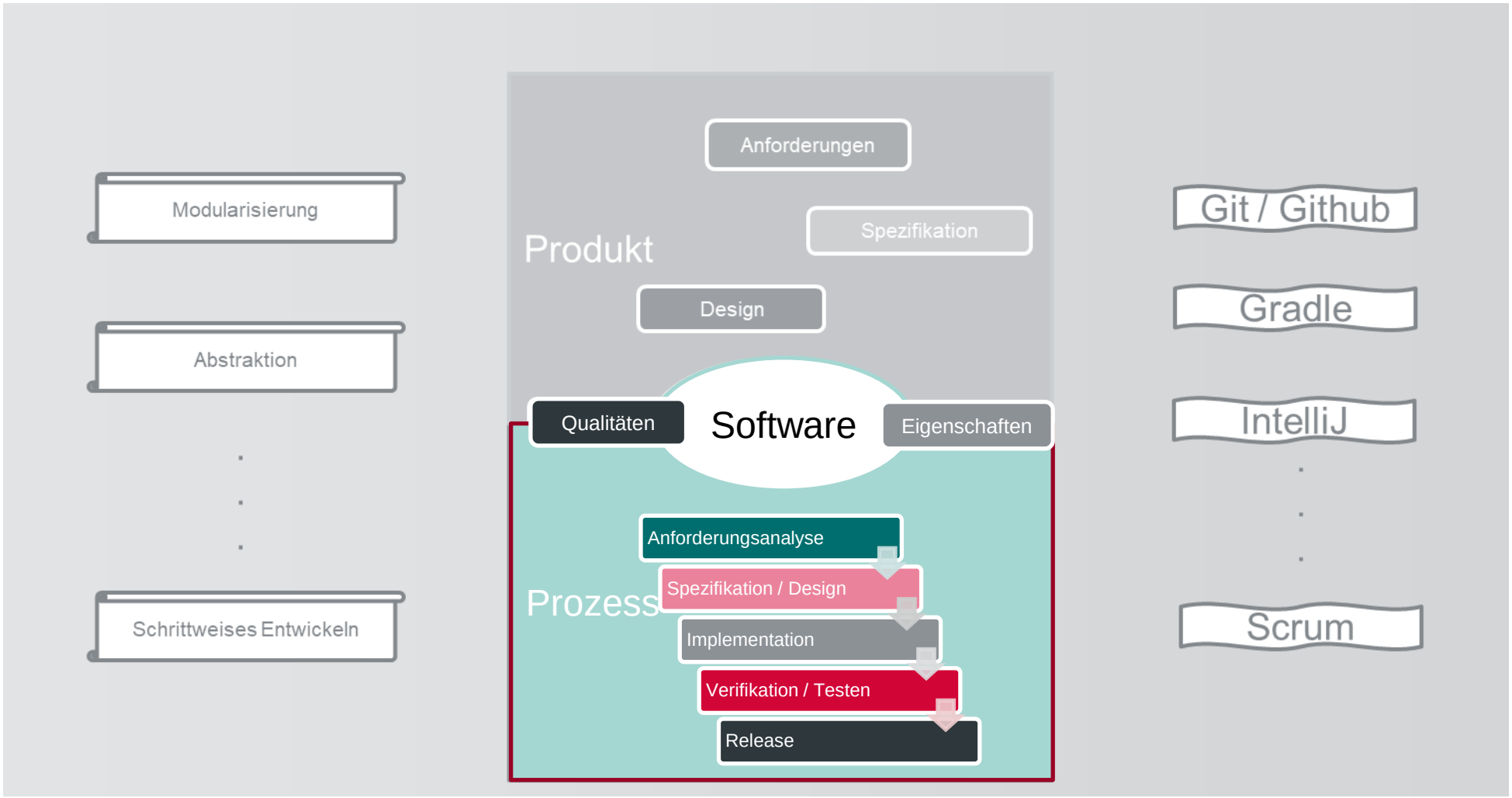
University of Basel

Software Engineering – Anforderungsanalyse

M. Lüthi, Universität Basel, 13. August 2022

Übersicht



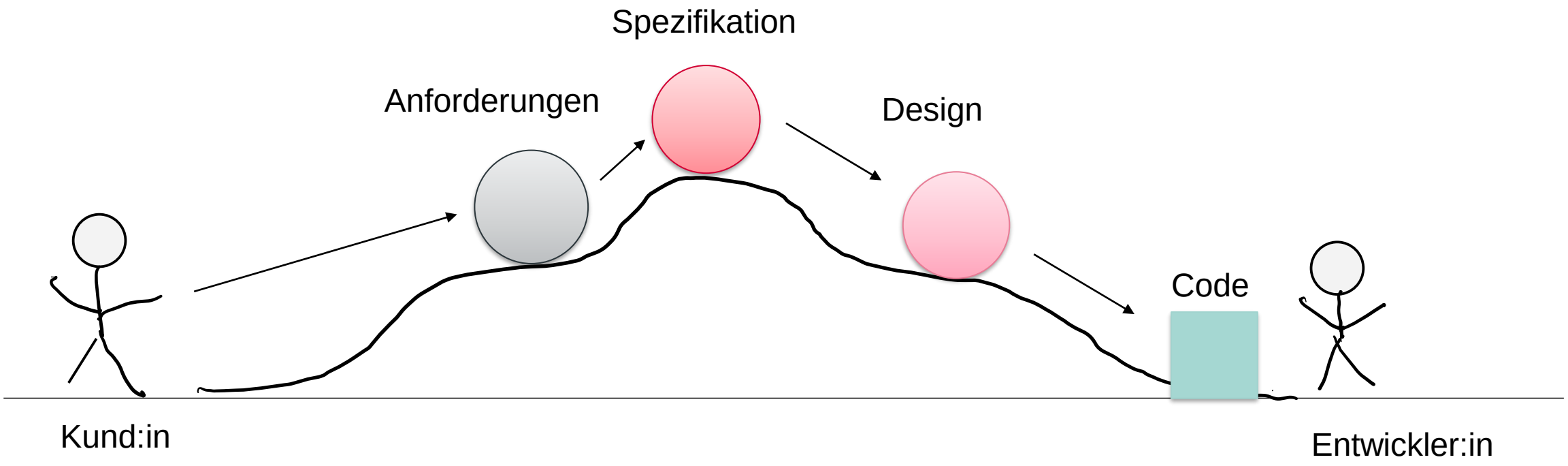
Prinzipien → Tools

Agenda 13. August 2022

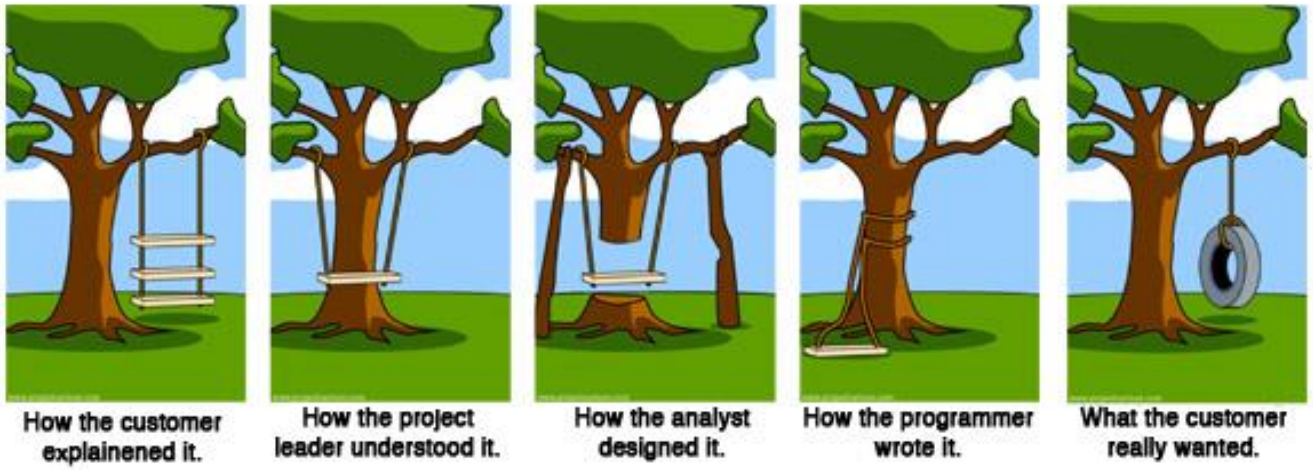
1	Anforderungsanalyse (30’)
2	Anforderungen spezifizieren (60’)
3	Praktische Übung: Pflichtenheft erstellen (135’)

Anforderungsanalyse

Anforderungen – vom Klienten zum Entwickler



Der Softwareentwicklungsprozess (karikiert)



Quelle: https://www.lorienpratt.com/decision-models-are-the-requirements-language-for-di-app

