

5 Requirements Engineering – Einführung

5.1 Was ist Requirements Engineering?

Erste Näherung:

Requirements Engineering (Anforderungstechnik) ist das systematische, disziplinierte und quantitativ erfassbare Vorgehen beim Spezifizieren (d.h. Erfassen, Beschreiben und Prüfen) sowie beim Verwalten von Anforderungen an Software.

- Ziel ist eine **vollständige, eindeutige, widerspruchsfreie** Spezifikation aller Anforderungen
- Typisch als **erste Phase** eines Projekts
- Riecht nach **Papier** und **Bürokratie**
- Wo sind die Menschen in diesem Prozess?
- Ist das Ziel überhaupt **realistisch**?

Zweite Näherung

Requirements Engineering (Anforderungstechnik) – Verstehen und Beschreiben, was die Kunden wünschen oder brauchen.

- Eine menschenzentrierte Sicht
- Ziel: zufriedene Kunden

- Was sind Kunden?
- Warum wünschen oder brauchen?
- Warum nicht gleich programmieren?

Dritte Näherung

Requirements Engineering (Anforderungstechnik) – Spezifikation und Verwaltung von Anforderungen mit dem Ziel, das **Risiko** zu minimieren, dass Software entwickelt wird, welche den Kunden nicht nützt oder gefällt.

Anforderungen schon bekannt? – ja → nicht spezifizieren

nein



Risiko tolerabel gering, dass der Kunde das entwickelte System nicht akzeptiert?



ja →

nicht spezifizieren

nein



Anforderungsspezifikation
notwendig

Spezifikation als Risikominimierung

„Wir haben keine Zeit für eine vollständige Spezifikation.“

„Ist uns zu teuer!“

„Bei agilem Vorgehen genügen grobe Stories vollständig.“

⇒ **falscher Ansatz**

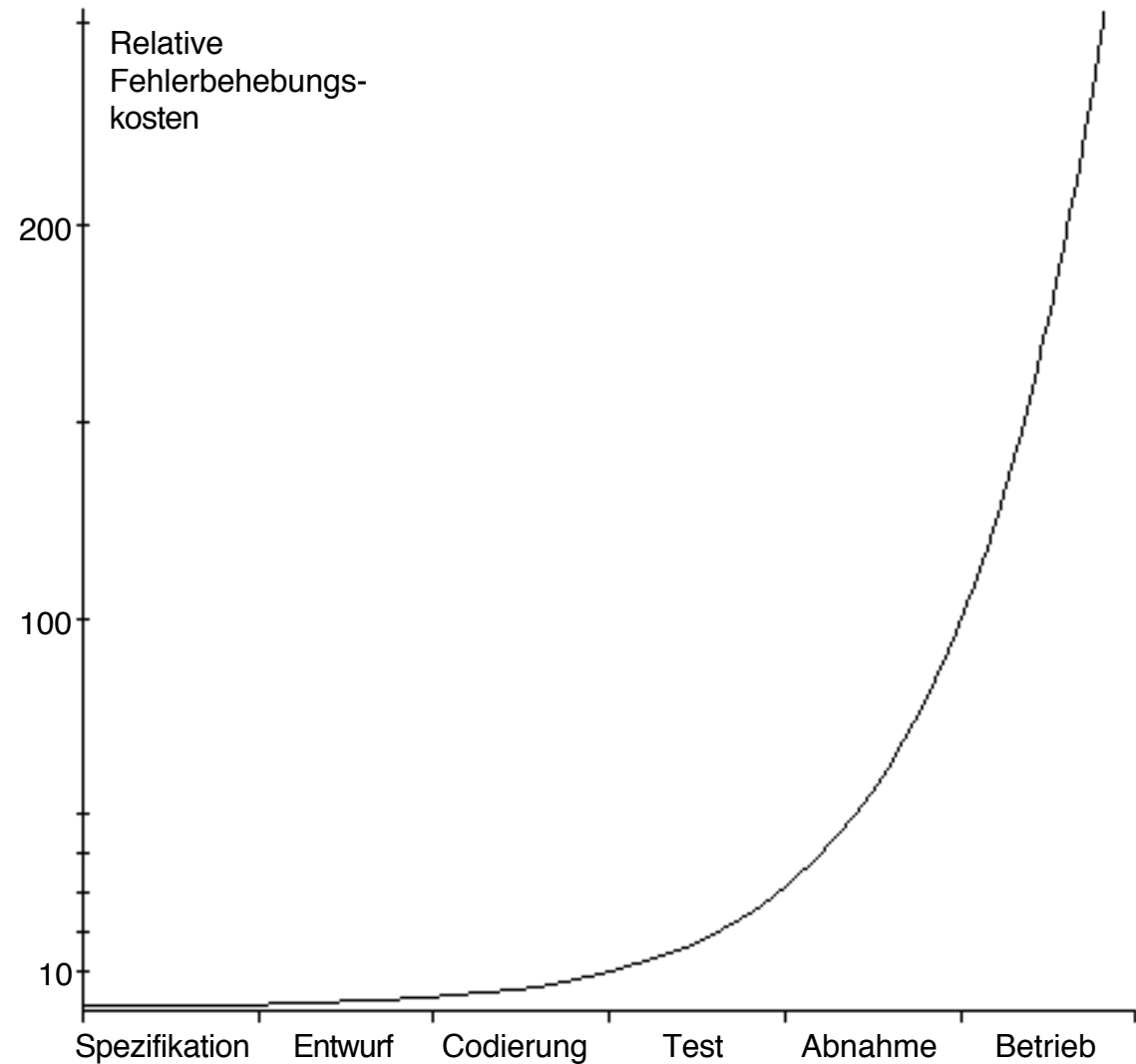
Richtige Frage: „Wie viel müssen wir tun, damit das Risiko eine Größe annimmt, die wir bereit sind zu akzeptieren?“

