

# Grundlagen der Anforderungsanalyse

28. Oktober 2014

---

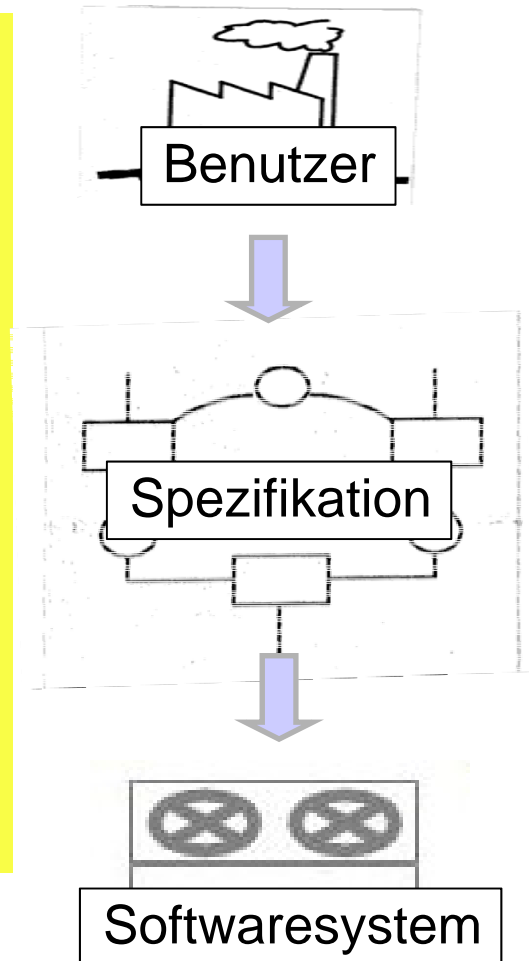
# Überblick

- Wie analysiert man die Anforderungen an ein neues Softwaresystem?
  - *Welche Methoden und Techniken gibt es?*
- Welche Probleme kann es bei der Anforderungserfassung geben?
- Wie werden Anforderungen systematisch beschrieben?
  - *Analyse der funktionalen Einheiten*
  - *strukturierte Anforderungsbeschreibung*
  - *Formen der Beschreibung:*
    - Lastenheft (Auftraggeber)
    - Pflichtenheft (Entwickler)

# Frühe Phasen der SW-Entwicklung: Zielsetzungen

Ziel der sog. *frühen Phasen der SW-Entwicklung* ist es,

- a) einen *Ausschnitt* der realen Welt als *Gegenstandsbereich* einer SW-Entwicklung zu *identifizieren* und *abzugrenzen*,
- b) für diesen Ausschnitt durch Abstraktion eine *Anforderungsspezifikation* zu *entwickeln* und darzustellen, und
- c) diese Spezifikation als *Grundlage* für den anschließenden *Softwareentwurf* und die *Realisierung* des SW-Systems verwenden zu können.



# Methoden der Anforderungserhebung

- Interviews
  - *Mitarbeiter werden durch Person befragt*
  - *Auswahl verschiedener Mitarbeiter mit verschiedenen Aufgaben*
- Fragebögen
  - *Mitarbeiter bearbeitet Bogen (Eindeutigkeit, Vollständigkeit, Kontrollfragen, Bearbeitungszeit)*
- Beobachtung
  - *Mitarbeiter wird beobachtet (offen, verdeckt (Achtung, ggf. illegal), quantitative/qualitative Erfassung)*
- Berichtsmethode
  - *Mitarbeiter erläutert schriftlich eine Thematik (Formfreiheit vs. Vorgabe)*
  - *Lastenheft*
- Inventurmethode
  - *Studium schriftlicher Unterlagen (abhängig von Qualität, Vollständigkeit, Aktualität,...)*
- Brainstorming
  - *neue Ideen zu bestehenden Arbeitsprozessen*

# Lastenheft

Der Auftraggeber beschreibt alle Anforderungen an das Softwaresystem in einem Dokument.

- Ist-Zustand: Worauf soll das SW-Projekt aufsetzen?
- Soll-Zustand: Welches Softwaresystem wird gewünscht?
- Wer ist im Projekt für welche Arbeiten zuständig?
- Was soll das Softwaresystem können?
- Welche weiteren Anforderungen sind wichtig? (Z.B.: Zuverlässigkeit, Wartbarkeit und Benutzbarkeit)

# Typische Probleme bei der Systemanalyse

- Kommunikationsprobleme, Sprachbarrieren
- widersprüchliche Sichten auf den Ist-Zustand
- unklare Zielvorstellungen vom System
- hohe Komplexität der zu lösenden Aufgaben
- sich ständig ändernde Ziele u. Anforderungen
- schlechte Qualität der Anforderungsbeschreibung
  - *mehrdeutig, redundant, widersprüchlich, ungenau...*
  - *unnötige Produktmerkmale*

# Festlegung der Anforderungen: Schritte

Im einzelnen lässt sich der Prozess der Anforderungsfestlegung in folgende Teilschritte aufgliedern:

**Anforderungsermittlung:** Die Anforderungen werden (z.B. durch Anwendergespräche, Lastenheft) gesammelt und ausgewertet.

