Softwareanforderungsanalyse

Einführung

Burkhardt Renz

THM, Fachbereich MNI

Wintersemester 2018/19

Übersicht

- Was ist Softwareanforderungsanalyse?
  - Definitionen
  - Dimensionen der Softwareanforderungsanalyse
  - Ziele und Anforderungen
  - Aufgaben in der Softwareanforderungsanalyse
- Anforderungen im Software-Lebenszyklus
- Warum Softwareanforderungsanalyse?
- Zur Vorlesung

Anforderung (*Requirement*)

1. Eine Bedingung oder Fähigkeit, die von einer Person zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die ein System erfüllen oder besitzen muss, um einen Vertrag, eine Norm oder ein anderes, formell bestimmtes Dokument zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Fähigkeit nach 1 oder 2. (IEEE 610.12-1990)

Requirements Engineering

1. Systematisches, diszipliniertes und quantitativ erfassbares Spezifizieren, d.h. Erfassen, Beschreiben und Prüfen von Anforderungen an ein System.
2. Verstehen und beschreiben, was die Kunden und Anwender wünschen und/oder brauchen.
3. Spezifikation und Verwaltung von Anforderungen mit dem Ziel, das Risiko zu minimieren, dass ein System entwickelt wird, welches den Kunden nicht nützt oder ihre Erwartungen nicht erfüllt.

Diskussion der Begriffe „Softwareanforderungsanalyse“, *Requirements Engineering*, „Anforderungstechnik“

Was ist Requirements Engineering? Die drei W

We need to discover, understand, formulate, analyse and agree on what problem be should solved, why such a problem needs to be solved and who should be involved in responsibility of solving that problem. Broadly, this is what requirements engineering is all about.

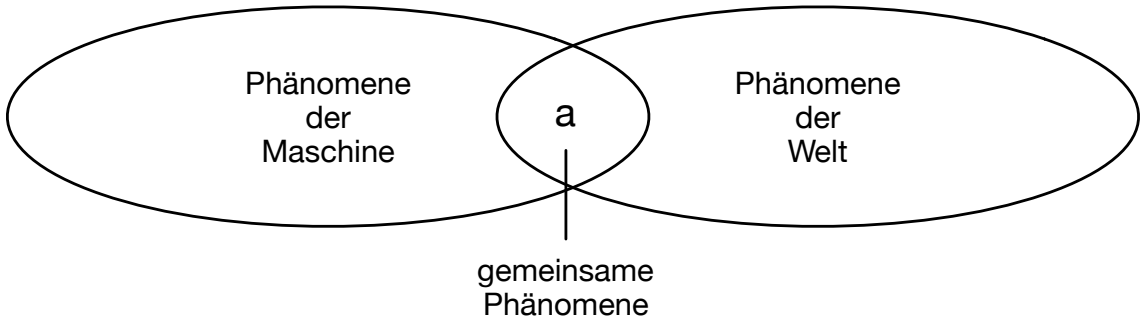
– van Lamsweerde

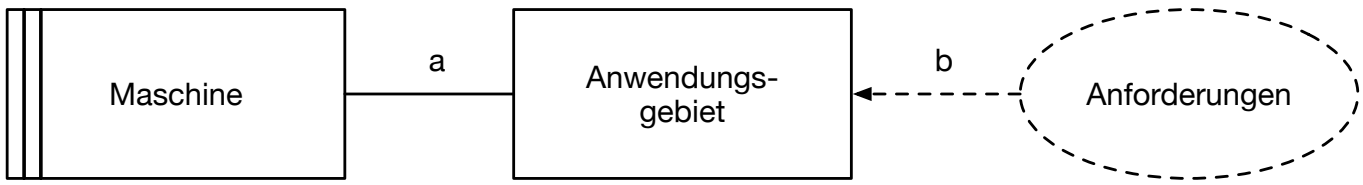
Bedeutung der Softwareanforderungsanalyse

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, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

– Frederick P. Brooks

Was sind Anforderungen?





