Grundlagen Software Engineering

Requirements Engineering

Inhalt

- *Was soll gemacht werden?* Ermittlung und Dokumentation von Anforderungen; kurze Wiederholung:
  - Lastenheft,
  - Pflichtenheft,
  - SA,
  - SA/RT,
  - UML
- *Wie viel Aufwand wird dafür benötigt?* **Aufwandsschätzung**
- *Wie erfolgt die Entwicklung konkret?* Projektplanung => SE 2
- *Wie stellen wir sicher, das die Software später auch benutzbar ist?* **Software-Ergonomie**

Lernziele

- Lasten-, Pflichtenheft erklären können
- Bedeutung von Anforderungen benennen können
- Ablauf Requirements Engineering verstehen
- Techniken zur Anforderungserhebung kennen
- UML in der Analyse anwenden können

Überblick

Aufgaben zu Beginn einer Softwareentwicklungsprojekts

- Erstellung des ***Lastenhefts*** (und ggf. eines ***Glossars***) als erste grobe Zusammenfassung der fachlichen Anforderungen an das zu entwickelnde Software-Produkt => in SE 2 bereits eingeführt
- ***Aufwandsschätzung*** => hier
- Durchführung von Machbarkeits- und Rentabilitätsuntersuchungen => Spezialthema; hier nicht
- Ggf. Erstellung eines ***Projektplans*** => in SE 2 bereits eingeführt
- Typische Ergebnisse der Planungsphase: **Lastenheft** und **Glossar**, **Aufwandsschätzung**, Ergebnisse der Machbarkeits- und Rentabilitätsuntersuchungen, **Projektplan**
- Ergänzt in der Analysephase durch das **Pflichtenheft** => in SE 2 bereits eingeführt

Was soll gemacht werden?
Wiederholung: Lastenheft

- Alternative Bezeichnung: Grobes Pflichtenheft
- Funktion
  - Knappe Beschreibung aller fachlichen Basisanforderungen aus Sicht des Auftraggebers (Beschreibung des "Was")
  - Erste, grobe Beschreibung der Anforderungen an ein neues Produkt
- Form
  - Festes Gliederungsschema
- Ermittlung der Lastenheftinhalte
  - Diverse Verfahren; besonders systematisch ist *Quality Function Deployment* (QFD)

Was soll gemacht werden?
Wiederholung: Lastenheft

- Gliederungsschema eines Lastenheftes
  - Zielbestimmung
  - Produkteinsatz
  - Produktübersicht
  - Produktfunktionen
  - Produktdaten
  - Produktleistungen
  - Qualitätsanforderungen
  - Ergänzungen

Was soll gemacht werden?
Wiederholung: Glossar

- Aufgaben des Glossars
  - Ergänzt das Lastenheft
  - Definiert und erläutert Begriffe, um eine einheitliche Terminologie sicherzustellen
  - Beispiel
    - *Kundensachbearbeiter*
      - *Verantwortlich für die Kommunikation mit →Kunden und →Firmen einschließlich der Auskunftserteilung und Buchung*
  - Wichtig ist, dass die in der jeweiligen Branche üblichen Begriffe verwendet werden, die insbesondere auch für den Produkt-Benutzer verständlich sind
  - Die Glossarbegriffe werden sowohl für die Benutzungsoberfläche als auch für die Online-Hilfe und das Benutzerhandbuch verwendet

Was soll gemacht werden?
Wiederholung: Pflichtenheft

- Funktion
  - Das Pflichtenheft enthält eine Zusammenfassung aller fachlichen Anforderungen, die das zu entwickelnde Software-Produkt aus der Sicht des Auftraggebers erfüllen muss. Außerdem werden Entwicklungsprioritäten aus Auftraggebersicht festgelegt
- Form und Inhalt
  - Die Inhalte stellen eine Konkretisierung und Detaillierung der Lastenheft-Inhalte dar. Das Lastenheft kann daher als Ausgangsdokument für das Pflichtenheft verwendet werden
  - Die Grundgliederung von Lasten- und Pflichtenheft sind identisch
  - Beschreibung des "Was", nicht des "Wie"
  - Das Pflichtenheft muss so abgefasst sein, dass es als Basis eines juristischen Vertrages dienen kann. Das Pflichtenheft stellt also die vertragliche Beschreibung des Lieferumfangs dar
  - Anhand des Pflichtenheftes soll das fertige Produkt abgenommen werden können.
  - Die beschriebenen Anforderungen müssen realisierbar sein
  - Entwurfs- und Implementierungsentscheidungen sollen nicht vorweggenommen oder unnötig eingeschränkt werden

Was soll gemacht werden?
Wiederholung: Pflichtenheft

- Gliederungsschema eines Pflichtenheftes
  - Zielbestimmung
    - Musskriterien: Unbedingt notwendige Anforderungen
    - Wunschkriterien: Möglichst - aber nicht zwingend - zu berücksichtigende Wünsche
    - Abgrenzungskriterien: Explizit nicht zu erreichende Ziele
  - Produkteinsatz
    - Anwendungsbereiche: z.B. Textverarbeitung im Büro
    - Zielgruppen: z.B. Sekretärinnen, Schreibkräfte
    - Betriebsbedingungen: z.B. Büroumgebung, Produktionsanlage, mobiler Einsatz, tägliche Betriebszeit: z.B. Dauerbetrieb bei Telekommunikationsanlagen, ständige Beobachtung des Systems durch Bediener oder unbeaufsichtigter Betrieb
  - Produktübersicht
    Gibt eine Übersicht über das Produkt, z.B. über alle wichtigen Geschäftsprozesse

Was soll gemacht werden?
Wiederholung: Pflichtenheft

- Gliederungsschema eines Pflichtenheftes (Fortsetzung)
  - Produktfunktionen
    - Konkretisierung und Detaillierung der Lastenheft-Funktionen (mit Querverweis auf die dortige Numerierung)
  - Produktdaten
    - Die persistenten Daten sind aus Benutzersicht detaillierter zu beschreiben.
    - Im einfachsten Fall verbale Beschreibung, ggf. auch formal (Data Dictionary: EBNF; OOA: Attribut-Spezifikation im Klassen-Diagramm)
  - Produktleistungen
    - Werden an einzelne Funktionen und Daten Leistungsanforderungen bzgl. Zeit, Genauigkeit oder Mengen gestellt, dann werden sie hier aufgeführt und mit /Lnn/ markiert
  - Qualitätsanforderungen
    - In diesem Kapitel wird festgelegt, welche Qualitätsmerkmale das zu entwickelnde Produkt in welcher Qualitätsstufe besitzen soll
    - Voraussetzung für die Qualitäts-Zielbestimmung ist, dass die Qualitätsmerkmale in operationalisierter Form vorliegen