> The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements … No other part of the work so cripples the resulting system if done wrong. No other part is a s difficult to rectify later.
>
> F.P. Brooks (1987)

Aufgabe der Anforderungsanalyse

Anforderungen an zu entwickelndes System

- ermitteln
- spezifizieren
- analysieren
- validieren

und daraus Lösung ableiten.

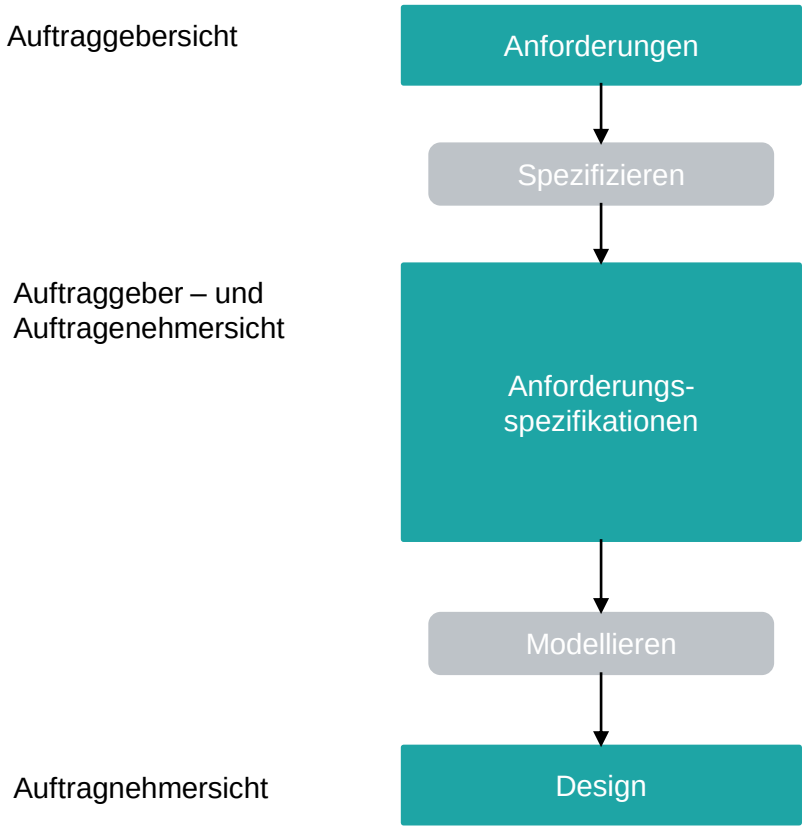
Ergebnis

Anforderungsspezifikation

Warum braucht es Spezifikationen

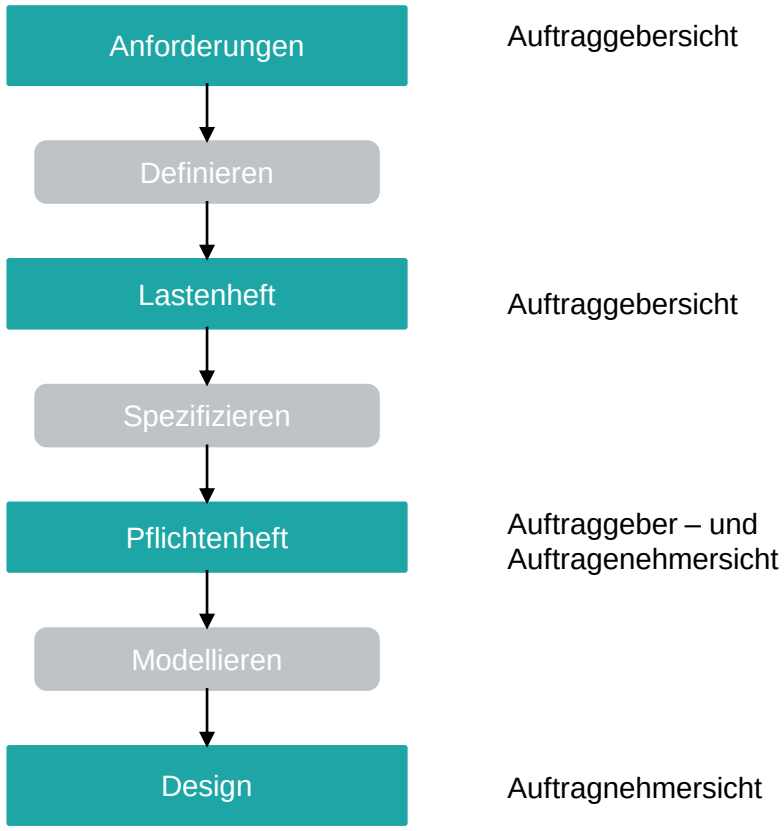
- Abstimmung mit Kunden
- Entwurf und Implementation
- Benutzungshandbuch
- Testvorbereitung
- Abnahme
- Wiederverwendung der Software
- Klärung späterer Einwände / Regressansprüche
- Wartung oder Re-Implementierung