Merkregel:

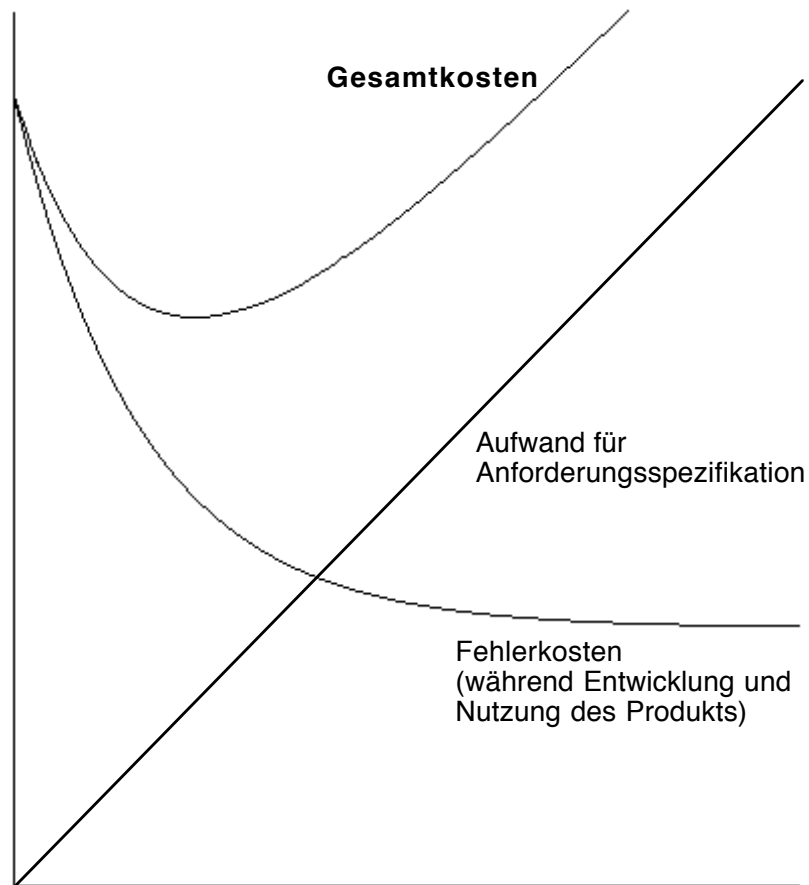
Der *Aufwand* für das Requirements Engineering soll **umgekehrt proportional** zum *Risiko* sein, das man bereit ist, einzugehen.

5.2 Warum ist Requirements Engineering wirtschaftlich?

- **Kosten** für die Behebung von Fehlern...
- ...abhängig von ihrer **Verweildauer** in der Software



Requirements Engineering **senkt** die **Fehlerkosten**:



→ Richtig betriebenes Requirements Engineering ist **wirtschaftlich**.

👉 Die wirtschaftliche Wirkung von Requirements Engineering ist immer indirekt; das RE selbst kostet nur!

5.3 Qualitätsmerkmale

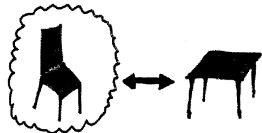
Wie können die Fehlerkosten gesenkt werden?