Was soll gemacht werden?
Wiederholung: Pflichtenheft

- Gliederungsschema eines Pflichtenheftes (Fortsetzung)
  - Benutzungsoberfläche
    - In diesem Kapitel werden grundlegende Anforderungen an die Benutzungs-oberfläche festgelegt, z.B. Fensterlayout, Dialogstruktur und Mausbedienung entsprechend dem Windows-Gestaltungs-Regelwerk (style guide) oder unternehmenseigenen Gestaltungs-Regelwerken
    - Die einzelnen Anforderungen werden analog wie die Funktionsanforderungen numeriert: /B nn/
  - Nichtfunktionale Anforderungen
    - Es werden alle Anforderungen aufgeführt, die sich nicht auf die Funktionalität, die Leistung und die Benutzungsoberfläche beziehen, z.B. einzuhaltende Gesetze oder Normen
  - Technische Produktumgebung
    - Software
    - Hardware
    - Orgware
    - Produkt-Schnittstellen

Was soll gemacht werden?
Wiederholung: Pflichtenheft

- Gliederungsschema eines Pflichtenheftes (Fortsetzung)
  - Spezielle Anforderungen an die Entwicklungs-Umgebung
    - Software
    - Hardware
    - Orgware
    - Entwicklungs-Schnittstellen
  - Gliederung in Teilprodukte
    - Das Produkt wird in Teilprodukte aufgeteilt, die getrennt aus Sicht des Auftraggebers entwickelt werden sollen. Die Funktionalität wird den einzelnen Teilprodukten zugeordnet (mehrere aufeinanderfolgende Versionen mit ansteigender Funktionalität)
  - Ergänzungen

Was soll gemacht werden?
Beispiel: Lastenheft und Pflichtenheft

- Lasten- und Pflichtenheft eines Getränkeautomaten

Lastenheft
Getränkeautomat

1. Änderungshistorie

KM-Version	Ausgabe-datum	Name des Autors	geänderte Abschnitte	Änderungsgrund, ggf. Änderungsmitteilung
0.1	12.03.2001	Gericke	Alle	Neuerstellung
1.0	12.03.2001	Gericke	4,6	Review-Anderungen

2. Zielbestimmung

Die Firma Durst-O-Matic möchte eine neue Serie von freundlichen Getränkeautomaten auf dem Markt etablieren. Diese Automaten sollen zuverlässig arbeiten und eine intuitive Bedienung erlauben. Als Marktentscheidender Vorteil sollen diese Automaten eine hohe Verfügbarkeit besitzen und wenig Wartung erfordern. Die interne Steuerung soll durch Software realisiert werden.

3. Produkteinsatz

Die Automaten kommen an unterschiedlichsten Orten und Ländern zum Einsatz.

Lastenheft
Getränkeautomat

4. Produktfunktionen

/LF10/	Die Art des Getränkes (Kaffee, Limonade u.ä.) kann über eine Tastatur gewählt werden.
/LF20/	Die Art des Getränke-Zusatzes (Zucker, Milch u.ä.) kann über eine Tastatur gewählt werden.
/LF30/	Die Getränke werden in einen Becher abgefüllt.
/LF40/	Die Becherausgabe kann unterdrückt werden, um eigene Tassen benutzen zu können.
/LF50/	Der Automat signalisiert leere Zustände.
/LF60/	Der Automat ermöglicht die Bezahlung mit Münzen.
/LF70/	Der Automat wechselt bei Überbezahlung.
/LF80/	Die Belegung der einzelnen Wahl-Tasten kann frei vorgenommen werden.
/LF90/	Der Automat signalisiert das Ende der Abfüllung.
/LF100/	Der Automat muss komfortabel an die jeweiligen länderspezifischen Gegebenheiten (z.B. Währung) anpassbar sein.

5. Produktdaten

/LD10/	Die Abgabemengen pro Tag sind zu speichern.
/LD20/	Ausfälle sind zu protokollieren.

Lastenheft
Getränkeautomat

6. Produktleistungen

/LL10/	Die Rückgabe des Geldes darf erst nach Beendigung der Abfüllung erfolgen.
/LL20/	Die Abfüllung muß spätestens ca. 4 Sekunden nach Betätigung der Auswahl erfolgen. Bei warmen Getränken muß spätestens ca. 20 Sekunden nach Betätigung der Auswahl die Abfüllung erfolgen.
/LL30/	Der aktuelle Betriebszustand muß dem Benutzer signalisiert werden.
/LL40/	Es müssen mindestens 10000 Getränke pro Tag ausgegeben werden können.
/LL50/	Die Korrektheit der Münzen ist zu prüfen.
/LL60/	Es müssen mindestens 8 verschiedene Getränke zur Verfügung gestellt werden.
/LL70/	Es müssen mindestens 3 verschiedene Getränkezusätze zur Verfügung gestellt werden.

7. Qualitätsanforderungen

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität		X		
Zuverlässigkeit	X			
Benutzbarkeit	X			
Effizienz			X	
Änderbarkeit		X		
Robustheit	X			
Wartung	X			
Ergonomie	X			

1. Ziel des Dokuments

In Zusammenarbeit mit der Firma *Durst-O-Matic* soll eine neue Generation eines bedienerfreundlichen Getränkeautomaten entwickelt werden (*Durst-O-Mat1*). Auf Basis der Auftraggeberanforderungen, wie sie im Lastenheft definiert sind, erfolgt hier die Spezifizierung der fachlichen Anforderungen im Rahmen eines Pflichtenhefts.

Die Systematik des Pflichtenhefts basiert auf der in /Balzert96/ [1] vorgestellten Methode. Dabei wurde besonders auf eine verständliche Darstellung zu Lehrzwecken geachtet.

Zur weiterführenden Literatur wird /Willmer84/ [2] empfohlen. Dort ist anhand eines Praxisprojekts der systematische Pflichtenheftaufbau beschrieben.

[1] Balzert, Helmut: Lehrbuch der Softwaretechnik: Software-Entwicklung – Heidelberg; Berlin: Spektrum 1996

[2] Willmer, Heidemarie: Fallstudie einer industriellen Softwareentwicklung – Mannheim; Wien; Zürich: Bibliographisches Institut, 1984

1.1. Änderungshistorie

KM-Version	Ausgabe-datum	Name des Autors	geänderte Abschnitte	Änderungsgrund, ggf. Änderungsmitteilung
1.0	16.03.2001	Neumann	alle	Neuerstellung
1.1	22.03.01	Neumann	2.3., 4.2.	Detaillierung der Hardwareanforderungen
1.2	09.05.2001	Neumann	7, 8, 10	Testergebnisse und neue Leistung
1.3	10.05.2001	Neumann	1, 5 – 10	Detaillierung und Lösung von Inkonsistenzen; /L40/ geändert
1.31	11.05.2001	Neumann	6.3	Ergänzung der Referenzdaten
1.4	15.05.2001	Neumann	/F50/, 10	präzisierter Bezahlvorgang

2. Zielbestimmung

-Ziele des Produkteinsatzes, legt Entscheidungsraum für Realisierung fest-

Durch das Produkt will die Firma Durst-O-Matic ihre Markposition im Bereich der bedienfreundlichen Getränkeautomaten ausbauen. Dazu ist neben der zuverlässigen Getränkeausgabe ein Augenmerk auf gute Bedien- und Wartbarkeit notwendig

2.1 Musskriterien

- Getränkeausgabe nach Wahl mit verschiedenen Zusätzen
- Landesspezifisch konfigurierbar
- Protokoll aller Vorgänge
- Bedienfreundlich
- Zuverlässig
- sicher
- Hochverfügbar
- Softwaregesteuert

2.2 Wunschkriterien

- ‚freundliche' Ausgabe

2.3 Abgrenzungskriterien

- Keine externe Kommunikationsanbindung, Daten werden über lokale Schnittstelle abgefragt
- Keine Sprachausgabe, nur Tastenkommunikation und Rückmeldung durch Lampen für Bediener
- Der Getränkepreis ist immer ein ganzzahliges Vielfaches der kleinsten Münzsorte

3. Produkteinsatz

-multikriterielle Abgrenzung der Einsatzbedingungen-

Die Automaten kommen an unterschiedlichsten Orten und Ländern zum Einsatz.