Ablauf – Schematisch (I)



Ablauf – Schematisch (II)

Aufraggeber ≠ Auftragnehmer



Lastenheft

Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers innerhalb eines Auftrags aus Auftraggebersicht

- Notwendig wenn Auftraggeber das Projekt ausschreibt
- Dient als Vertragsgrundlage
- Kann lückenhaft, vage oder sogar inkonsistent sein

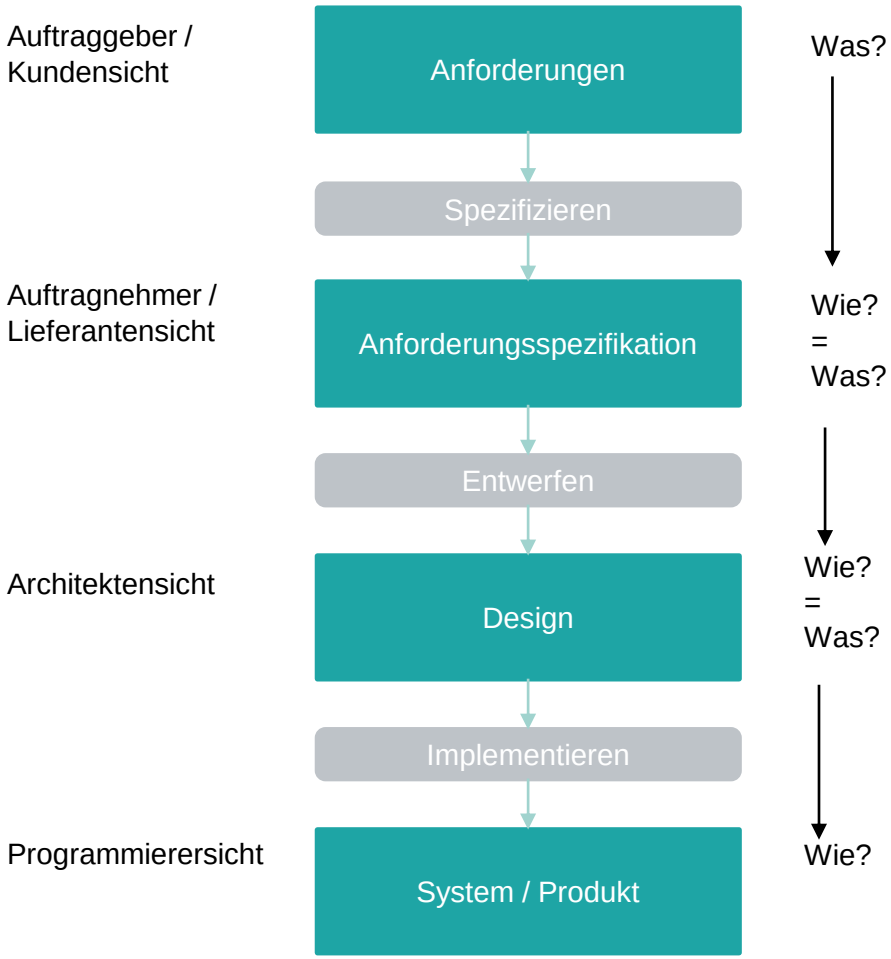
Pflichtenheft

Erarbeitete Realisierungsvorgaben aufgrund des Lastenheft

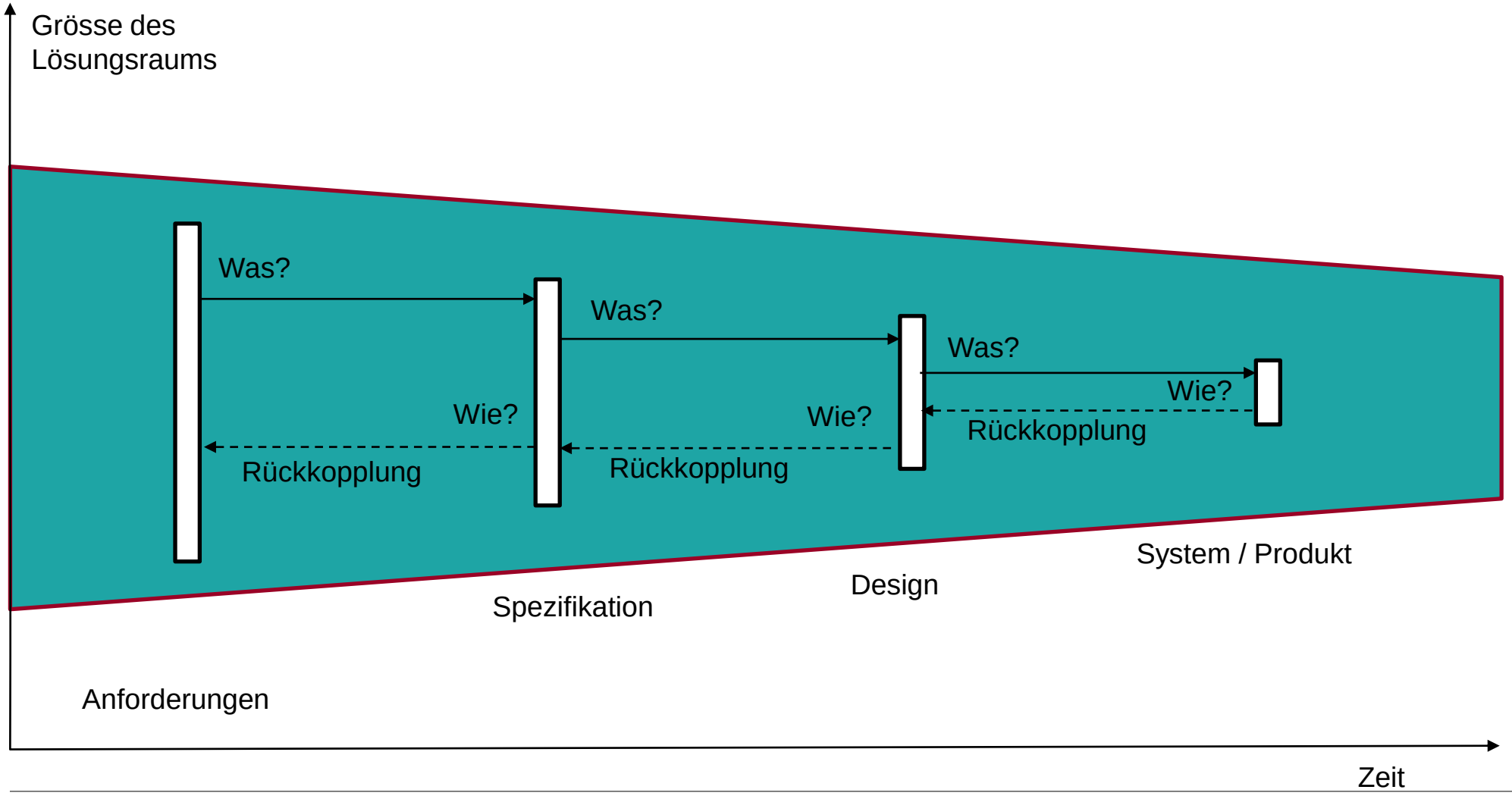
- Anwendervorgaben werden detailliert
- Realisierungsforderungen werden beschrieben.
- Definiert wie und womit die Anforderungen zu realisieren sind.
- Muss vollständig klar und konsistent sein