Definition: Anforderung und Spezifikation

Anforderung

A requirement is a desired relationship among phenomena of the environment of a system, to be brought about by the hardware/software machine that will be constructed and installed in the environment.

Spezifikation

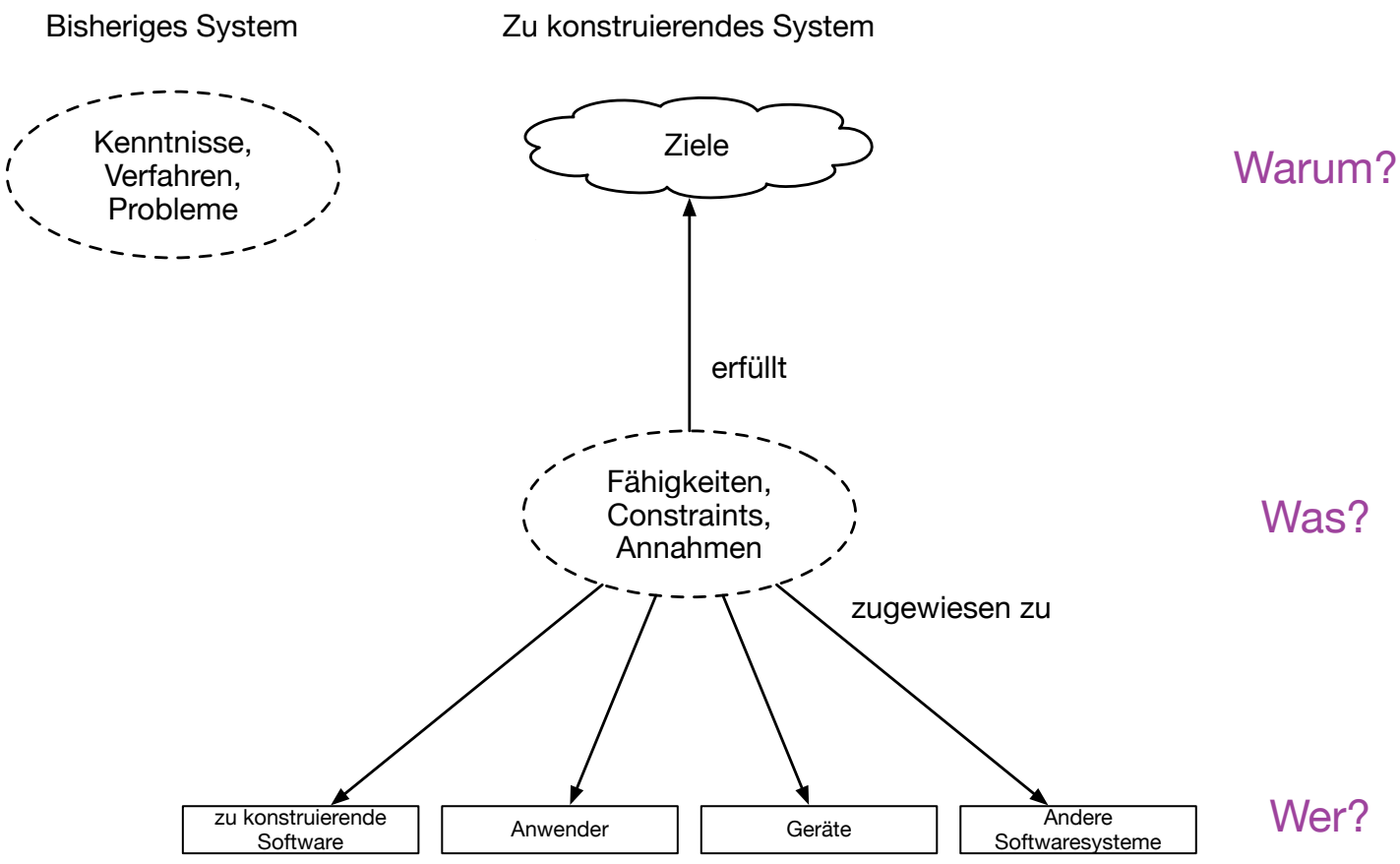
A specification describes machine behaviour sufficient to achieve the requirement.