**Anforderungsanalyse:** Die gesammelten Anforderungen werden klassifiziert, bewertet (z.B. unter Kosten-/Nutzen-Aspekten), verglichen und geprüft (z.B. auf Konsistenz und Vollständigkeit)

**Anforderungsbeschreibung:** Die gesammelten und geprüften Anforderungen werden in einheitlicher Form (z.B. als Anwendungsfälle) beschrieben und in einem Dokument, dem Pflichtenheft, niedergelegt.

**Anforderungsrevision:** Die Anforderungen werden - falls notwendig und gefordert - zu einem späteren Zeitpunkt neu überprüft und ggf. geändert. Dazu bedarf es eines formellen Änderungsverfahrens - die Folge sind mögliche neue Projekt- oder Entwicklungszyklen.

# Inhalt der Anforderungsbeschreibung

- Zielsetzung
- allgemeine Beschreibung
- Definitionen und Abkürzungen
- Produktumfeld
- funktionale Anforderungen
- nicht-funktionale Anforderungen
- Abnahmekriterien
- Glossar, Index, Referenzen

IEEE 830-98  
Standard für  
Anforderungs-  
dokumente



# Klassifizierung von Anforderungen

Anforderungen an ein Anwendungssystem lassen sich grundsätzlich in zwei Kategorien einteilen:

- **funktionale** Anforderungen und
- **nicht-funktionale** Anforderungen

- **Funktionale** Anforderungen beziehen sich darauf, **was** das System leisten soll.

Sie werden z.B. gegliedert nach dem Schema:

- . Funktionskomplexe und einzelne Funktionen,
- . die dazu benötigten **E**ingaben,
- . evtl. dazu zwingend erforderliche **V**erarbeitungsschritte,
- . ihre erwarteten **A**usgaben.

(**EVA-Prinzip**)

# Funktionale Einheit: Anwendungsfall

## *Grundidee:*

Die Funktionalität des Anwendungssystems in logisch zusammengehörige und handliche funktionale Einheiten (die "Anwendungs-" oder "Nutzungsfälle") gliedern und diese in standardisierter Form beschreiben.

Ein **Anwendungsfall** ist eine abgeschlossene, zusammenhängende Einheit, welche einen Teil der Funktionalität des Systems repräsentiert.

## Beschreibung eines Anwendungsfalls als Erweiterung des EVA-Prinzips:

- *Titel:*
- *Kurzbeschreibung:*
- *Aktoren:*
- *Vorbedingungen (Eingaben):*
- *Beschreibung des Ablaufs (Verlauf):*
- *Auswirkungen (Ausgaben):*
- *Anmerkungen:*

# Aktoren

**Aktor:** Objekt der Systemumgebung, das mit dem System interagiert (und einen oder mehrere Anwendungsfälle auslösen kann).  
Aktoren können Personen (System-Nutzer), externe Geräte oder mit dem System verbundene Nachbarsysteme sein.

Aktoren tauschen mit dem System Nachrichten aus und können als Sender und/oder Empfänger von Nachrichten auftreten.

Aktoren sind i.a. selbst nicht Bestandteile des Systems. Oft müssen jedoch Daten über sie (z.B. zur Regelung der Zugangsberechtigung) mit verwaltet werden.

# Beispiel: Anwendungsfall

- *Titel: Vorlesung eintragen*
- *Kurzbeschreibung: Dozent gibt Raum, Zeit und Titel einer Vorlesung ein und legt einen Vorlesungseintrag an.*
- *Aktor: Dozent*
- *Vorbedingungen: Eine Vorlesung mit diesem Titel gibt es noch nicht.*
- *Beschreibung des Ablaufs:*
  - *Prüfe, ob der Raum zur angegebenen Zeit schon vergeben ist*
  - *Wenn nicht, wird die neue Vorlesung eingetragen und die Daten der Vorlesung werden angezeigt.*
  - *Falls vergeben, wird die Vorlesung nicht eingetragen und eine entsprechende Fehlermeldung wird angezeigt.*
- *Auswirkungen: Die Vorlesung ist gespeichert oder ein Fehler wird gemeldet.*
- *Anmerkungen:*

# Funktionale Anforderungen: Weitere Beispiele

- Aktionen, die vom System ausgeführt werden sollen
  - *Bsp.: Das System muss Ausleihgegenstände in den Bestand aufnehmen können.*
- Systeminteraktionen, die dem Nutzer ermöglicht werden
  - *Bsp.: Das System muss es dem Administrator bei der Aufnahme eines Ausleihgegenstandes in den Bestand ermöglichen, den Autor, den Titel und die ISBN einzugeben.*
- allg. funkt. Vereinbarungen u. Einschränkungen
  - *Bsp.: Der Client ist für den Kommunikationsaufbau zuständig.*

# Nichtfunktionale Anforderungen

- **Nicht-funktionale** Anforderungen beziehen sich darauf, **wie** das System seine Leistungen erbringen soll.

