



Requirements Engineering (Anforderungserhebung)

Übersicht

2

- 1. Was ist Requirements Engineering?**
- 2. Wozu Requirements Engineering?**
- 3. Anforderungen**
- 4. Quellen für Anforderungen**
- 5. Anforderungsdokument**

- dt. Anforderungserhebung
- Unterteilbar in
 - Anforderungsaufnahme
 - Anforderungsanalyse
 - Anforderungsspezifikation
 - Anforderungsbewertung *(IEEE)*
- Teil des Anforderungs-Management Prozesses
 - Dazu gehören noch Änderungsmanagement, Risikomanagement und Umsetzungsmanagement

4

“Unter Requirements Engineering verstehen wir das Erfassen, Analysieren, Entwickeln, Strukturieren, Dokumentieren und Verwalten von Anforderungen und Lösungskonzepten.

Ziel ist die Erarbeitung einer präzisen und konsistenten Anforderungs- und Systemspezifikation, die den Zielen und Anforderungen der beteiligten Stakeholder in der Systementwicklung entspricht.”

TU München

Fakultät für Informatik

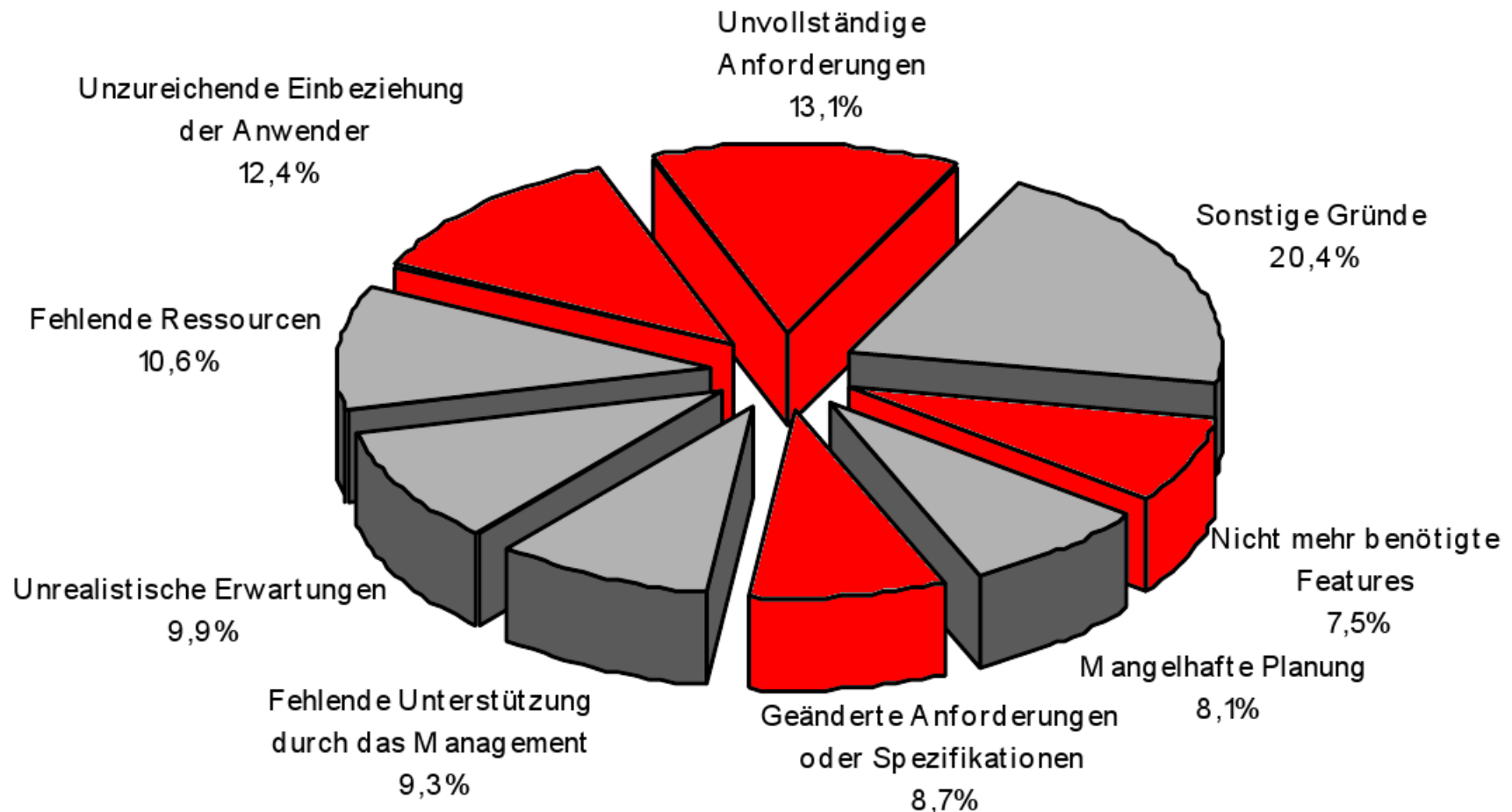
Wozu Requirements Engineering?