Ablauf: Problem vs. Lösung

Lösung in einem Schritt wird zum Problem für nächsten Schritt

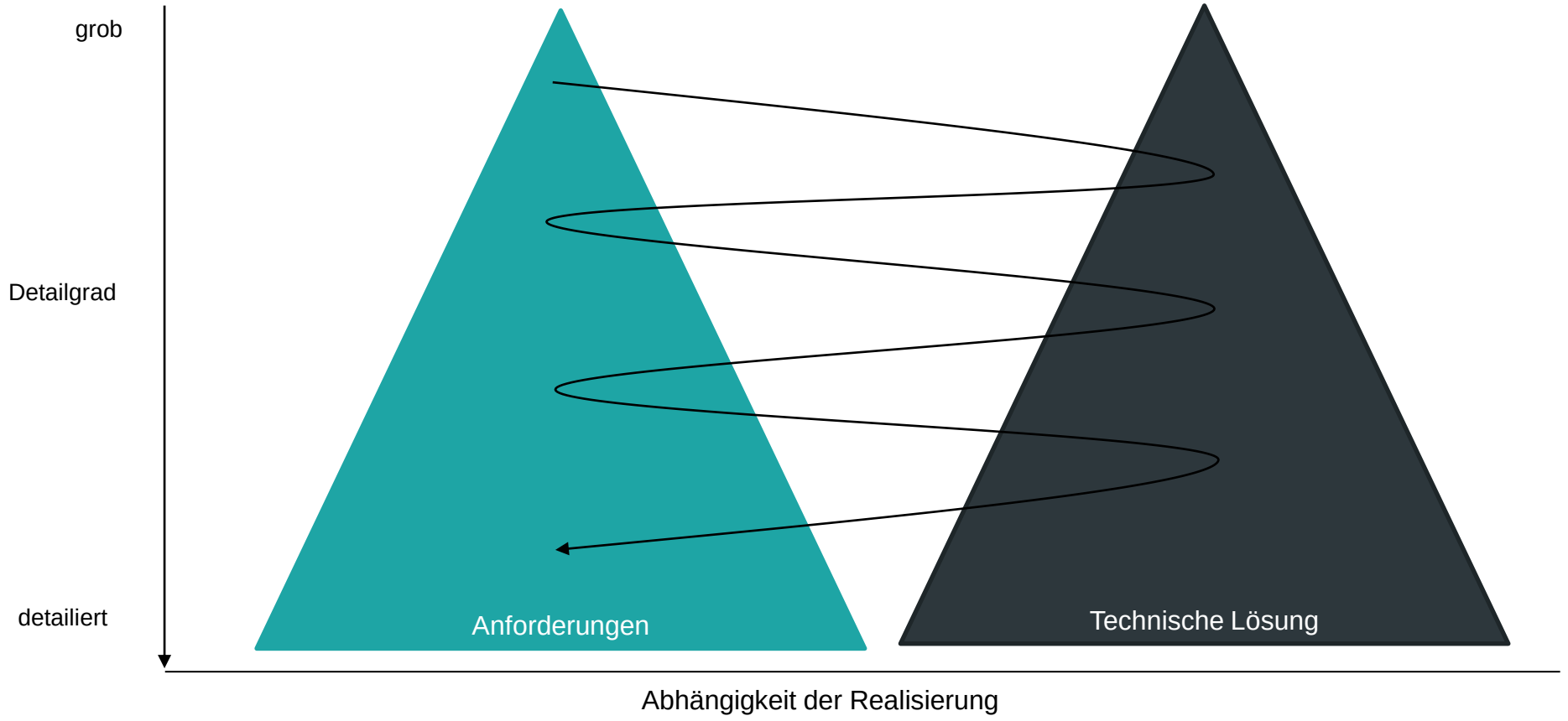


Problem / Lösungsraum



Wechselwirkung

Fehlender technische Lösung → Problem muss angepasst werden



Anforderungen

Anforderungen ermitteln: Beispieldialog

Sie öffnen also morgens die Tür am Haupteingang?

Ja, habe ich Ihnen doch gesagt,

Jeden Morgen?

Natürlich

Auch am Wochenende?

Nein, am Wochenende bleibt der Eingang zu

Und während der Betriebsferien?

Da bleibt er auch zu

Und wenn Sie krank sind oder Urlaub haben

Dann macht das Herr X

Und wenn auch Herr X ausfällt

Dann kopft irgendwann ein Kunde ans Fenster, weil er nicht reinkommt.

Was bedeutet “morgens”?

…

Dialog nach Lichter und Ludwig: Software Engineering

Anforderungen

Anforderungen legen fest, was man von einem Softwaresystem als Eigenschaften erwartet.

„man“ sind die Stakeholder:

- Parteien die
  - Ein Interesse an der Software haben
  - Von der Entwicklung/Einsatz der Software betroffen sind
- Verallgemeinert den traditionellen Kundenbegriff

Eigenschaften teilen sich auf in

- Funktionale Eigenschaften
- Nichtfunktionale Eigenschaften

Funktionale Eigenschaften

Legen eine vom System bereitzustellende Funktion fest.

Leitfrage: Was soll ein System tun?

Beispiele:

- Das System soll den PIN vom Kunden prüfen.
- Das System soll den Rechnungsbetrag erfragen, und falls genügend Geld vorhanden ist, das Konto belasten.

Nichtfunktionale Eigenschaften

Alle Anforderungen die nicht funktional sind.