Aufgabe der Anforderungsanalyse

$S, E \vdash R$

– Michael Jackson, Pamela Zave

Die drei Dimensionen der Anforderungsanalyse



Warum?

Warum ein neues System konstruieren?

- Identifizieren und Analysieren der Ziele des zu konstruierenden Systems
- Defizite des existierenden Systems erfassen
- Anwendungsgebiet verstehen lernen
- Alternativen mit einbeziehen
- Konfligierende Ziele austarieren

Was?

Welche Features soll das zu konstruierende System haben?

- Identifizieren und Definieren der funktionalen Eigenschaften
- Nicht-funktionale Qualitätsmerkmale berücksichtigen
- Die *richtigen* Features finden!
- Präzise Formulierungen, verständlich für alle Beteiligten
- Verfolgbarkeit der Realisierung der Ziele, des *Warum*

Wer?

Wer ist verantwortlich für die Features?

- Zuordnung der Features zu
  - der zu konstruierenden Software
  - den Anwendern der Software
  - Geräten, die im zu konstruierenden System verwendet werden
  - anderen Softwaresysteme, die im Kontext verwendet werden
- Untersuchung der Fähigkeiten und Eigenschaften aller beteiligten Komponenten
- Richtigen Grad der Automation ermitteln, alternative Lösungen erwägen

Ziele vs. Anforderungen

Anforderung

Eine zu erreichende *Eigenschaft*, meist konkret, beschreibt ein Feature.
In der Regel wird eine *Anforderung* durch das Wirken *eines* Agenten des Systems erreicht.

Ziel

Ein erwünschter *Zustand*, der durch das Zusammenwirken von Agenten des Systems (etwa Anwender, Geräten, Softwarekomponenten) erreicht wird.
Ein Ziel erfordert die *Zusammenarbeit* von Agenten.

Arten von Aussagen

Für die Formulierung von Anforderungen und Ziele ist es wichtig zu unterscheiden:

- Beschreibende, deskriptive Aussagen:
  sind Feststellungen über das Anwendungsgebiet oder das System, die einen Sachverhalt beschreiben, der als gegeben zu betrachten ist
- Vorschreibende, präskriptive Aussagen:
  sind gewünschte Eigenschaften, gewünschtes Verhalten des Systems

Arten von Aussagen, 2

- Eine Systemanforderung ist eine präskriptive Aussage, was die Software in Kooperation mit anderen Agenten erreichen soll.
- Eine Softwareanforderung ist eine präskriptive Aussage, was die Software alleine erreichen soll, also nur in Bezug auf Phänomene, die Software und Welt teilen.
- Eine Eigenschaft des Anwendungsgebiets ist eine deskriptive Aussage über die Welt in der das System arbeiten soll.
- Eine Annahme ist eine präskriptive Aussage über die Welt, die aber nicht durch das System erreicht werden soll, sondern für dieses vorausgesetzt werden kann.
- Eine Definition ist die Festlegung einer präzisen Sprechweise.

Beispiele für die Arten von Aussagen

- Systemanforderung: Die Türen des Zugs müssen immer geschlossen sein, solange er in Bewegung ist
- Softwareanforderung: Die Outputvariable `doorsState` hat den Wert `closed`, wenn die Inputvariable `measuredSpeed` ungleich `0` ist
- Eigenschaft: Der Zug bewegt sich genau dann, wenn seine Geschwindigkeit nicht `0` ist
- Annahme: Die gemessene Geschwindigkeit in `measuredSpeed` ist genau dann ungleich `0`, wenn die physikalische Geschwindigkeit ungleich `0` ist
- Definition: Das Phänomen, dass der Zug von einem Gleissegment in ein anderes wechselt, bezeichnen wir als *Bewegung* des Zugs.