5

- 30 - 40% aller IT-Projekte werden vorzeitig abgebrochen
- 70% der Projekte versagen bei der Realisierung geforderter Features
- Geplante Projektkosten werden im Durchschnitt um 89%, die geplante Projektlaufzeit sogar um ca. 120% überschritten
- 45% der realisierten Funktionen werden niemals genutzt

Quelle: Standish Group & Scientific American 1996

Woran scheitern IT-Projekte?



Quelle: Standish Group & Scientific American 1996

7

- Fast alle IT-Manager halten das Anforderungsmanagement für sehr wichtig
- trotzdem gehen nur die wenigsten wirklich systematisch vor
- Viel zu schnell wird umgesetzt und programmiert

(Umfrage FHS St. Gallen 2007)

Warum Anforderungsmanagement?

8

- Dynamik des IT-Umfelds
- Zunehmende Komplexität
- Kurze Entwicklungszeiten
- Einhaltung von Budget und Zeitplanung
- Produktqualität
- Abwehr von juristischen Problemen und wirtschaftlichem Schaden

Anforderungen

- Eine Anforderung ist in der Regel ein ganzer Satz
keine einzelnen Worte oder Fragmente!
- Vermeidung von Abkürzungen, Akronymen und Fachbegriffen
- Wichtig ist eine klare Aussage, die alle Beteiligten verstehen können
kurze Sätze!
- Es ist zu vermeiden, dass zwei sich ähnliche Anforderungen, zu einer verbunden werden
Konjunktionen sollen nicht benutzt werden

RFC 2119

- **muss**

Fordert die unbedingte Umsetzung der Anweisung. Genauso: „erforderlich“, „nötig“, „soll“

- **darf nicht**

eine unbedingte negative Anweisung. Genauso: „soll nicht“

- **sollte**

Gibt einen sorgsam abzuwägenden Freiraum bei der Umsetzung von Anforderungen. Genauso: „ist empfohlen“

- **kann**

eine wirkliche optionale Anforderung, deren Realisierung nicht garantiert ist

Ist die Anforderung...

- **korrekt?**

möglich, machbar, legal?

- **komplett?**

ganze Sätze

- **klar?**

unzweideutig und nicht verwirrend

- **konsistent?**

Nicht im Konflikt mit anderen Anforderungen

- **Nachprüfbar?**

Kann geprüft werden, ob das System der Anforderung genügt?

- **Nachvollziehbar?**

Begründet und sinnvoll

- **Realisierbar?**

Umsetzbar im Zeit- und Kostenrahmen

- **Anwenderanforderungen**
 - ***„Die Software soll bei der Kursbelegung prüfen, ob ein Student alle notwendigen Anforderungen erfüllt.“***
 - Beschreibung des Problems
 - In der Sprache des Anwenders
 - Nach Zielen strukturiert
- **Systemanforderungen**
 - ***„Die Software soll in Java geschrieben sein.“***

- **Nicht-funktionale Anforderungen**

Anforderungen an die Qualität

- **Sicherheit/Vertrauenswürdigkeit**

"Eine Benutzerkennung soll nach drei hintereinander fehlgeschlagenen Zugangsversuchen gesperrt werden."