3.1 Anwendungsbereiche

Der Durst-O-Mat1 dient ausschließlich zur Getränkeausgabe bei Bezahlung durch landesspezifische Münzen.

3.2 Zielgruppe

Alle Personengruppen und –bereiche, zur Wartung technisch geschultes Personal

3.3 Betriebsbedingungen

Aufstellung in geschlossenen Räumen mit Büroumgebung

4. Produktumgebung

Der Durst-O-Mat1arbeitet als eigenständige Applikation ohne externe Verbindungen (ausgenommen Energie- und ggf. Wasserversorgung).

4.1 Software

Die besonders gute Bedienbarkeit soll durch Anwendung der gängigen Normen und Richtlinien gewährleistet werden.

4.2 Hardware

Das optisch ansprechende Äußere des Getränkeautomaten soll Kunden anziehen. Wegen der ständigen Betriebsbereitschaft ergibt sich die Forderung nach einem niedrigen Energieverbrach. Durch die Aufstellung in öffentlich zugänglichen Gebäuden sind die Sicherheitsbestimmungen zu beachten, weiterhin muß der Automat stabil, sicher und vandalismusgeschützt konstruiert sein.

4.3 Orgware

keine Netzwerkverbindungen –

4.4 Produkt – Schnittstellen

Um die Konfigurierbarkeit zu gewährleisten, muß die Möglichkeit der Dateneingabe und –anzeige direkt am Gerät (über Schnittstelle nach Absprache) bestehen. Das Protokoll aller Vorgänge muß zur weiteren Auswertung über eine serielle Schnittstelle abrufbar sein.

5. Produktfunktionen

-Funktionale Beschreibung des Produktes aus Benutzersicht –

Die Funktionalität des Getränkeautomaten ist in die Bereiche Getränkeausgabe, Bezahlung, Konfigurierung und Protokollierung und Anzeige unterteilt.

5.1 Grundfunktion Getränkeausgabe

/F10/	Auswahl der Getränkeart über Tastendruck
/F20/	Auswahl der Getränke-Zusatzart über Tastendruck
/F30/	Becherausgabe kann durch Tastendruck unterdrückt werden
/F40/	Nach Auswahl wird ggf. Becher ausgegeben und anschließend das Getränk

5.2 Grundfunktion Bezahlung

/F50/	Der Automat ermöglicht die Bezahlung mit 5 Münzsorten
/F60/	Die Korrektheit der Münzen ist zu prüfen.
/F65/	Ungültige Münzen sind sofort auszugeben
/F70/	Der Automat wechselt bei Überbezahlung.
/FW80/	Eine Abbruchtaste gibt das eingeworfene Geld des Vorgangs zurück und unterbricht den Ausgabevorgang
/F90/	Der Abbruch kann nur vor der Getränkeausgabe erfolgen
/F100/	Der Getränkepreis ist immer ein ganzzahliges Vielfaches der kleinsten Münzsorte
/F110/	Rückgabe der Überzahlung in größtmöglichen Münzsorten.
/F120/	Korrekte Bezahlung: Rückgabe = Einwurf – Preis

5.3 Grundfunktion Konfigurierung

/F130/	freie Zuordnung der Getränke- und Zusatztasten zu Produkten
/FW140/	Elektronische Anzeige der Getränkezuordnung
/F150/	Ausgabe und Anzeige aller Texte in Landessprache
/F160/	Betragsanzeige in Landeswährung
/F165/	Eingebbare Münzsorten können frei konfiguriert werden

5.4 Grundfunktion Protokollierung

/F170/	Alle in 6.1. geforderten Daten sind zu protokollieren
/F180/	Der Automat bietet eine Zeit- und Datumsbasis

6. Produktdaten

- Beschreibung der Daten aus Benutzersicht –

Gespeicherte Daten dienen ausschließlich zur Protokollierung und Verarbeitung interner Zustände.

6.1 Protokolldaten

/D10/	Abgabemenge pro Tag für jedes Getränk.
/DW20/	Einzelmenge pro Verkauf mit Zeit
/D25/	Getränkezusatz pro Tag für jeden Zusatz
/DW30/	Becher mit ausgegeben
/D40/	Ausfall mit Zeit, Datum, Bedienvorgang und Getränkestand

6.2 Zustandsdaten

/D50/	Füllstände der Getränke
/D60/	Becheranzahl
/D70/	Füllstände der Zusätze
/D80/	Anzahl der einzelnen Münzsorten
/D90/	Aktuell eingeworfener Betrag
/D100/	Aktuell gewähltes Getränk

6.3 Referenzdaten

/D110/	Getränkepreise
/D120/	Preise der Zusätze
/D130/	Zuordnung der Tasten zu Getränken
/D140/	Landesspezifische Texte für Benutzerkommunikation und gewählte Sprache

7. Produktleistungen

- zeit- oder umfangsbezogene Produktanforderungen -

/L10/	Die Rückgabe des Geldes darf erst nach Beendigung der Abfüllung beginnen.
/L20/	Die Abfüllung der Getränke muss spätestens ca. 4 Sekunden nach Betätigung der Auswahl beginnen.
/L30/	Bei warmen Getränken muss spätestens ca. 20 Sekunden nach Betätigung der Auswahl die Abfüllung beginnen.
/L40/	Es müssen mindestens 500 Getränke pro Tag ausgegeben werden können.
/L50/	Es müssen mindestens 8 verschiedene Getränke zur Verfügung gestellt werden.
/L60/	Es müssen mindestens 3 verschiedene Getränkezusätze zur Verfügung gestellt werden.
/L70/	Es werden mindestens 200 mögliche Überzahlvorgänge pro Tag garantiert
/L75/	Innerhalb eines Bezahlvorgangs müssen mindestens 10 Münzen pro Münzsorte eingebbar sein.
/L80/	Protokolldaten werden mind. 30 Tage gespeichert

8. Bedienoberfläche

/B05/	Anzeige in jeweiliger Landessprache oder entsprechenden Piktogrammen
/B10/	Für den Getränkeverkauf (Benutzer) ist eine Eingabe über Tasten vorgeschrieben
/B15/	Leere Zustände werden angezeigt
/B17/	Die ausgewählten Funktionen werden bis zum Ende des Kaufvorgangs angezeigt
/B20/	Für den Getränkeverkauf (Benutzer) ist die Benutzerrückmeldung über Anzeigelampen vorgeschrieben
/B23/	Bei Auswahltastendruck wird der resultierende Preis angezeigt
/B25/	Der aktuell einzuwerfende Restbetrag bei einem Bezahlvorgang muß angezeigt werden
/B28/	Die in /B25/ geforderte Anzeige umfaßt mindestens 5 Stellen mit variabel konfigurierbaren Dezimalpunkt und ein Vorzeichen
/B29/	Das Abfüllende wird signalisiert.
/B30/	Zusätzliche akustische Rückmeldung
/B40/	Keine Einschränkung der Kommunikation auf Konfigurationsebene
/B50/	Der strukturierte Entwurf ist zu dokumentieren
/B60/	Normen und Richtlinien in diesem Bereich sind zu beachten
/B70/	Dem Benutzer dürfen keine Konfigurationsmöglichkeiten zur Verfügung stehen
/B80/	Das Konfigurationsfeld muß vor dem Benutzer verborgen sein