Arten von Anforderungen

- Funktionale Anforderungen
- Nicht-funktionale Anforderungen
- Projektanforderungen
  Termine, Kosten, Projektattribute
- Prozessanforderungen

Qualitätsmerkmale

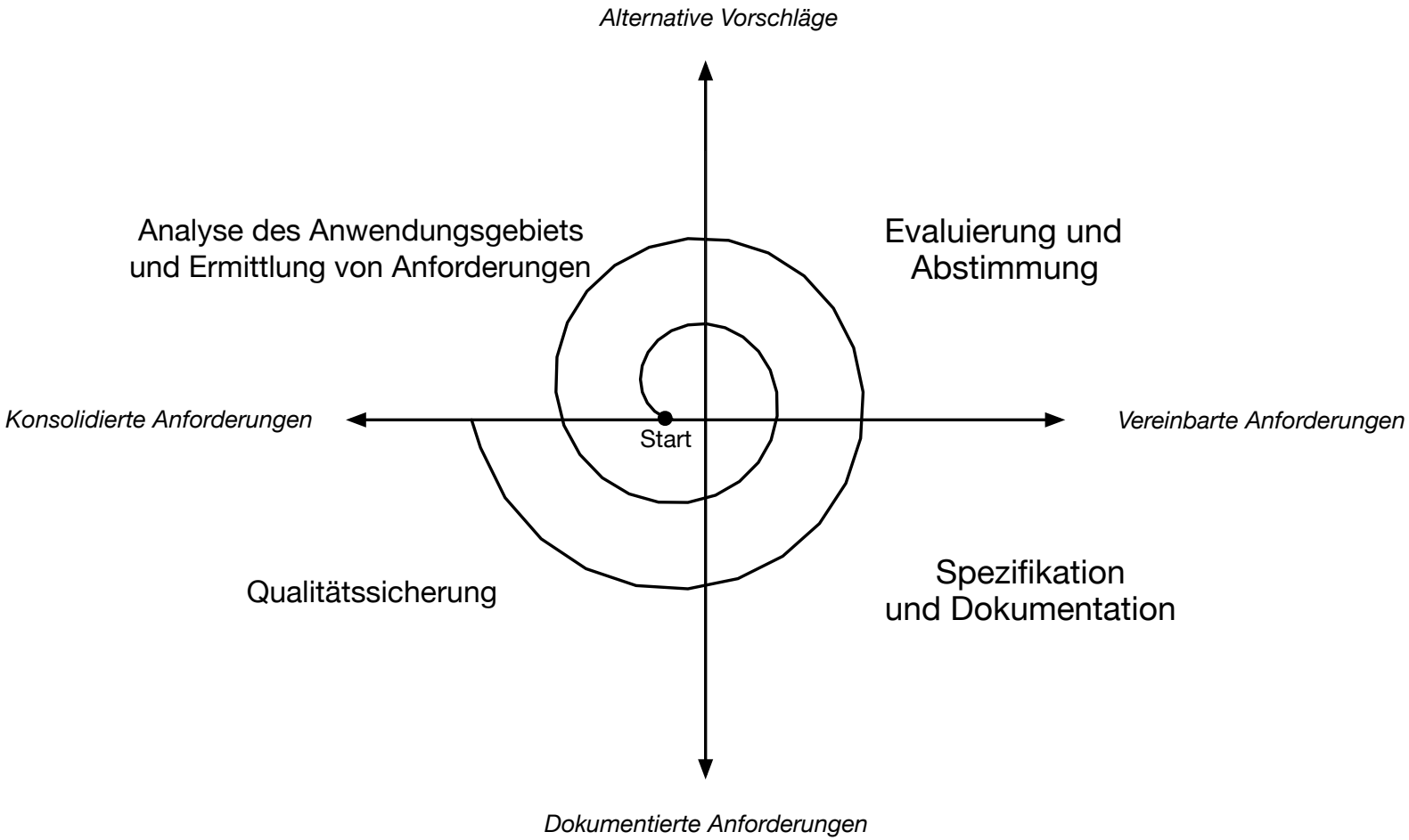
Was ist Qualität?