- **Zuverlässigkeit**

"Nach dem Auftreten eines Fehlers der Kategorie 2 muss die volle Leistungsfähigkeit des Systems innerhalb von 2 Stunden wiederhergestellt werden."

- **Verfügbarkeit**

"Das System soll mit Ausnahme von Sonntag täglich von 6 Uhr bis 22 Uhr im Jahresschnitt zu mindestens 99% verfügbar sein."

- **Zeitverhalten**

"97% der Transaktionen sollen 8 Wochen nach Einführung des Systems in weniger als 2 Sekunden beendet sein. Die maximale Antwortzeit darf 5 Sekunden nicht überschreiten."

- **Verbrauchsverhalten**

„Die Anwendung darf im Betrieb nicht mehr als 200 Megabyte Arbeitsspeicher belegen.“

- **Portabilität**

„Die Anwendung soll unter Windows und Linux 2.6.x lauffähig sein.“

- **Benutzbarkeit und Ergonomie**

„Ein Verkäufer soll per Wizzard ein Verkaufsangebot binnen 10 Minuten eingeben können.“

- **Rahmenbedingungen**

- Rechtliche
- Kulturelle (Anrede, Höflichkeitsformeln, Inhalte)
- Organisatorische
- Physikalische (bei Software weniger wichtig)
- IT-technische

*Hardware, Netzwerk, Betriebssystem,
Datenbanksysteme, Applikations-Server,
Schnittstellen*

- **Entwicklungs- und Produktionsanforderungen**
 - **Entwurfsanforderungen** formulieren Anforderungen an die zukünftige Architektur (Erweiterbarkeit, Wartbarkeit und Wiederverwendbarkeit)
 - Anforderungen an die Implementierung fordern die Verwendung bestimmter Programmiersprachen oder Bibliotheken (auch Namenskonventionen und Styleguides)
 - **Produktionsanforderungen** definieren die Anforderungen an die Einführung und den Betrieb (Installation, Migration, Betreuung, Schulung und Dokumentation)
 - Definition von Anforderungen Werkzeuge (Entwicklungsumgebungen, Design Tools) sichert die Weiterentwicklung eines Systems beim Kunden
 - Zeit- und Kostenrahmen

- **Benutzer**

werden die Software nutzen. Haben oft unterschiedliche Anforderungen

- **Kunden**

haben die Software in Auftrag gegeben

- **Marktanalysten**

ermitteln die Bedürfnisse des Markts (Massensoftware hat keine Auftraggeber)

- **Aufsichtsbehörden**

In verschiedenen Bereichen (Medizin, Banking, ...) muss Software bestimmten Spezifikationen genügen

- **Softwareentwickler**

- Bewusste Anforderungen basieren auf
 - Konkreten Vorstellungen gewünschter oder benötigter Features
 - Mängeln bestehender Lösungen
 - Dem Wunsch oder der Notwendigkeit nach Einsatz neuer Technologien
- Bewusste Anforderungen haben offensichtliche Gründe
- Leicht zu formulieren

- Selbst offensichtliche Anforderungen können übersehen werden
- Projekt-beteiligte können aus dem Umgang mit bereits existierendem System an die Erfüllung bestimmter Anforderungen gewöhnt sein
- Treten zutage, wenn eine „selbstverständliche“ Funktion fehlt
- Beruhen häufig auf der (falschen) Annahme, ein neues System werde alle positiven Attribute seines Vorgängers oder der Konkurrenz übernehmen

- Anforderungen, deren Umsetzung den Projektbeteiligten gar nicht in den Sinn kommt
- Anforderungen von denen angenommen wird, dass sie sowieso nicht umgesetzt werden können
- Ungeahnte Anforderungen treten sehr häufig im Projektverlauf zu Tage, wenn die Projektbeteiligten neue Technologien und Möglichkeiten verstehen