Beispielkategorien:

- Qualitätsmerkmale (Performance, Wartbarkeit, ...)
- Sicherheitsanforderungen
- Ethische oder regulatorische Anforderungen

Betreffen meistens alle Funktionen des Systems

Anforderungen an Anforderungen

Inhalt

- Zutreffend
- Vollständig
- Widerspruchsfrei (oder konsistent)
- Neutral (oder abstrakt)
- Nachvollziehbar
- Objektivierbar

Form

- Leicht verständlich
- Präzise
- Leicht erstellbar
- Leicht verwaltbar

Darstellung der Spezifikationen

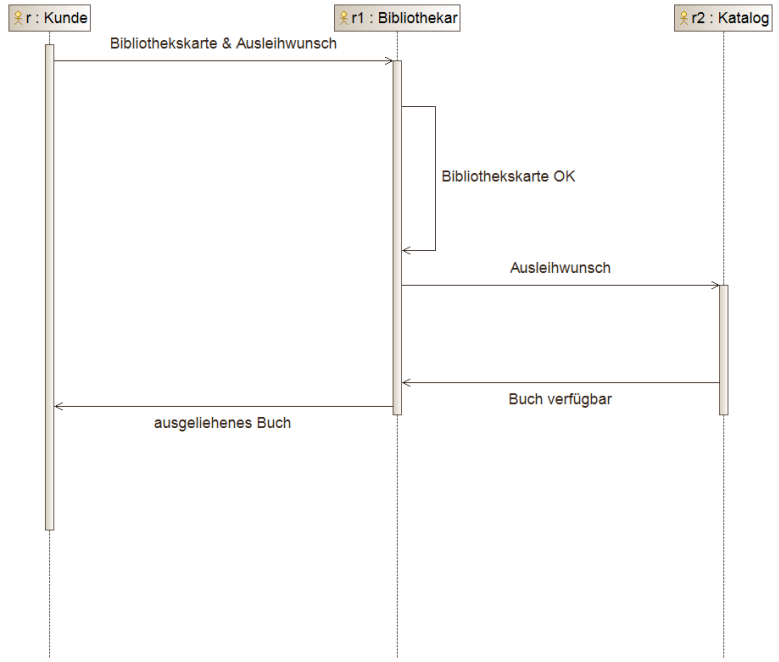
Formal

{
{$i1>0$ and $i2>0$}
P
(exists $z1,z2(i1=o\cdot z1$ and $i2=o\cdot z2)$
and not
(exists h(exists $z1,z2(i1=h\cdot z1$ and $i2=h\cdot z2)$ and $h>o$))
}

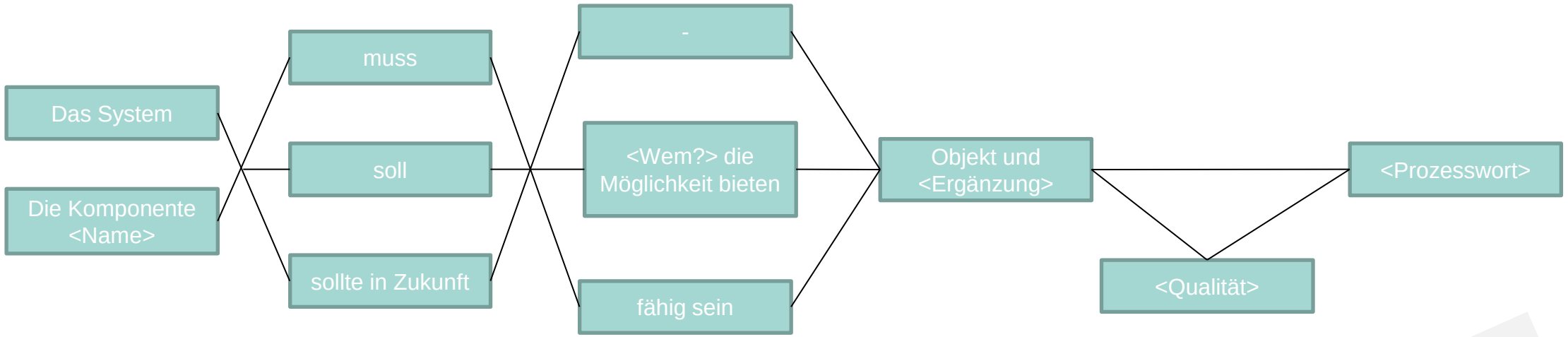
Natürlichsprachig

“Das System muss die Benutzerdaten alle 2 Minuten in eine Datenbank schreiben”

Grafisch



Natürlichsprachliche Anforderungen: Sprachschablonen



Muss: Pflicht
Soll: Wunsch
Sollte in Zukunft: Absicht für nächste Version

Strukturiert den Satzbau
Schränkt Vokabular ein

- Verben haben wenn möglich klare Bedeutung

Natürlichsprachliche Anforderungen : Glossar

Um Begrifflichkeiten zu klären wird ein Begriffslexikon (Glossar) geführt

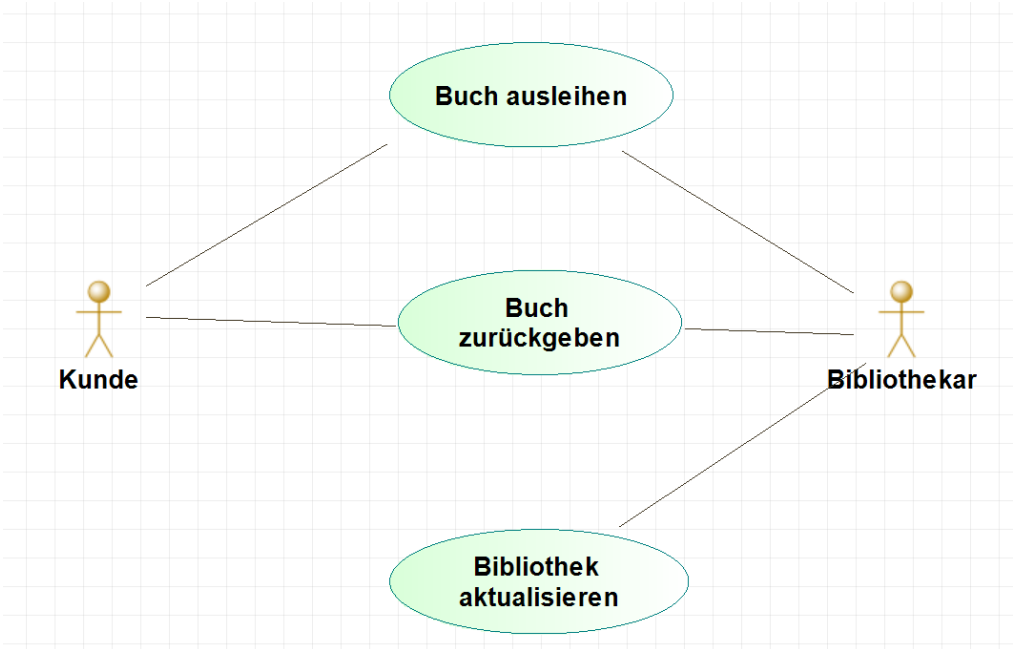
Begriff	Student, synonym Studentin, Studierende
Bedeutung	Eine Person, die an der Uni Basel immatrikuliert ist
Abgrenzung	Gasthörer und Studierende anderer Hochschulen sind im Sinne dieses Systems keine Studenten
Gültigkeit	Mit der Immatikulation entsteht ein neuer Student. Er existiert bis zur Exmatrikulation
Bezeichnung	Ein Student ist durch die Matrikelnummer une einen Zeitpunkt eindeutig bestimmt. Alle anderen Attribute können mehrmals vorkommen
Unklarheiten	Es ist noc ungeklärt, wie Namen aus anderen Schriftsystemen (Chinesisch, Arabisch) dargestellt warden
Querverweise	Matrikelnummer, Gasthörer

Beispiel aus Lichter & Ludwig: Software Engineering

Use cases

Use case: Textuelle Beschreibung einer Systeminteraktion eines Akteur um fachliches Ziel zu erreichen

- Einfache Technik um Anforderungen zu finden
  - Trick: Einnahme von Benutzersicht
- Standardisiert in UML



Use case

Was muss gemacht werden um Ziel des Nutzers zu verwirklichen?

