



## Kapitel 1: Requirements Engineering

Dr. Dominik Haneberg

### Software Engineering 2

Wintersemester 2018/2019

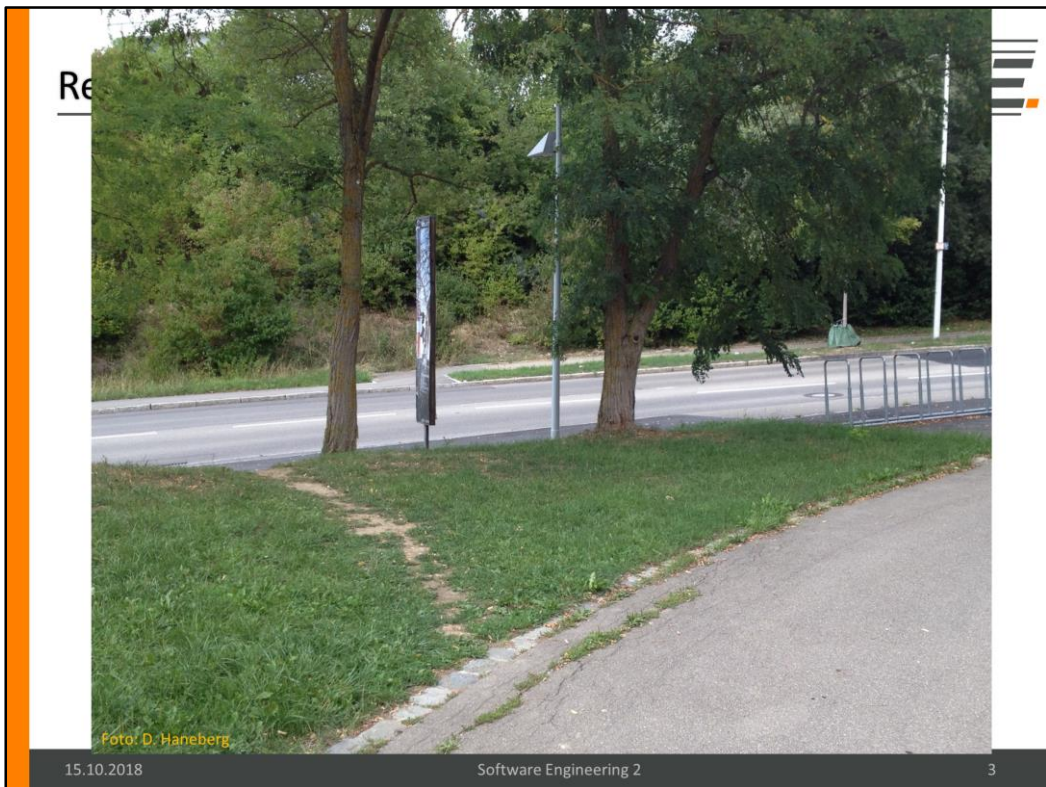


Abschnitt 1.1

# MOTIVATION UND INHALTE

An Tafel: Was erwarten Sie von Requirements Engineering, was stellen Sie sich unter dem Begriff vor?

Oder ausführlichere Aufgabe „Stellenanzeige“



[-> Trampelpfad-Ulm]

- Requirements Engineering: Das Problem
- Requirements Engineering Begriffe und Aufgaben
- Kontext, Stakeholder, Ziele und Anforderungen
- Arten von Anforderungen
- Anforderungsgewinnung
- Anforderungsdokumentation
- Qualität und Prüfung von Anforderungen

**Anforderungsgewinnung** hat Techniken und Methoden zum Inhalt, aus **Dokumenten, Anwendern** und **Stakeholdern** (möglichst alle) funktionalen und nicht-funktionalen **Anforderungen** widerspruchsfrei und vollständig zu **ermitteln**.

Bei **Anforderungsdokumentation** geht es insbesondere um verschiedene **Sprachen** zur **Dokumentation von Anforderungen**, von textuellen Repräsentationen bis zu einer formalen Sprache.

**Anforderungsverwaltung** beschäftigt sich mit **Organisation** und **Versionierung** von Anforderungen, Anforderungsbasislinien, Change-Requests u.ä.