Quellen

- Gute Anforderungen benötigen gute Quellen
- Finden der Quellen gehört zur Anforderungsanalyse
- **Anwender**
 - Primäre Quelle für Anforderungen
 - Kennen Bedürfnisse und Abläufe
 - Interviews

- Beobachtung der Anwender bei der Arbeit mit der aktuellen Lösung
- Übernahme der Arbeit der Anwender
 - Erkennung der Probleme mit der alten Lösung
 - Erkennung der bewährten Abläufe

- Vielfach gibt es alte Lösungen oder ähnliche Systeme anderer Hersteller
- Aus diesen kann gelernt werden
- Bewährte Teillösungen können übernommen werden

- Features ähnlicher Anwendungen können ebenfalls Diskussionsgrundlage für die Anforderungen sein

- Große Unternehmen besitzen einen Benutzerservice, der Probleme löst und protokolliert
- Schulungsmitarbeiter sind mit den Problemen der Anwender vertraut
- Gute Quelle

- Zweckentfremdung von Anwendungen durch den Benutzer häufig
- Motivation erfragen

Workshops

29

- Ermittlung von Anforderungen in zwei bis fünf Tagen und Abschätzung der Kosten
- Schneller und besser, wenn die richtigen Personen zusammen kommen
- Wegen der vielen Teilnehmer erscheint ein Workshop teuer, doch er spart Zeit

Anforderungs- dokument

- Dient der Dokumentation und Organisation von Anforderungen
- Unterliegt einer logischen Struktur
- Aktivitäten als Hauptabschnitte in natürlicher Reihenfolge
- Signifikante Ausnahmen werden an der Stelle, an der sie auftreten, dokumentiert
- Verwandte Aktivitäten werden logisch gruppiert

Beispiel

32

- 1 Projektziel
- 2 Übersicht der Anwendungsfälle
- 3 Anwendungsfälle im Detail
 - 3.1 Registrierung
 - 3.1.1 Eingabe E-Mail und Passwort
 - 3.1.1.1 Kein Passwort angegeben
 - 3.1.1.2 Doppelt Passwort-Eingabe stimmt nicht überein
 - 3.1.2 Eingabe der persönlichen Daten
 - 3.1.2.1 Pflichtfelder nicht ausgefüllt
 - 3.1.3 Auswahl der Optionen
 - 3.1.4 Zustimmung zu den Nutzungsbedingungen
 - 3.1.4.1 Zustimmung nicht gegeben
 - 3.2 Persönliche Einstellungen
 - 3.2.1 Pflege der persönlichen Daten
 - 3.2.2 Pflege der Optionen
 - 3.3 Administration
 - 3.3.1 Benutzer aktivieren/deaktivieren
 - 3.3.2 Benutzerdaten ändern
 - 3.3.3 Benutzerpasswort setzen
- 4 Nicht-funktionale Anforderungen
 - 4.1 Sicherheit
 - 4.2 Zuverlässigkeit
 - 4.3 Verfügbarkeit
 - 4.4 Zeitverhalten
 - 4.5 Verbrauchsverhalten
 - 4.6 Portabilität
 - 4.7 Benutzbarkeit und Ergonomie
 - 4.8 Rahmenbedingungen
 - 4.9 Entwicklungs- und Produktionsanforderungen
- 5 Verzeichnisse und Informationen
 - 5.1 Benutzertypen / Aktoren
 - 5.2 Systemumfeld
 - 5.3 Glossar

- **Guide to the Software Engineering Body of Knowledge – SWEBOK**

Chapter 2 – Software Requirements

<http://www.swebok.org/ch2.html>

- **Frank Müller – Analyse und Verwaltung von Anforderungen**

http://frank.mweb.de/wiki/Analyse_und_Verwaltung_von_Anforderungen

- **Stefan Blaubach – Anforderungsmanagement in IT-Projekten**

www.modulo3.de/vortraege/lt2002_vortrag_anforderungsmanagement_in_it_projekten.pdf

- **TU München – Fakultät für Informatik – Lehrstuhl IV: Software & Systems Engineering**

<http://www4.informatik.tu-muenchen.de/research/requirements/index.shtml>