Template

- Name: Name des use cases
- Akteure: Akteur 1, Akteur
- Vorbedingung:
- Nachbedingung (Erfolg):
- Nachbedingung (Fehler):

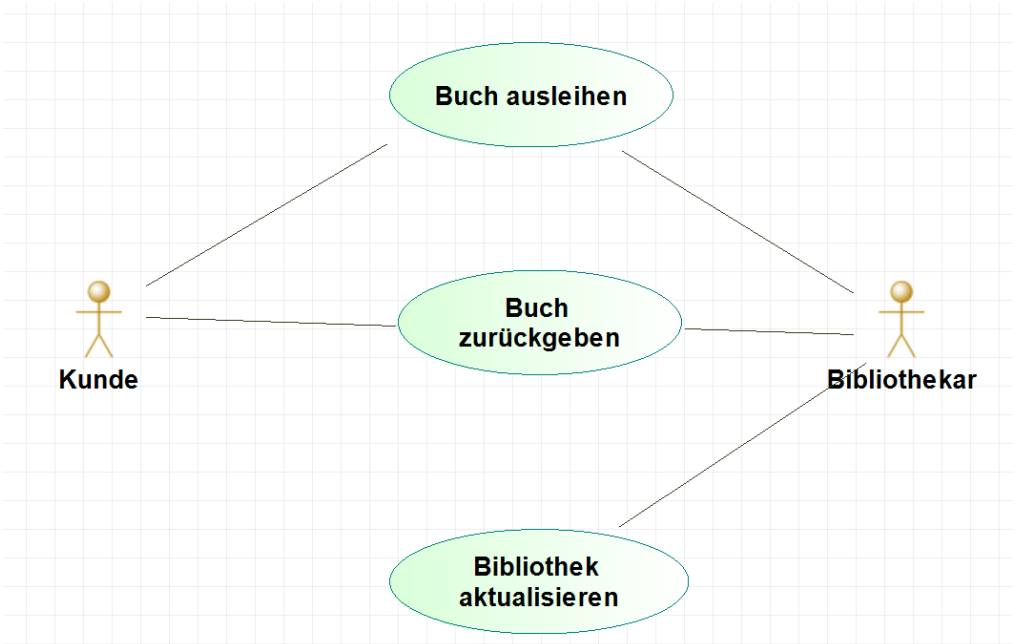
- Standardablauf:
  - Schritt 1
  - Schritt 2
  - …

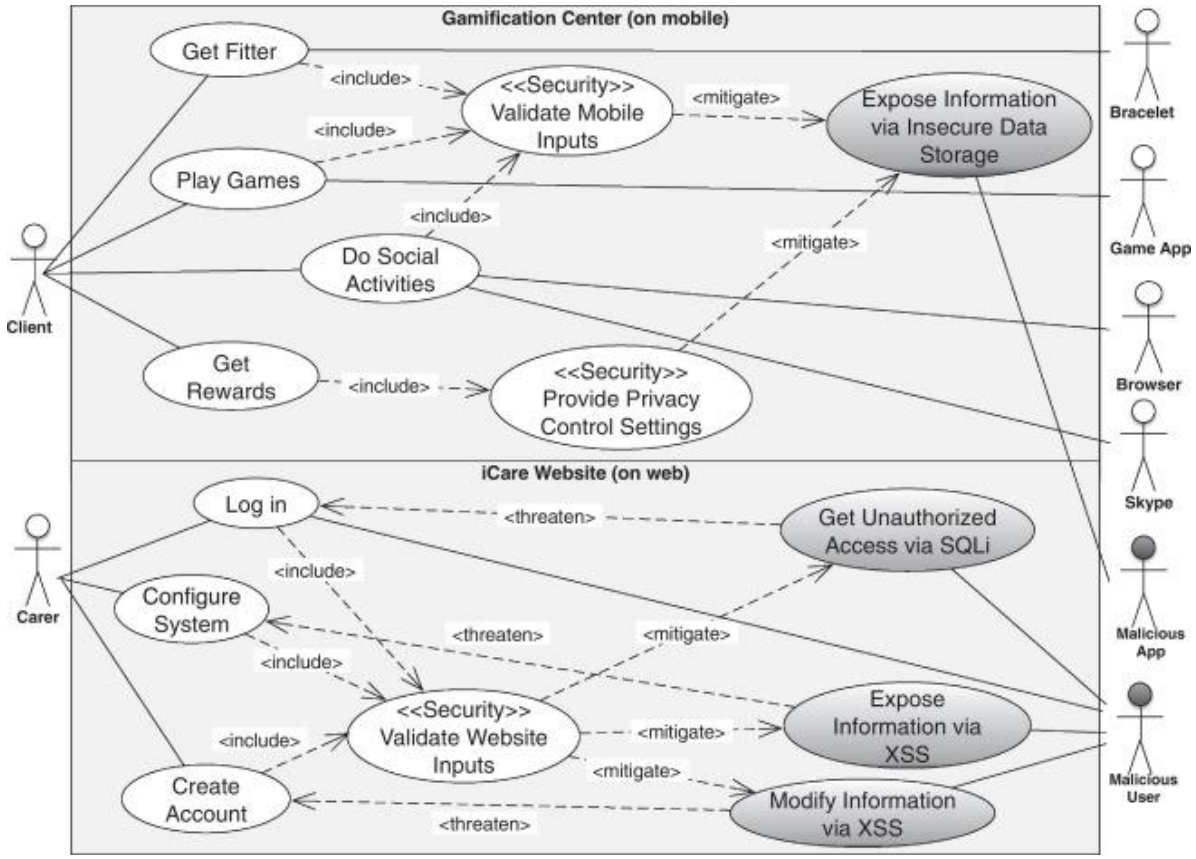
Use case

Beispiel

- Name: Buch ausleihen
- Akteure: Bibliothekar, Kunde
- Vorbedingung: Buch ist verfügbar
- Nachbedingung (Erfolg): Ausleihe im System vermerkt
- Nachbedingung (Fehler): -
- Standardablauf:
    1. Kunde gibt Bibliothekar seine Karte
    2. Bibliothekar prüft Karte und schaut sich getätige Vorbestellung an
    3. Bibliothekar holt alle bestellten Bücher
    4. Bibliothekar vermerkt im System das Bücher vom Kunden ausgeliehen sind.

Use case - Diagramme





- Grafische Übersicht über alle Use-cases.