9. Qualitätsanforderungen

Produktqualität	*Sehr gut*	*Gut*	*Normal*	*Nicht relevant*
Funktionalität				
Angemessenheit		X		
Richtigkeit		X		
Interoperabilität				X
Ordnungsmäßigkeit		X		
Sicherheit		X		
Zuverlässigkeit				
Reife	X			
Fehlertoleranz	X			
Wiederherstellbarkeit		X		
Benutzbarkeit				
Verständlichkeit	X			
Erlernbarkeit	X			
Bedienbarkeit	X			
Effizienz				
Zeitverhalten			X	
Verbrauchsverhalten			X	
Änderbarkeit				
Analysierbarkeit		X		
Modifizierbarkeit		X		
Stabilität	X			
Prüfbarkeit		X		
Robustheit	X			
Wartung	X			

10. Testfälle

-globale Testfälle für den Abnahmetest –

Aus der Anforderungsspezifikation ergibt sich der Bedienvorgang mit verschiedenen Verzweigungen . Daraus sind die Testfälle ableitbar; um die Abdeckung aller Entscheidungen zu erreichen muss jede Entscheidungsmöglichkeit einmal auftreten. Dies ist die Minimalbedingung für die korrekte Funktion.

10.1 Testen der Benutzerrolle

/T10/	Anforderung jedes Getränks mit jedem Zusatz; dabei alles vorhanden
/T20/	Anforderung jedes Getränks mit jedem Zusatz; dabei jeweils ein Produkt leer
/T30/	/T10/ und /T20/ mit und ohne Becher, Becher vorhanden
/T40/	/T10/ und /T20/ mit und ohne Becher, Bechervorrat leer
/T50/	Bezahlung mit ungültigen Münzen
/T60/	Passende Bezahlung
/T70/	Überzahlung, Prüfung auf korrektes Rückgeld
/T80/	/T60/ und /T70/ mit ungenügendem Wechselgeldvorrat
/T90/	Abbruchtastendruck während Getränkeauswahl
/T110/	Abbruchtastendruck während Bezahlung
/T120/	Abbruchtastendruck nach Bezahlung
/T130/	Abbruchtastendruck während Getränkeausgabe
/T140/	Abbruchtastendruck bei Münzrückgabe

10.1 Testen der Benutzerrolle

/T150/	Überprüfung des Protokolls bei /T10/ bis /T140/
/T160/	Neubelegung der Getränketasten und Zusatztasten
/T170/	Preisänderung der Getränke und Zusätze und ein exemplarischer Fall aus /T10/
/T180/	Landestyp- Umstellung (Ausgabe und Münzprüfung) und ein exemplarischer Fall aus /T10/

11. Entwicklungsumgebung

-Konfiguration der Entwicklungsumgebung –

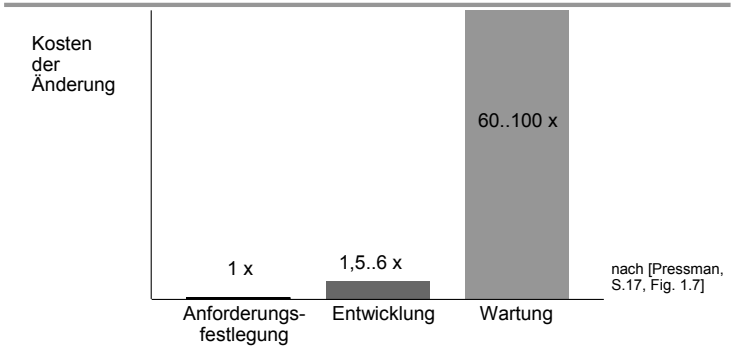
11.1 Software
Der Softwareentwurf ist objektorientiert strukturiert vorzunehmen und zu dokumentieren.
11.2 Hardware
Keine Einschränkungen.
11.3 Orgware
Keine Einschränkungen
11.4 Entwicklungs- Schnittstellen
Keine Einschränkungen

12. Ergänzungen

-Normen, Aufstellungsbedingungen, Personalbedarf, Begriffsdefinition –

13. Anhang

Bedeutung von Anforderungen
Kosten von Änderungen



- ☐ Je später Fehler gefunden werden, um so teuer ist die Behebung!
- ☐ WHISCY Problem: Why in the hell isn't Sam coding yet?

Ausgangssituation: Anforderungen festgelegt

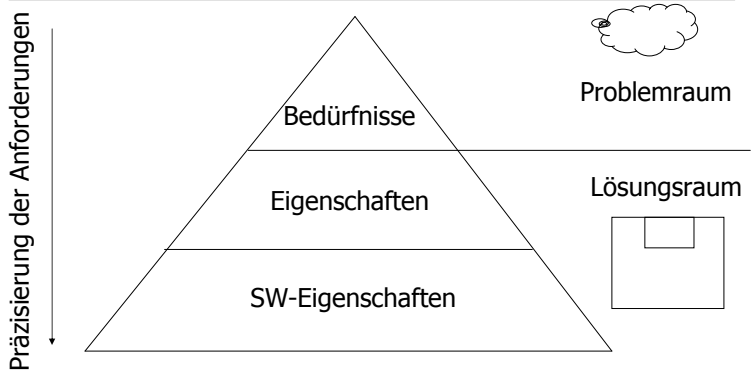
- ☐ Auslöser: Idee, Problem, Geld übrig
- ☐ Interessensgemeinschaft zur Unterstützung
- ☐ Erste Sammlung von Informationen
  - Erste Skizze des zu erstellenden Systems
- ☐ Dann Top-Management-Entscheidung bzgl. Freigabe des Geldes, Go / No-Go Entscheidung, ...

Zielsetzung: Anforderungen festgelegt

- ☐ Von allen Interessenvertretern (Stakeholders) und Projektbeteiligten soll
  - eine gemeinsame Sprache,
  - gegenseitiges Verständnis und
  - von allen akzeptierte Ziele

  definiert werden
- ☐ Verständlich und präzise beschreiben WAS das System leisten soll
- ☐ Umfang der Leistung und Lieferung definieren
- ☐ Bedingungen und Kriterien für die Abnahme festlegen

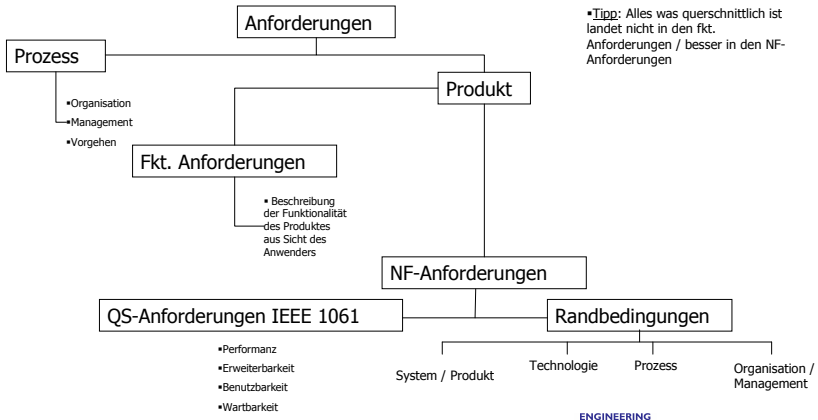
Wie entstehen Anforderungen?