*„There is nothing so useless as doing efficiently that which should not be done at all.“*  
— Peter Drucker

*„The hardest single part of building a software system is deciding precisely what to build.“*  
— Frederick P. Brooks

*„It isn't that they can't see the solution. It is that they can't see the problem.“*  
— Gilbert Keith Chesterton

*„The cheapest, fastest and most reliable components of a computer system are those that aren't there.“*  
— Gordon Bell

*„Bittet, so wird euch gegeben“*  
— Matthäus 7,7

*„I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone.“*  
— Bjarne Stroustrup

Das Brooks Zitat beschreibt das Trampelpfadproblem.

Chesterton beschreibt eine der großen Herausforderungen des Business Analyse.

Bell: Was man nicht braucht, sollte man nicht einbauen, da es verursacht nur Kosten.

Stroustrup: Featureritis ist eine Gefahr.

Drucker: Unnötig Arbeit ist Verschwendung, deshalb möglichst nicht machen.

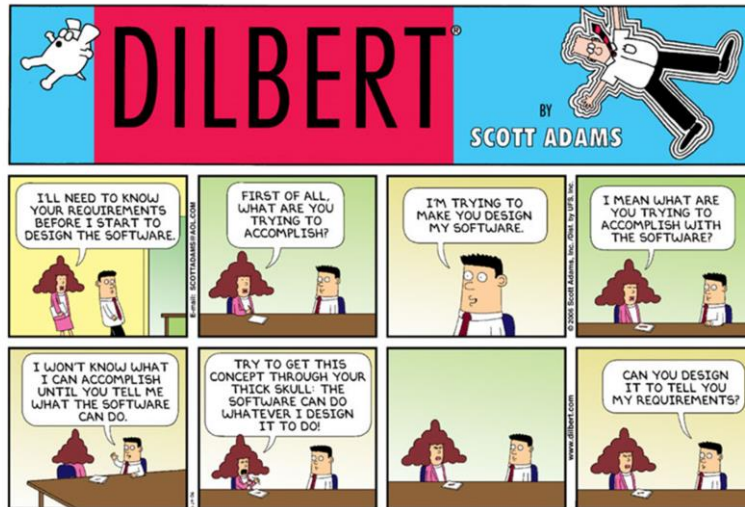
Drucker: Ameri.-Österr. Ökonom, Vater der Managementlehre

Bell: Amerik. Informatiker, Entwickler der DEC VAX und einiger PDP-Maschinen

Brooks: Amerik. Informatiker, Entwickler von OS/360, Autor von „Mythical Man Month“: „Adding manpower to a late project makes it later.“

Chesterton: Brit. Journalist und Schriftsteller, Autor der „Pater Brown“-Krimis

# Der Stakeholder, das unbekannte Wesen



© Scott Adams, Inc./Dist. by UFS, Inc.

Abschnitt 1.2

# GRUNDBEGRIFFE UND AUFGABEN

# Definition Requirements Engineering

**Requirements Engineering** (RE), im Deutschen gelegentlich **Anforderungsmanagement** genannt, ist eine Managementaufgabe für die effiziente und fehlerarme Entwicklung komplexer Systeme. Es umfasst die Themengebiete **Anforderungsdefinition** (englisch **requirements definition**) und **Anforderungsverwaltung** (englisch **requirements management**).

Anforderungsdefinition beinhaltet dabei die Teilgebiete **Anforderungsermittlung** (englisch **requirements elicitation**), **Anforderungsdokumentation** (englisch **requirements documentation**) und **Anforderungvalidierung** (englisch **requirements validation**), während Anforderungsverwaltung Maßnahmen zur **Steuerung**, **Kontrolle** und **Verwaltung** von Anforderungen, also Risikomanagement, Änderungsmanagement und Umsetzungsmanagement umfasst.