Aber: Text, nicht Diagramme sind wichtig

Beispielpflichtenheft (aus Balzert, Lehrbuch der Softwaretechnik)

Veranstaltungsbetreuer: Betreut die →Teilnehmer und →Dozenten einer →Veranstaltung.

Pflichtenheft SemOrg

Version	Autor	Quelle	Status	Datum	Kommentar
0.1	Manfred Muster-mann	Geschäfts-führer Teachware	in Bear-beitung	11/09	Verfeine-rung des Lastenhefts V0.1

Voreinstellungen *(Kursiv dargestellt)*:
Priorität aus Auftraggebersicht = {hoch, *mittel*, niedrig}
Priorität aus Auftragnehmersicht = {hoch, *mittel*, niedrig}
Stabilität der Anforderung = {fest, *gefestigt*, volatil}
Kritikalität der Anforderung = {hoch, mittel, *niedrig*, keine}
Entwicklungsrisiko der Anforderung = {hoch, mittel, *niedrig*}

eis Alle Anforderungen im Pflichtenheft, die einen Bezug zu einer Anforderung im Lastenheft haben, müssen eine entsprechende Referenz auf das Lastenheft haben. Da das Pflichtenheft eine Verfeinerung und Erweiterung des Lastenhefts darstellt, gibt es neue Anforderungen, von denen kein Bezug zum Lastenheft hergestellt werden kann. Es gibt dann keine entsprechende Referenz.

1 Visionen und Ziele
/V10/ (/LV10/) Die Firma Teachware soll durch das System in die Lage versetzt werden, die von ihr veranstalteten Seminare sowie Kunden und Dozenten effizient rechnerunterstützt zu verwalten.
/V20/ (/LV20/) Die Kunden der Firma Teachware sollen über das Web möglichst viele Vorgänge selbst durchführen können.
/Z10/ (/LZ10/) Ein Interessent oder ein Kunde kann mindestens 20 Stunden jeden Tag Seminare und Veranstaltungen über das Web selektieren und eine Veranstaltung online buchen, damit die Mitarbeiter der Fa. Teachware von solchen Tätigkeiten entlastet werden.

2 Rahmenbedingungen
/R10/ (/LR10/) SemOrg ist eine kaufmännisch/administrative Web-Anwendung.
/R20/ (/LR20/) Zielgruppe sind die Mitarbeiter der Fa. Teachware (Kundensachbearbeiter, Seminarsachbearbeiter, Veranstaltungsbetreuer) sowie Interessenten und Kunden.
/R30/ Das System wird in einer Büroumgebung eingesetzt.
/R40/ (/LZ10/) Die tägliche Betriebszeit des Systems muss mindestens 20 Stunden jeden Tag betragen.

/R50/ Der Betrieb des Systems muss unbeaufsichtigt ablaufen.
/R60/ Eingesetzte Software auf der Zielmaschine: Client: Webbrowser (Die marktführenden 3 Webbrowser müssen unterstützt werden), Server: Betriebssystem Windows.
/R70/ Hardwarevoraussetzungen: Client: PC, Bildschirm mit mindestens XGA-Auflösung (1024 x 768), Server: TBD.
/R80/ (/LK10/) Netzwerkverbindung des Servers zum Buchhaltungsrechner.
/R90/ (/LK20/) Alle Clients sind über ein Intranet mit dem Server verbunden, der Server hat einen Internetanschluss.
/R100/ Die Entwicklungsumgebung kann identisch mit der Zielumgebung sein.

3 Kontext und Überblick
/K10/ (/LK10/) Das System besitzt eine Softwareschnittstelle zu einem Buchhaltungssystem: TBD.

4 Funktionale Anforderungen
/F10/ (/LF10/) Das System *soll* Interessenten und Kunden die Möglichkeit bieten, sich über Seminare und Veranstaltungen zu informieren, Veranstaltungen zu buchen und einen Seminarkatalog anzufordern.
/F11/ Wenn ein Kunde oder eine Firma sich von einer bereits gebuchten Veranstaltung mehr als X Wochen vor der Veranstaltung abmeldet, dann *muss* das System Stornogebühren in Höhe von Y Euro berechnen oder nach einem Ersatzteilnehmer fragen.
/F12/ Wenn ein Kunde oder eine Firma sich von einer bereits gebuchten Veranstaltung später als X Wochen vor der Veranstaltung abmeldet, dann *muss* das System Stornogebühren in Höhe der Veranstaltungsgebühr berechnen oder nach einem Ersatzteilnehmer fragen.
/F20/ (/LF20/) Das System *muss* dem Kundensachbearbeiter die Möglichkeit bieten, neue Kunden/Firmen zu erfassen und vorhandene Kunden-/Firmendaten zu aktualisieren und Kunden/Firmen zu löschen.
/F30/ (/LF30/) Das System *muss* dem Kundensachbearbeiter die Möglichkeit bieten, Seminare und Veranstaltungen zu selektieren, Veranstaltungen für Interessenten und Kunden zu buchen und zu stornieren sowie für angeforderte Seminarkataloge Versandpapiere zu erstellen.
/F40/ (/LF40/) Das System *muss* dem Seminarsachbearbeiter die Möglichkeit bieten, neue Dozenten zu erfassen, vorhandene Dozentendaten zu aktualisieren, Dozenten zu löschen und Dozenten Seminaren und Veranstaltungen zuzuordnen.
/F140/ Wenn ein Dozent eine Seminarveranstaltung leitet, dann *muss* das System dies speichern.
/F150/ Wenn ein Kunde oder eine Firma im Zahlungsverzug ist, dann *muss* das System folgende Daten dazu speichern: Datum der Rechnung, die noch nicht bezahlt ist, sowie Betrag der Rechnung.
/F160/ Minimal 3 Tage vor einer Veranstaltung *muss* das System dem Kundensachbearbeiter und dem betreffenden Dozenten die Möglichkeit bieten, eine Teilnehmerliste für die Veranstaltung mit folgenden Daten zu erstellen: Seminartitel, Datum von, Datum bis, Veranstaltungsort, Dozent(en) Pro Teilnehmer: Name, Vorname, Firma, Ort.
/F170/ Nach dem Ende einer Veranstaltung *muss* das System dem Kundensachbearbeiter und dem betreffenden Dozenten die Möglichkeit bieten, eine Teilnehmerurkunde für jeden Veranstaltungsteilnehmer mit folgenden Daten zu erstellen: Anrede, Titel, Vorname, Nachname, von Datum, bis Datum, Seminartitel, Veranstaltungsort, Inhaltsübersicht, Veranstaltungsleiter.
/F180/ Das System *muss* dem Dozenten und dem Veranstaltungsbetreuer die Möglichkeit bieten, für eine Veranstaltung Beurteilungsbögen auszudrucken.
/F190/ Wurde eine Veranstaltung durchgeführt, dann *muss* das System fähig sein, eine Honorarmitteilung an die Buchhaltung zu senden.