Definition Anforderung

- Eine Eigenschaft oder Bedingung, üblicherweise vom Kunden festgelegt, um ein Problem zu lösen oder ein Ziel zu erreichen.
- Eine Eigenschaft oder Bedingung, die ein System oder eine Systemkomponente erfüllen muss, um einen Vertrag, einen Standard, eine Spezifikation oder andere formal festgelegte Dokumente zu erfüllen.
- Eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung wie in den vorigen Punkten beschrieben
- Beispiel
  - Das System muss dem Administrator die Möglichkeit bieten die Benutzerliste zu drucken.
- Es gibt funktionale und nicht- funktionale Anforderungen

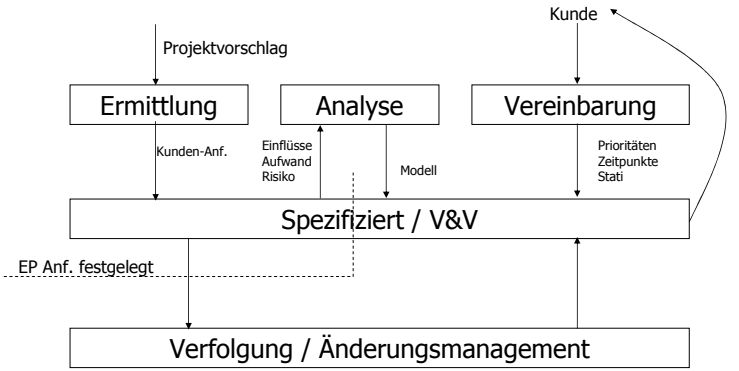
Kategorien von Anforderungen



Qualitätskriterien: Wann ist eine Anforderung gut?

- Anforderungen sollten
  - Verständlich sein (Anwender/Entwickler)
  - Eindeutig sein (nicht interpretierbar/konsistent)
  - Minimal sein
  - Überprüfbar und umsetzbar sein
  - Den Entwurf oder die Implementierung nicht vorwegnehmen
- Bsp.: Das System sollte einen Login-Screen haben.
- Besser: Das System muss einen Mechanismus zur Identifizierung von Anwendern beinhalten.
- Bsp.: Akzeptable Performanz
- Besser: Identifizierung muss innerhalb von 5 Sekunden erfolgen.

Der Anforderungsmanagementprozess



Techniken zur Anforderungserhebung

- ☐ Anforderungsworkshop
- ☐ Exploratives Prototyping
- ☐ Anforderungserfassung mit strukturierten Texten
- ☐ Geschäftsprozessanalyse / Business Process Reengineering
- ☐ Szenarienbasierte Analyse / CRC-Karten

Verfahren und Techniken: Anforderungsworkshops (1)

(1) Input abholen: Projektvorschlag, Text-Dokument, Bilder, ...
(2) Input sichten und analysieren
  - Sich ein Bild verschaffen

(3) Anforderungs-Workshops durchführen
  - Wer sind die Stakeholder / Anf.-Träger / Beteiligte
  - Agenda/Methodik überlegen
  - Einladen
  - Workshop durchführen
  - Workshop dokumentieren und Ergebnisse verteilen

(4) Lastenheft erarbeiten
(5) Ggf. Prototypen bauen
(6) Lastenheft und Prototypen dem Anwender zum Prüfen übergeben
(7) Anforderungsbewertung durchführen

Verfahren und Techniken: Anforderungsworkshops (2)

Allgemeines Vorgehen beim Anforderungs-Workshop

- ☐ Aufwärmen
  - Ziele, Zeitrahmen, Zeitplanung
  - Gegenseitiges Kennenlernen
  - Ggf. kurzes Training
- ☐ Vorgehen klären
  - Rollen: Wer hat welche Aufgaben
  - Prozesse: Welche Schritte machen wir um Ziel zu erreichen
- ☐ Aufgabe ausführen
- ☐ Ergebnisse zusammenfassen

Verfahren und Techniken: Anforderungsworkshops: Beispielagenda (1)

Frust abbauen	
Warm werden	
8:00-8:45	Zielsetzung und Methodik vorstellen
8:45-9:00	Karten Vor- und Nachteilen beschriften Aktuelles System/Situation
9:00-10:00	Clustering Nachteile
	→ Pause
10:30-11:30	Clustering Vorteile
11:30-12:00	Bewertung Vorteile/Nachteile
	→ Mittagessen
12:45-13:00	Ziele und Methodik
13:00-13:30	Karten mit Anforderungen erstellen
13:30-15:00	Clustering der Anforderungen
15:00-15:30	Rollen identifizieren und zuordnen
15:30-15:45	Priorisierung der Anforderungen
	→ Szenarien durchspielen für Anwendungsfall

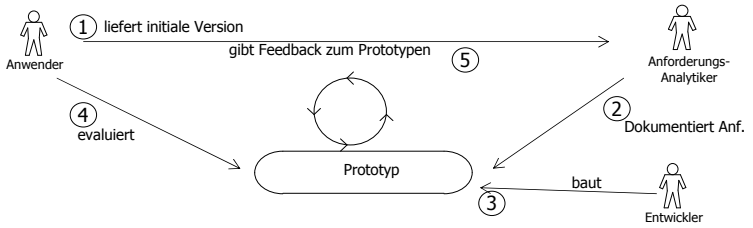
Verfahren und Techniken: Anforderungsworkshops: Beispielagenda (2)

15:45-Ende	Szenarien durchspielen für 1. Anf.
Ende	Zusammenfassung und Feedback
Nacht	Prototypen bauen
2. Tag	Szenarien
	Weitere Szenarien durchspielen

Ergebnis

Verfahren und Techniken: Exploratives Prototyping

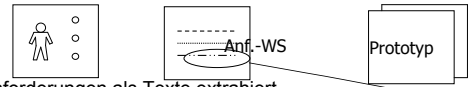
- Ansatz: Durch Zyklus von Anwender-Feedback und Prototypenbau Anforderungen herausfinden und fixieren



⇒ Prototyp ist <u>das</u> Kommunikationsmedium!

Verfahren und Techniken: Anforderungserfassung mit strukturierten Texten (1)

- Input



- Aus Input werden Anforderungen als Texte extrahiert
- Anforderungstexte werden dann gruppiert und ggf. unterstrukturiert
- Anforderungen werden in einer Tabellenform beschrieben
- Lastenheft Kapitel 3: Funktionale Anforderungen
  - → AF1: Lehrerverwaltung
    - → AF1.1: Lehrer erfassen
  - → ...