Wikipedia: [http://de.wikipedia.org/wiki/Requirements\\_Engineering](http://de.wikipedia.org/wiki/Requirements_Engineering)

Bei der Anforderungsverwaltung ist Traceability das wichtigste Thema, also Nachverfolgbarkeit

Beispiel für Problem: Anforderung dokumentiert oder Lösung? [->Anforderungen\_nicht\_Lösungen, [->Anforderungen\_nicht\_Lösungen-Deutsch]



### Requirements Engineering

1. Das systematische, disziplinierte und quantitativ erfassbare Vorgehen beim Spezifizieren (Erfassen, Beschreiben und Prüfen) von Anforderungen an ein System
2. Verstehen und Dokumentieren, was der Kunde will oder braucht
3. Spezifikation und Verwaltung von Anforderungen, mit dem Ziel, das Risiko zu minimieren, etwas zu entwickeln, was dem Kunden nicht gefällt oder nicht nützt

**1.: Systemanforderungen erfassen und gut dokumentieren**

**2.: Kundenbedürfnisse erfüllen helfen**

**3.: Projektrisiken minimieren**

**Requirements Engineering** ist in den Software-Entwicklungsprozess eingebettet und kann als eine **Phase** (typischerweise sehr früh) stattfinden oder **immer wieder im Projektverlauf** (iteratives Vorgehen). **Im Extremfall ist immer Requirements Engineering (agile Vorgehensweisen).**

Vor nächster Folie: Kurzes Tafelbild zu linearem und inkrementellem Modell [->RE-im-Softwareprozess]

# Was ist Requirements Engineering?

## Technische Sicht

Requirements Engineering ist das systematische und disziplinierte Vorgehen beim Spezifizieren (Erfassen, Beschreiben und Prüfen) und Verwalten von Anforderungen an ein System.

- Typischerweise **Softwaresysteme** oder **software-intensive Systeme**
- Ziel ist eine vollständige, unmissverständliche und widerspruchsfreie Spezifikation
- Klingt nach Aktenbergen und Bürokratie
- Wo sind die Menschen?
- Ist das realistisch?

## 3 Sichtweisen!

**Typischerweise am Anfang:** Das stimmt zwar insofern, dass man ohne etwas RE gemacht zu haben, nicht Designen oder Implementieren kann. Es heißt aber **nicht**, dass das **komplette** Requirements Engineering **am Anfang** des Entwicklungsprozesses abgehandelt wird. In iterativen Vorgehensmodellen ist Requirements Engineering ein Prozess, der **parallel** zu Design und Entwicklung über (fast) die ganze Projektzeit läuft. Es ist in diesem Fall nicht die erste Phase, die am Anfang des Prozesses bearbeitet und dann abgeschlossen wird.

Zum letzten Punkt: Studis fragen. Antwort: Nein! Requirements sind nie fertig

# Was ist Requirements Engineering?

## Kundenorientierte Sicht

Requirements Engineering — Verstehen und Dokumentieren, was der Kunde will oder braucht.

- **Menschenzentrierte Sicht**
- Ziel sind **zufriedene Kunden**
- Was sind Kunden?
- Warum wollen oder brauchen?
- Warum nicht gleich programmieren?

Beim Begriff Kunden sollte man aufpassen: Meist meint „**Kunde**“ den, der am Ende die Rechnung bezahlt bzw. der den Auftrag erteilt hat. Dieser ist aber **oft nicht der Benutzer der Software**. Er ist als Quelle von Anforderungen an das System nur teilweise nützlich. **Daher ist der Blick auf alle Stakeholder** (Beteiligte/Interessenvertreter) zu richten, die von dem neuen System betroffen sind.

Es ist Aufgabe der Businessanalyse, die Bedürfnisse und Ziele der Stakeholder zu erkennen, und dann Anforderungen zu formulieren, die diese Ziele erreichen helfen.

Ein anderes Beispiel für „**was der Kunde will**“ vs. „**was der Kunde braucht**“:

Sie haben ein Autohaus und zu Ihnen kommt der Kunden: „Ich fahr’ da immer durch eine Kurve und da sitzt ein Hase. Dem will ich ausweichen und fahr’ dann gegen einen Baum. Machen Sie mir eine bessere Stoßstange.“

Wenn man dem Kunden dann eine stabilere Stoßstange baut, hat man ihm nicht geholfen. Er will zwar die Stoßstange, er braucht aber eigentlich eine bessere Strategie, um um die Kurve zu fahren.“

STUDIS nach Einschätzung fragen:

Bsp. für eine Lösung, die dem Kunden gibt, was er denkt, das sein Problem lösen würde, **in Wirklichkeit aber nur ein Symptom bekämpft**: „**Die Mikrowelle soll innen aus Edelstahl sein.**“

# Was ist Requirements Engineering?

## Risikoorientierte Sicht

Requirements Engineering — Dokumentieren und Verwalten von Anforderungen mit dem Ziel, das **Risiko** zu minimieren, ein System zu entwickeln, das vom Kunden nicht akzeptiert wird (ihm nicht gefällt oder nicht nützt).