- ⇒ wenig Anforderungsfehler machen
- ⇒ möglichst viele der dennoch gemachten Fehler möglichst früh finden

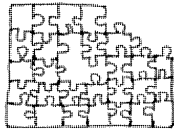
Wir brauchen also

- ① eine gute Anforderungsspezifikation
- ② einen guten Spezifikationsprozess

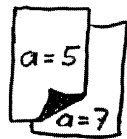
Merkmale einer guten Anforderungsspezifikation



Adäquatheit – das beschreiben, was der Kunde will bzw. braucht



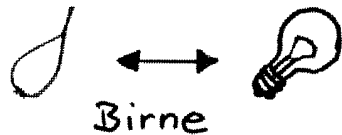
Vollständigkeit – alles beschreiben, was der Kunde will bzw. braucht



Widerspruchsfreiheit – sonst ist die Spezifikation nicht realisierbar

$$\bigwedge_{n \in \mathbb{N}} \bigwedge_{\substack{m_i \\ \text{is in}}} \sum_{i=1}^n |m_i| \geq P \wedge \\ \sum_{i=1}^{n-1} |m_i| < P \leftrightarrow W(m_1 \dots m_{n-1}) \\ \wedge \neg W(m_1 \dots m_{n-1})$$

Verständlichkeit – für den Kunden und für die Informatiker

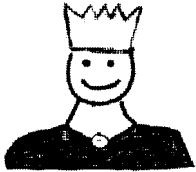


Eindeutigkeit – damit Fehler durch Fehlinterpretationen vermieden werden



Prüfbarkeit – feststellen können, ob das realisierte System die Anforderungen erfüllt

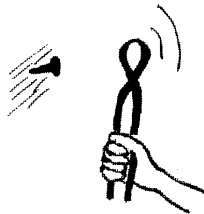
Merkmale eines guten Spezifikationsprozesses