  } Hierarchische Anforderungen

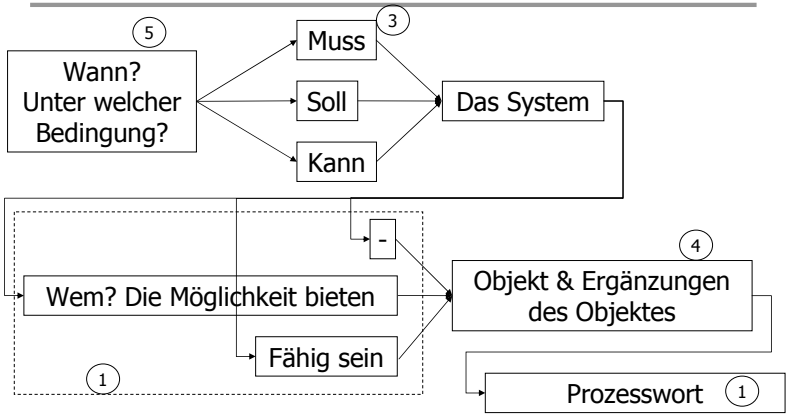
- Lastenheft Kapitel 4-5: selbes Verfahren

Verfahren und Techniken: Anforderungserfassung mit strukturierten Texten (2)

1. Bestimmen des Prozesswortes
2. Charakterisieren der Aktivität des Systems
   - Selbstständige Systemaktivität (-)
   - Benutzerinteraktion (die Möglichkeit bieten)
   - Schnittstellenanforderung (fähig sein)
3. Festlegen der rechtlichen Verbindlichkeit
   - Zwingende Anforderung (Muss)
   - Optionale Anforderung (Soll)
   - Zusätzliche Anforderung (Kann)
4. Feinschliff für den Prozess
5. Formulierung von logischen und zeitlichen Bedingungen

Beispiel: Während der Pausenplanung muss das System die aktuellen Pausenaufsichtstatistiken anzeigen. (1)

Verfahren und Techniken: Anforderungserfassung mit strukturierten Texten (3)



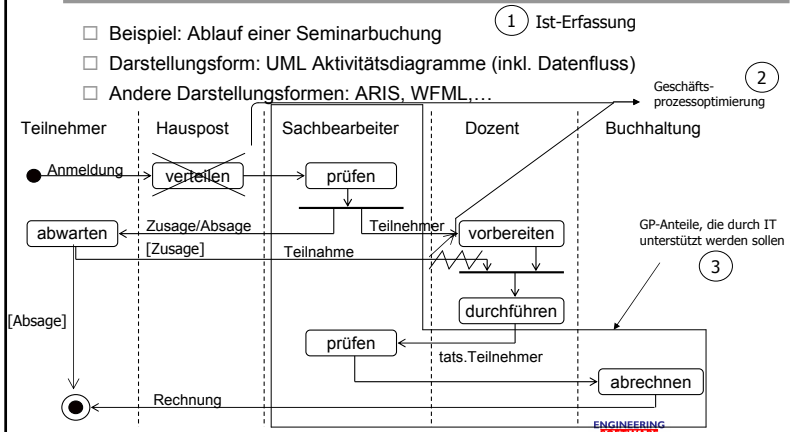
Verfahren und Techniken: Geschäftsprozessanalyse / Business Process Reengineering

- Geschäftsprozess ≙ zeitlich-logische Abfolge von Tätigkeiten
  - Erfüllung einer Betriebswirtschaftlichen Aufgabe
  - Transformation von Material und/oder Information
- GP-Analyse / BPR
  - (1) → Erfassung der IST-GP
  - (2) → Optimierung und Beschreibung der SOLL-GP
  - (3) → Identifikation der Teile des GP, die durch IT unterstützt und automatisiert werden sollen

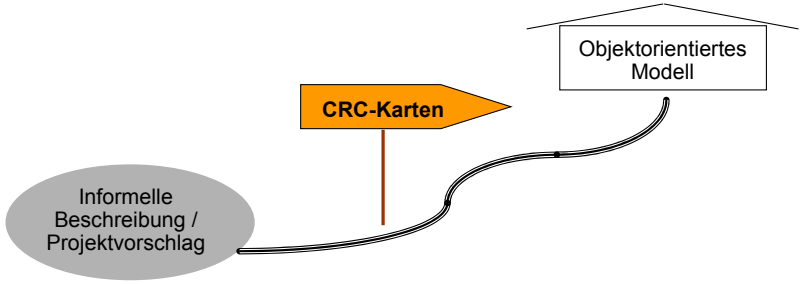
  ⇒ Anforderungen an das System

Beispiel: Geschäftsprozessanalyse und Systemgrenze

- Beispiel: Ablauf einer Seminarbuchung
- Darstellungsform: UML Aktivitätsdiagramme (inkl. Datenfluss)
- Andere Darstellungsformen: ARIS, WFML,…



Verfahren und Techniken: Szenarienbasierte Analyse Beispiel: CRC-Karten-Methode



Was sind CRC-Karten?

- CRC = Class – Responsibility – Collaborator
- Beck, Cunningham, Wilkerson, Wirfs-Brock (1989-1995)
- Technik zur Gruppenarbeit (Rollenspiele)
- Spielerische Hinführung zu objektorientiertem Denken
- Einziges Hilfsmittel: Zu beschriftende Karteikarten

Klassenname (class)	
Ober- und Unterklassen	
Verantwort-lichkeiten *(responsibili-ties)*	*Mithelfer* *(collabora-tors)*

(Vorderseite)

Klassenname (class)
Definition
Attribute *(attributes)*

(Rückseite)

CRC-Karten-Methode: Klassen finden

Substantiviertes Prozesswort + „System"

Informelle Beschreibung:

Es ist ein Terminverwaltungssystem für Arbeitsgruppen zu entwickeln. Das System soll alle geplanten Teambesprechungen (z.B. Projektbesprechungen) speichern und die Reservierung von Besprechungsräumen unterstützen. Das System soll automatisch Kollisionen mit bereits bekannten Terminen vermeiden. Deshalb soll auch die Eintragung privater Termine möglich sein.

Zu allgemein/nicht auf die Domäne beschränkt

1. Substantive
2. Streiche alles ohne klaren Verantwortungsbereich
3. Mehrfache Verantwortungsbereiche
4. Haben wir alles?
   - Mitarbeiter / Team
   - Projekt
5. CRC beschriften

CRC-Karten-Methode: Szenarien durchspielen

Teambesprechung	
Oberklassen: *Unterklassen:*	Termin
Titel wissen Datum wissen Teilnehmer wissen Teilnehmer einladen Raum festlegen	Teammitglied

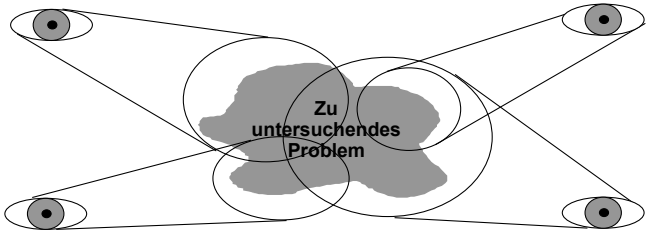
Ziel: Auffinden aller Anforderungen

Mittel: Untersuchung aus verschiedenen Gesichtspunkten

Rückseite:

Teambesprechung
Ein Objekt 'Teambesprechung' beschreibt genau einen Termin, an dem mehrere Teil-nehmer der Gruppe teilnehmen sollen.
Titel Datum

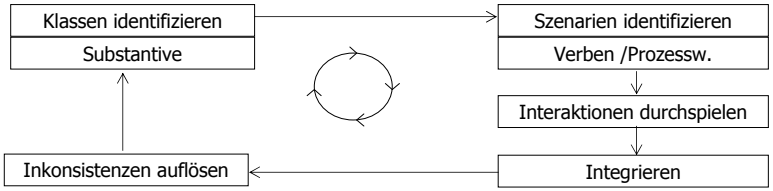
Verfahren und Techniken: Szenarienbasierte Analyse (1)



Ziel: Auffinden aller Anforderungen

Mittel: Untersuchung aus verschiedenen Gesichtspunkten / Szenarien

Verfahren und Techniken: Szenarienbasierte Analyse (2)



Ansatz: ▪Unterschiedliche Szenarien analysieren und beschreiben

▪Dann integrieren

↳„Alle im Boot"

↳„Kein blinder Fleck"

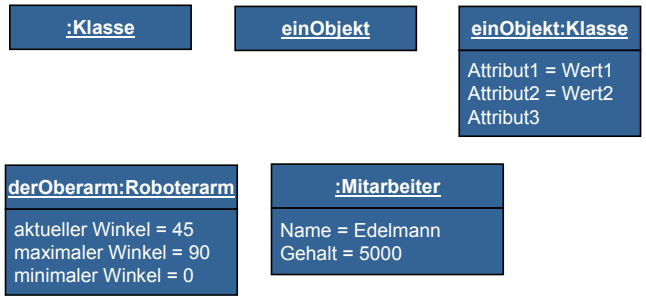
Varianten: Ausnahme-Szenarien, Fehler-Szenarien, Test-Szenarien,...

UML: Kurze Wiederholung

- Kurze Wiederholung
  - Objekte
  - Klassen
  - Operationen
- Assoziation
  - Kardinalität
  - Assoziationsnamen und Rollen
  - Sonderfälle von Assoziationen
  - Assoziative Klassen
  - Aggregationen und Kompositionen
- Vererbung
- Pakete
- Botschaften
- Szenarios

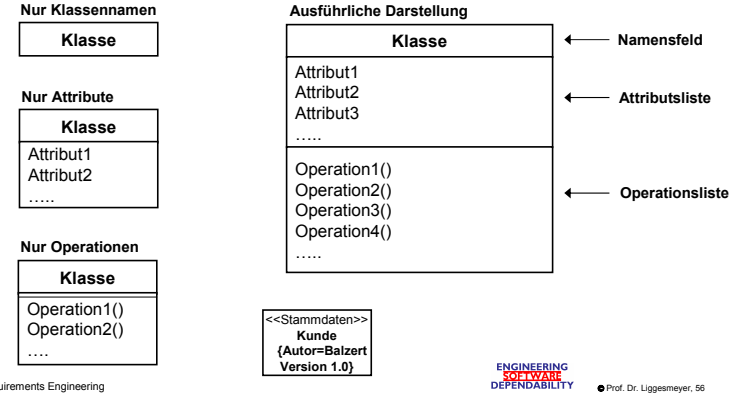
Objekte

- UML-Notation Objekt



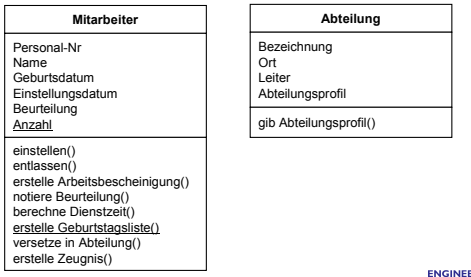
Klassen

- UML-Notation Klasse



Operationen

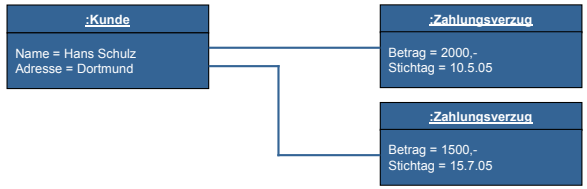
- Drei Klassen von Operationen
  - Objektoperationen (kurz: Operationen)
  - Konstruktoroperationen (kurz: Konstruktoren)
  - Klassenoperationen



Assoziation

- Fallstudie »Seminarorganisation«

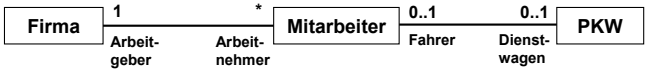
Objektdiagramm



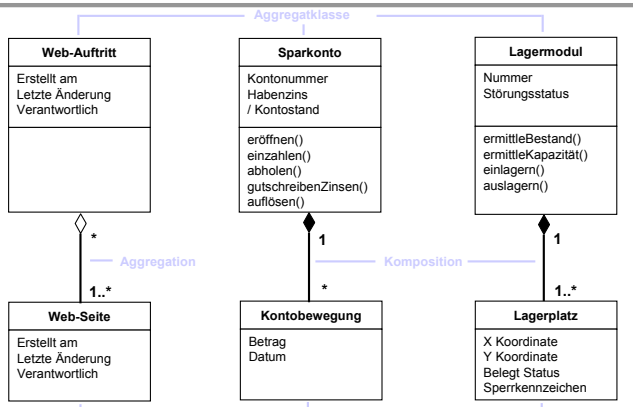
Klassendiagramm

Assoziation
Assoziationsnamen und Rollen

- Rolle
  - Beschreibt, welche Funktion ein Objekt in einer Assoziation innehat
  - Beispiel