- Man stelle sich die Frage: „Wie viel müssen wir tun, damit das Risiko so klein wird, dass wir bereit sind es zu akzeptieren?“
- Aufwand für das Requirements Engineering sollte umgekehrt proportional zur Größe des Risikos sein, das man bereit ist, einzugehen.

15.10.2018

Software Engineering 2

12

Anforderung schon bekannt?

Ja: Nicht Spezifizieren

Nein: Ist das Risiko akzeptabel gering, dass der Kunde das entwickelte System nicht akzeptiert?

Ja: Nicht Spezifizieren

Nein: Anforderungsspezifikation nötig

„Keine Zeit für vollständige Spezifikation.“

„Ist uns zu teuer.“

sind keine guten Ansätze

**RE ist zentral für das Risikomanagement in Softwareprojekten.**

**CHAOS-Report der Standish-Group nennt als Top-2 Gründe für Projektscheitern:**

- **Unvollständige Requirements**
- **Fehlende Einbindung der Stakeholder**

# Aufgaben des Requirements Engineering



Vor nächster Folie: Grafik zu Fehlerkosten [->Fehlerkosten]

## Erwarteter Nutzen

- **Kosten reduzieren**

- Geringere Herstellungskosten
- Weniger Nachbesserungen im Betrieb
- Geringere Wartungskosten

Mehr verdienen

- **Risiken minimieren**

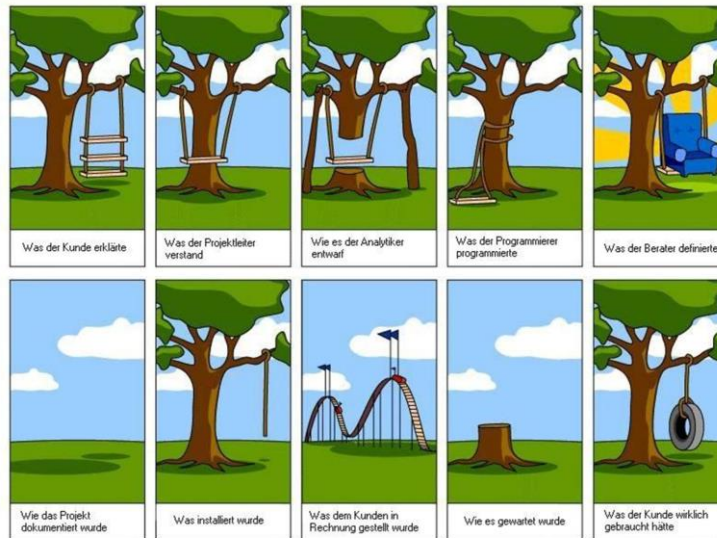
- Kundenerwartungen erfüllen
- Termine und Kosten einhalten

Kundenzufriedenheit  
verbessern

**Der Nutzen, den RE liefert ist immer indirekt.** RE selber kostet nur und nur über Sekundäreffekte entsteht (möglicherweise) eine positive wirtschaftliche Wirkung

Vor nächster Folie: Grafik Wirtschaftlichkeit von Anforderungsspezifikation [->Wirtschaftlichkeit-RE]

## Anforderung im Wandel der Zeit



<http://projectcartoon.com/create/>

15.10.2018

Software Engineering 2

15

George Bernard Shaw (1856-1950): „Das größte Problem mit der Kommunikation ist die Illusion, sie sei gelungen.“

Dazu 2 Versuche:

[Übung Stille Post]: [-> Anforderung-Stille-Post]

[Übung Zeichnung]: [-> Aufgabe Zeichnung, -> AbstraktesBild]