Die Qualität eines Softwaresystems ist seine Eignung für den vorgesehenen Zweck (IEEE).

Qualitätsmerkmale?

- Funktionalität
- Benutzbarkeit (*usability*)
- Verfügbarkeit (*availability*)
- Leistungsfähigkeit (*performance*)
- Sicherheit (*security*)
- Änderbarkeit (*modifiability*)
- Testbarkeit (*testability*)

Spiralmodell für die Softwareanforderungsanalyse



Qualitätsmerkmale der Anforderungsdokumentation

- vollständig
- konsistent
- adäquat
- eindeutig
- verständlich, gut strukturiert
- prüfbar
- relevant, risikogerecht
- verfolgbar

Übersicht

Rolle im Software-Lebenszyklus

- Anforderungen sind Input für eventuelle Prototypen
- Spezifikation der Anforderungen wesentlicher Input für Architekturentwurf und -evaluierung
- Anforderungen sind die Referenz für die Qualitätssicherung, also insbesondere wesentlich fürs Testen
- Anforderungsspezifikation hilft für die Benutzerdokumentation
- Entwicklung der Anforderungsspezifikation führt zu und leitet die Softwarewartung
- Anforderungsspezifikation wesentlich fürs Projektmanagement

Agile Prozesse und Requirements Engineering

Definition

In agile requirements engineering, the requirements are elicited, analysed and specified in an ongoing and close collaboration with a customer or customer representative in order to achieve high reactivity to changes in the requirements and in the environment. Continuous requirement re-evaluation is vital for the success of the solution system, and the close collaboration with the customer or cumstomer representative is the essential method of requirements and system validation.

– Ville T. Keikkilä et el.

Vor- und Nachteile des agilen RE

Vorteile

- Schlankerer Prozess
- Einbindung des Kunden hilft beim Entdecken und Verständnis der Anforderungen
- Schnellere Reaktion auf Änderungen
- Schnellere Rückmeldung durch inkrementelle Entwicklung
- Bessere Beziehung zum Kunden

Probleme

- Überforderung des Kundenrepräsentanten
- Nicht-funktionale Qualitätsmerkmale kommen in *user stories* nicht vor
- Wachsende *technische Schuld*
- Abhängigkeit von implizitem Wissen über die Ziele und Anforderungen

Agilität und Softwareanforderungsanalyse – Fazit

- Softwareanforderungsanalyse *muss nicht* als großer Vorwegschritt gemacht werden.
- Spiralmodell unterstützt inkrementelles Vorgehen.
- Risiken kann man in den Vordergrund stellen und die Analyse auf die risikoreichen Aspekte konzentrieren.
- aber: „Iterating on code is more expensive than iterating on concepts“ (Wiegers)
- aber: „We would obviously not like our air traffic control, transportation, power plant, medical operation or e-banking systems to be obtained through agile development of critical parts of the software“ (van Lamsweerde)
- Also: Rigidität des Vorgehens und Grad der (formalen) Überprüfung hängt von der Art des zu konstruierenden Systems ab.

Übersicht

- Was ist Softwareanforderungsanalyse?
- Anforderungen im Software-Lebenszyklus
- Warum Softwareanforderungsanalyse?
  - Gründe für Scheitern von Softwareprojekten
  - Ergebnisse der Softwareanforderungsanalyse
- Zur Vorlesung

Gründe für das Scheitern von Softwareprojekten

- Untersuchungen zeigen, dass etwa 40% der Fehler in Software durch falsche oder falsch verstandene Anforderungen entstehen.
- Diese Situation hat sich trotz Erhöhung der Qualität in den letzten 20 Jahre nicht verändert.
- Kosten für die Behebung von Fehlern ist abhängig von der Dauer ihres Bestehens, d.h. Fehler früh in der Entwicklung sind am teuersten.
- Also: möglichst wenige Anforderungsfehler machen und möglichst Fehler früh finden.
- Andererseits: „It's virtually impossible to write excellent requirements at the very beginning of a complex project. Only when we see the interfaces and allow programming to start can we begin to truly refine our ideas for the requirements.“ (Kovitz)

Was erreicht man mit Anforderungsanalyse?