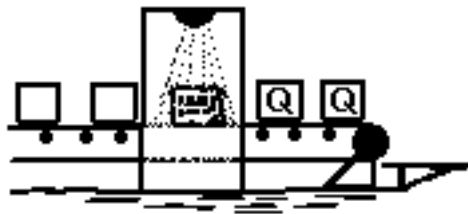
Kundenorientierung



Methodisches und zielgerichtetes Vorgehen



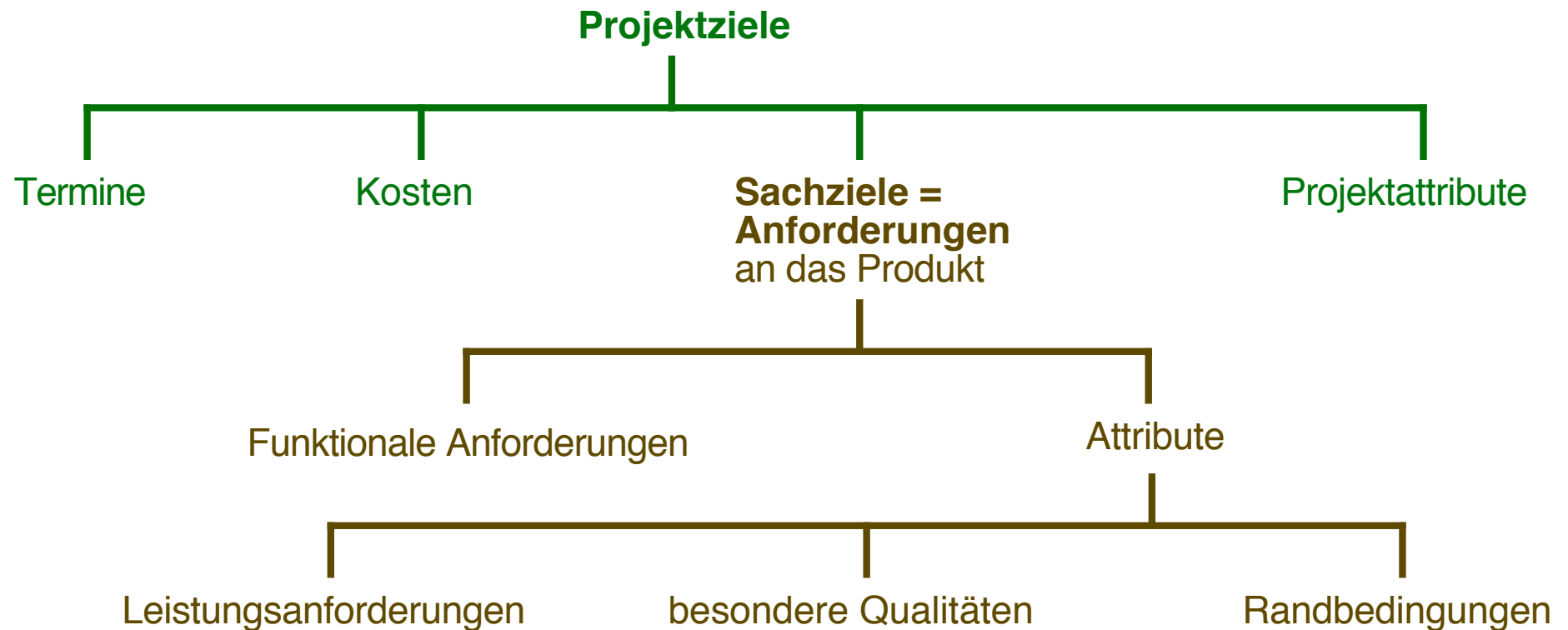
Verwendung geeigneter Mittel



Integration von Erstellung und Prüfung von Anforderungen

5.4 Klassifikation von Anforderungen

Verschiedene Arten von Anforderungen:



Hinweise

- ☆ Statt von Attributen wird häufig von **nicht-funktionalen Anforderungen** gesprochen
- ☆ Die **Abgrenzung** zu funktionalen Anforderungen ist **nicht scharf**

Beispiel:

Eine Anforderung, welche als **besondere Qualität** zu klassifizieren wäre:
„Das System muss den unautorisierten Zugriff auf die Kundenstammdaten verhindern, soweit dies technisch möglich ist“

Durch **Operationalisierung** werden daraus **funktionale Anforderungen**:
„Der Zugriff auf die Kundenstammdaten muss über eine Login-Prozedur mit Passworten geschützt werden“. „Die Kundenstammdaten müssen verschlüsselt gespeichert werden.“

5.5 Priorisierung von Anforderungen

- ☆ **Muss**-Anforderungen – unverzichtbar
- ☆ **Soll**-Anforderungen – wichtig, aber bei zu hohen Kosten verzichtbar
- ☆ **Wunsch**-Anforderungen – schön zu haben, aber nicht essenziell

Nötig bei

- ❑ harten Preisobergrenzen
- ❑ Beschaffung

Nützlich bei

- ❑ Festlegung von Inhalt und Umfang der Inkremente bei inkrementeller Entwicklung
- ❑ Releaseplanung bei der Weiterentwicklung bestehender Systeme

5.6 Aufgaben der Anforderungsspezifikation

Anforderungen

- gewinnen (elicitation)
- analysieren und dokumentieren (analysis and documentation)
- validieren (validation)
- verwalten (requirements management)
 - freigeben (release management)
 - ändern (change management)
 - rückverfolgen (traceability)

Was ist (fast immer) zu tun?

- Die wichtigen **Beteiligten** kennen und einbeziehen
- Das **Problem**, das zum Bedarf für das zu spezifizierende System geführt hat, kennen
- Die drei bis sieben wichtigsten **Systemziele** identifizieren und aufschreiben
- Für jedes Ziel
 - Analysieren und festhalten, zu welchem **Geschäftsziel** das Ziel wie beiträgt
 - Analysieren und festhalten, welchen **Nutzen** das Ziel für welchen der Beteiligten hat
- Die **Schlüsselbegriffe** des Systems und des Anwendungsbereichs in einem Glossar definieren
- Die **Hauptfunktionen** identifizieren und dokumentieren
- Kritische **Randbedingungen** und Risiken identifizieren und dokumentieren

Erschwerende Faktoren (⇒ Mehr Aufwand für das RE notwendig)

- Hohe **Komplexität** des Anwendungsbereichs
- Entwicklungsteam mit dem Anwendungsbereich **nicht vertraut**
- **Viele Beteiligte**
- **Verteilte** Entwicklung und/oder Beteiligte
- **Lange Durchlaufzeit**
- **Sicherheitskritische** Anforderungen
- Hohe **Projektrisiken**

5.7 Prinzipien des Requirements Engineerings

- Die **Betroffenen** kennen: Alle Betroffenen (stakeholders) kennen und berücksichtigen
- **Ziele** identifizieren: Wenige klar formulierte und überprüfbare Ziele sind wichtiger als eine Fülle von Detailanforderungen
- **Randbedingungen** erheben: Eine Systementwicklung kann an nicht erkannten Randbedingungen scheitern
- Den **Wert** berücksichtigen: Kosten und Nutzen der Realisierung einer Anforderung
- **Adäquat** spezifizieren: Die Anforderungsspezifikation dokumentiert genau das, was die Betroffenen wollen/brauchen
- Anforderungen **messbar** spezifizieren: Nur dann sind sie wirklich nützlich
- **Konsens** finden: Verschiedene Betroffene haben unterschiedliche Vorstellungen und Bedürfnisse
- **Validieren** und Verifizieren: Das Richtige spezifiziert? Richtig spezifiziert?