5 Qualitätsanforderungen
Die Qualitätsanforderungen sind in der Tab. 7.0-2 aufgeführt.
/QF10/ Beim Zugriff über das Internet muss das System eine sichere Übertragung (z. B. https) ermöglichen.
/QF20/ Das System muss die Rollen entsprechend der Tab. 7.0-3 unterscheiden und die dazugehörigen Zugriffsrechte sicherstellen können.
/QF30/ Wenn ein Benutzer SemOrg nutzen will, dann muss das System eine Autorisierung vom Benutzer verlangen.
/QB10/ Die Grundsätze der DIN EN ISO 9241–110 von 2006 mit dem Titel »Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung« sind einzuhalten.
/QE10/ (/LQE10/) Alle Reaktionszeiten auf Benutzeraktionen müssen unter 5 Sekunden liegen.

6 Abnahmekriterien
/A10/ Gültiges Abnahmeszenario: Ein Seminar neu erfassen, eine Veranstaltung neu erfassen, die Veranstaltung dem Seminar zuordnen, einen Dozenten erfassen und dem Seminar und der Veranstaltung zuordnen.
/A20/ TBD usw.

Pflichtenheft Template

Pflichtenheft

(Nach Lichter & Ludwig, Software Engineering: Grundlagen, Menschen, Prozesse, Techniken)

1. Einleitung

1.1 Zweck

Beschreibt den Zweck und den Leserkreis der Spezifikation.

1.2 Einsatzbereich und Ziele

Gibt an, wo die Software eingesetzt werden soll und welche wesentlichen Funktionen es haben wird. Wo sinnvoll, sollte auch definiert werden, was die Software nicht leisten wird.

Beschreibt die mit der Software verfolgten Ziele.

1.3 Definitionen

Dokumentiert alle verwendeten Fachbegriffe und Abkürzungen. Alternativ können Sie auch ein separates Glossar nutzen.

1.4 Referenzierte Dokumente

Verzeichnet alle Dokumente, auf die in der Spezifikation verwiesen wird.

1.5 Überblick

Beschreibt, wie der Rest der Spezifikation aufgebaut ist, insbesondere, wie Kapitel 3 strukturiert ist.

2. Allgemeine Beschreibung

2.1 Einbettung

Beschreibt, wie das System in seine Umgebung eingebettet ist und wie die Software mit den umgebenden Komponenten und Systemen zusammenspielt. Dazu werden die Schnittstellen, Kommunikationsprotokolle etc. definiert.

2.2 Funktionen

Skizziert die wichtigsten Funktionen

2.3 Benutzerprofile

Charakterisiert die Benutzergruppen und die Voraussetzungen die diese jeweils mitbringen (Ausbildung, Know-how, Sprache)

2.4 Einschränkungen

Dokumentiert Einschränkungen, die die Freiheit der Entwicklung reduzieren (Basis-Software, Ziel-Hardware, Gesetzliche Grundlagen, ...)

2.5 Annahmen und Abhängigkeiten

Nennt explizit die Annahmen und externen Voraussetzungen, von denen bei der Spezifikation ausgegangen wurde.

3. Einzelanforderungen

Beschreibt die Anforderung i so genau, dass bei der Verwendung der Spezifikation (im Entwurf usw.) keine Rückfragen dazu notwendig sind.

Identifizieren Sie jede Funktionale Anforderung mit einer Nummer, so dass diese Nachverfolgbar sind. Zusammengehörende Funktionale Anforderungen können durch geeignete Nummerierung angezeigt werden.

Zur Spezifikation der Software sollen Sprachschablonen benutzt werden.

- /F10/ Funktion 1 des Systems
- /F11/ Weitere Detaillierung Funkion 1
- /F20/ Funktion 2 des Systems

Die Funktionalen Anforderungen sollen mithilfe von Use-cases erhoben werden. Die Use-cases sollen in Anhang A detailliert beschrieben werden.

4. Abnahmekriterien

Beschreiben Sie hier, wie die Anforderungen bei der Abnahme auf ihre Realisierung überprüft werden können.

Definieren Sie hier mindestens ein Abnahmekriterium

- /A10/ Abnahmekriterium 1
- /A20/ Abnahmekriterium 2

Anhang

Anhang A. Use-cases

An dieser Stelle können detaillierte Use-cases angegeben werden Diagram

Use Case 1:

- Name: *Name des Use-cases*
- Akteure: *Akteur1, Akteur2, ...*
- Vorbedingungen: *Was muss vor Beginn des Ablaufs gelten*
- Standardablauf
  - Schritt 1
  - Schritt 2
- Nachbedingungen Erfolg: *Was muss nach dem Ende des erfolgreichen Ablaufs gelten*
- Nachbedingung Sonderfall: *Was gilt nach dem Ende, wenn der Ablauf fehlgeschlagen ist*

Sonderfall 1a: Ausnahme 1

- Ablauf Sonderfall 1a
  - Schritt 1
  - Schritt 2

Sonderfall 1b: Ausnahme 2

- Ablauf Sonderfall 1b
  - Schritt 1
  - Schritt 2

Aufgabe

View on GitHub

Software Engineering - GymInf 22

Vorlesungsseite für die Vorlesung Softwaretechnik im Rahmen des Programms GymInf

Projektschritt 1: Pflichtenheft und Projektplan

Bevor Sie die folgenden Aufgaben bearbeiten, vergewissern Sie sich bitte, dass Sie das Dokument Projektübersicht gelesen haben, die entsprechenden Branches auf Ihrem Repository angelegt haben, und Ihnen die Struktur vom Projekt klar ist.

Allgemeiner Hinweis zur Arbeitsweise

In dieser Aufgabe sind die einzelnen Schritte strikte geteilt und geordnet. Es macht jedoch Sinn, diese Aufgaben iterativ zu lösen und nach der Bearbeitung jeden Schrittes zu schauen, ob daraus neue Erkenntnisse für einen anderen Schritt entstanden sind. Sie können auch schon parallel zur Erstellung des Pflichtenhefts einen Prototypen beginnen zu implementieren. Dies hilft Ihnen die Probleme früh zu erkennen.

Pflichtenheft

Vorbereitung

Schauen Sie sich die Theorie zum Thema Anforderungsanalyse nochmals genau an. Überlegen Sie sich wer ihre Stakeholder sind und wie diese von den Änderungen betroffen sind.

Einleitung und allgemeine Beschreibung

Schauen Sie sich das Template für das Pflichtenheft an.

Schreiben Sie die Abschnitte 1 - 2 für ihr Projekt.

Schauen Sie sich als Inspiration auch die verlinkten Beispiele an. In den Beispielen wird zwar ein anderes Template verwendet, es werden aber genau dieselben Aspekte definiert.

Hinweis: Sie finden alle Templates als Markdown Files im docs Ordner des Github Repositories.