Dazu gehören:

- **Qualitäts**anforderungen an das System oder an einzelne Komponenten wie z.B.:
  - . Performanz (Antwortzeiten, Speicherkapazität),
  - . Zuverlässigkeit,
  - . Änderbarkeit und Wartbarkeit
  - . Portabilität;
- Anforderungen an die **Benutzbarkeit** des Systems wie z.B.:
  - . ergonomische Gestaltung der Benutzungsschnittstelle,
  - . Angemessenheit der Aufgaben,
  - . Erlernbarkeit,
  - . Robustheit bzgl. Fehlern;

# Nicht-funktionale Anforderungen (Forts.)

- Anforderungen an die **Realisierung** des Systems, bestehend aus
  - Entwicklungsumgebung,
  - Zielumgebung (Hardware/Software-Plattform, sonstigen Geräten und Schnittstellen),
  - Methoden, Werkzeugen und Vorgehensweisen;
- Anforderungen an die **Einführung, Nutzung** und **Betreuung** des Systems. Dazu gehören
  - Übergangsmöglichkeiten,
  - Versionsführung,
  - Änderungs- und Weiterentwicklungsverfahren,
  - benötigte Ressourcen,
  - zu beachtende Vorschriften, Prüfverfahren etc.
  - Benutzungsdokumentation und Handbücher.

# Nicht-funktionale Anforderungen: Beispiele

- technische Anforderungen
  - *Bsp.: Das System muss mit Java entwickelt werden.*
- ergonomische Anforderungen
  - *Bsp.: Das System muss die gespeicherten Objekte formatiert ausgeben können.*
  - *Bsp.: Daten sollen über eine graphische Oberfläche ein- und ausgegeben werden.*
- Anforderungen an die Dienstqualität
  - *Bsp.: Das System muss jede Anfrage des Benutzers innerhalb von 30 Sekunden ausführen.*



# Nicht-funktionale Anforderungen: Beispiele (2)

- Anforderungen an den Entwicklungsprozess
  - *Bsp.: Der Entwickler muss mit dem Auftraggeber monatliche Reviews der zu erstellenden Dokumente durchführen.*
- rechtlich-vertragliche Anforderungen
  - *Bsp.: Der Auftraggeber leistet für jeden abgenommenen Meilenstein ein Drittel der vertraglich vereinbarten Summe für die Entwicklung des Systems.*

# Qualitätskriterien für Anforderungsbeschreibungen

- vollständig
- korrekt
- rechtlich klar
- konsistent
- testbar
- aktuell
- verständlich
- realisierbar
- notwendig
- bewertbar
- eindeutig

Bsp.: Das System soll schnell reagieren.  
Worauf? Wie schnell?  
Wie reagieren?

# Wie schreibt man richtige Anforderungen?

## Keine unvollständigen Anforderungen:

- unvollständige Prozesse und Vergleiche:
  - *Bsp.: Die Auslastung der Systemressourcen soll überwachbar sein.*  
*Wer überwacht? Was und wie wird überwacht?*  
*Wer oder was ist ausgelastet? Durch was ist er/sie/es ausgelastet?*
- implizite Annahmen:
  - *Bsp.: Der Typ des Leihobjekts wird im Eingabeformular eingegeben.*  
*Es gibt einen Typ des Leihobjekts. Es gibt ein Eingabeformular.*

# Wie schreibt man richtige Anforderungen?

## Keine unzulässigen Verallgemeinerungen:

- unzulässige Universalquantoren:
  - *Bsp.: Das System soll eine durch den Benutzer eingeleitete Sicherung von aufgezeichneten Daten auf Band ermöglichen.*  
*Jeder Benutzer? Alle aufgezeichneten Daten? Kann die Sicherung immer geschehen? Auch parallel?*
- unvollständig spezifizierte Bedingungen:
  - *Bsp.: Das System soll bei Störung gestoppt werden.*  
*Wann liegt eine Störung vor?*

# Wie schreibt man richtige Anforderungen?

Keine vagen Beschreibungen:

- Nominalisierte Prozesse:

- *Bsp.: Es sollen Datenverluste erkannt werden.*

*Welche Daten gehen verloren? Wie wird der Verlust der Daten erkannt?*

- Funktionsverbgefüge:

- *Bsp.: Das System muss die Berechnung der Statistiken innerhalb einer Stunde zu Ende bringen.*

*Wird die Berechnung abgebrochen oder muss sie innerhalb einer Stunde vollständig erledigt sein?*

# Zusammenfassung

- Die Anforderungsanalyse ist zentral für die erfolgreiche Entwicklung von Softwaresystemen.
  - *Der Auftraggeber kann diese mit einem Lastenheft unterstützen.*
- Auftraggeber und Entwickler müssen sich auf eine Anforderungsbeschreibung einigen. Hier treten viele Probleme auf.
  - *Das Ergebnis ist ein Pflichtenheft.*
- Es werden funktionale und nichtfunktionale Anforderungen an eine Anwendung unterschieden.