- Geringere Kosten
  - Weniger Fehler während der Implementierung
  - Unterstützung und Vorbereitung in allen Phasen der Entwicklung
  - Geringere Kosten für Fehlerkorrekturen nach Auslieferung
  - Geringere Kosten für die Wartung
- Weniger Risiken
  - Kundenerwartungen besser erfüllen
  - Bessere Prognostizierbarkeit von Terminen und Kosten

Fazit

„Die wirtschaftliche Wirkung von Requirements Engineering ist immer indirekt; das RE selbst kostet nur!“ (Martin Glinz, Uni Zürich)

Übersicht

- Was ist Softwareanforderungsanalyse?
- Anforderungen im Software-Lebenszyklus
- Warum Softwareanforderungsanalyse?
- Zur Vorlesung
  - Thematische Gliederung
  - Materialien
  - Begleitendes Projekt

Teil 1: Aufgaben der Softwareanforderungsanalyse

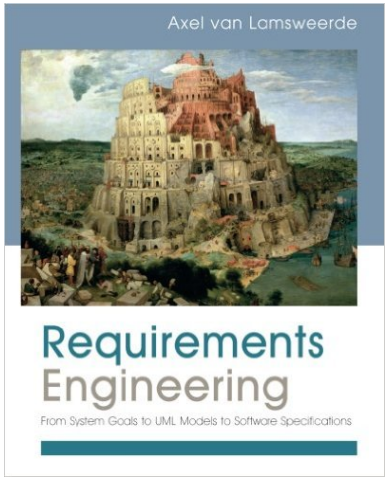
1. Einführung (diese Vorlesung)
2. Ermittlung von Zielen und Anforderungen
3. Evaluation von Anforderungen
4. Spezifikation und Dokumentation von Anforderungen
5. Validierung von Anforderungen
6. Evolution von Anforderungen

Teil 2: Modellbasiertes Vorgehen in der Anforderungsanalyse

1. Ziele und ihre Modellierung
2. Risikoanalyse
3. Objektmodell
4. Agenten und ihre Verantwortlichkeiten
5. Modell der Operationen
6. Modell des Verhaltens des Systems
7. Zusammenfassung zum Vorgehen

Literatur und andere Quellen

Axel van Lamsweerde
Requirements Engineering From System Goals to UML Models to Software Specifications
John Wiley & Sons, 2009



Klaus Pohl, Chris Rupp *Requirements Engineering Fundamentals*
Rocky Nook, 2015

Martin Glinz *Vorlesungsskript Requirements Engineering I*
Universität Zürich, 2008
`http://www.ifi.uzh.ch/rerg/courses/archives/hs08/re-i.html`

Literatur und Links

Benjamin L. Kovitz *Practical Software Requirements: A Manual of Content and Style* Manning, 1998

Karl Wiegers *Software Requirements* Microsoft Press, 2013

Jeremy Dick, Elizabeth Hull und Ken Jackson *Requirements Engineering* Springer, 2017

International Requirements Engineering Board IREB
`https://www.ireb.org/de/`

Mehr Literatur

Michael Jackson *Software Requirements & Specifications: a lexicon of practice, principles and prejudices* acm Press, 1995

Michael Jackson *Problem frames: Analyzing and structuring software development problems* Addison-Wesley, 2001

Projekt

Drei Fallstudien aus dem Buch von Lamsweerde in Gruppenarbeit:

- Bibliotheksverwaltung
- Zugsteuerung
- Terminvereinbarungen