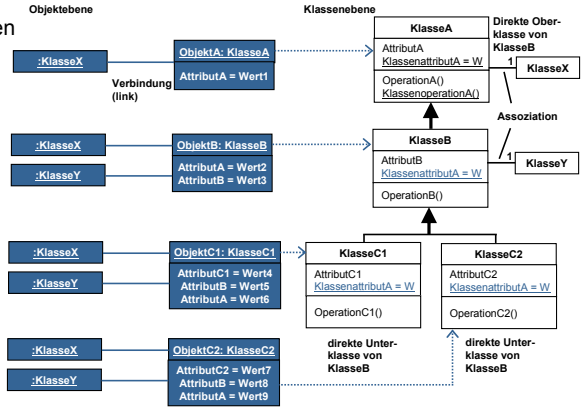


Assoziation
Aggregationen und Kompositionen: Aggregation versus Komposition



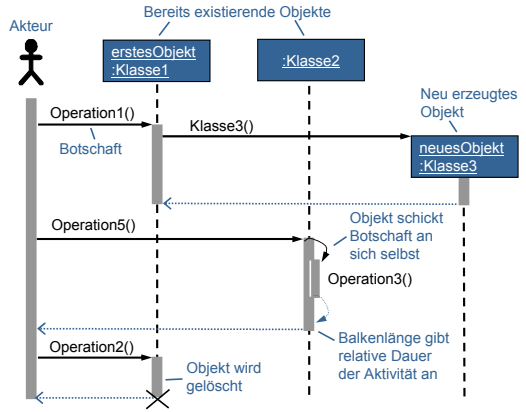
Vererbung

- Beispiel für den Vererbungs-mechanismus



Szenarios
UML-Darstellung dynamischer Abläufe

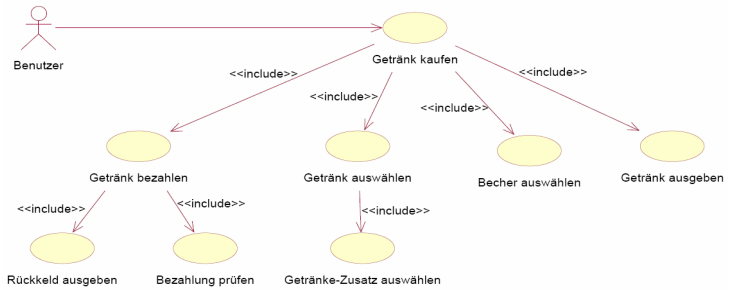
- Sequenz-diagramm



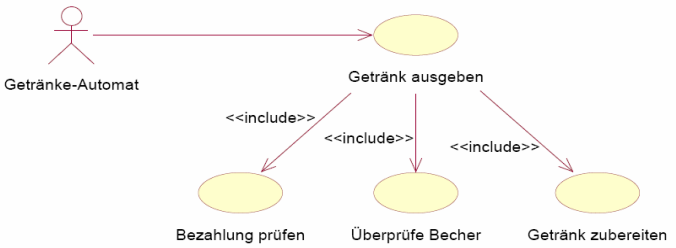
UML-Beispiel: Getränkeautomat

- Einige Diagramme aus dem UML-Modell des Getränkeautomaten

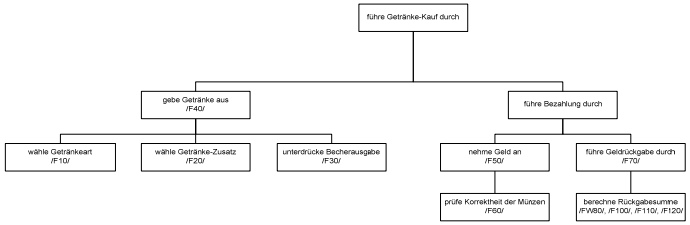
Beispiel UML: Getränkeautomat
Use-Case: Getränk kaufen



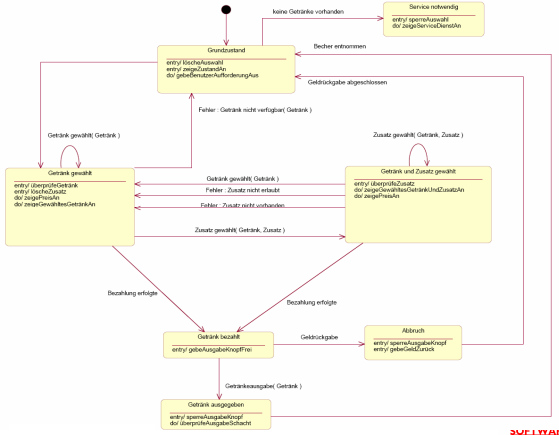
Beispiel UML: Getränkeautomat
Use Case: Getränk ausgeben



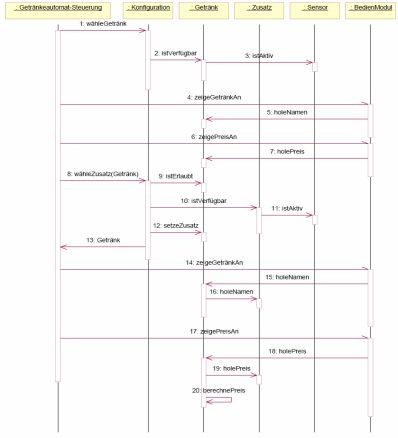
Funktionsbaum
Getränkeautomat – Grundfuntionen
Ausgabe und Bezahlung



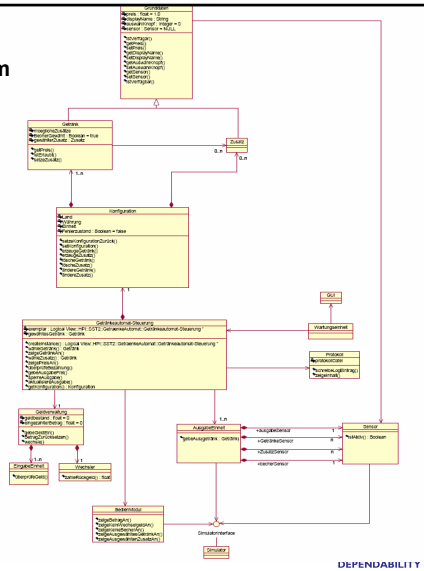
Statechart
Getränkeautomat



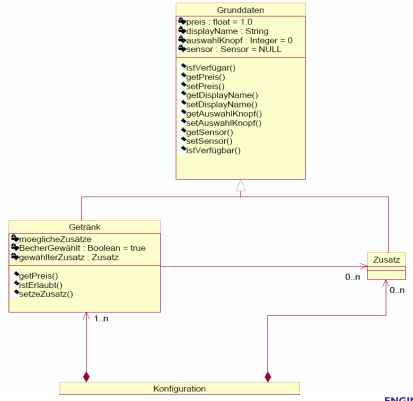
Sequenzdiagramm
Getränk wählen



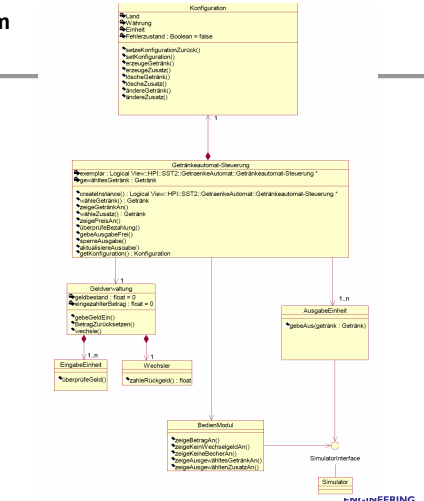
Klassendiagramm Überblick



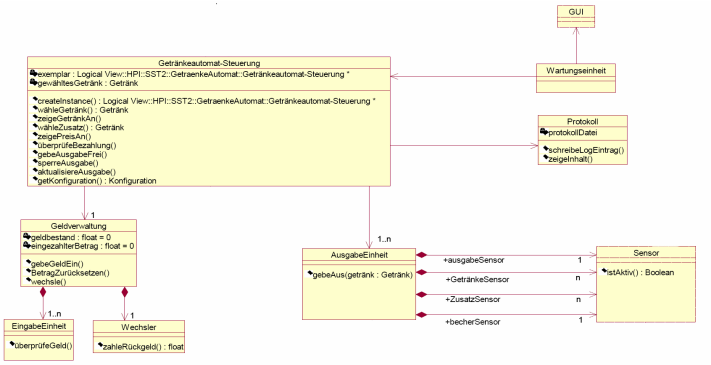
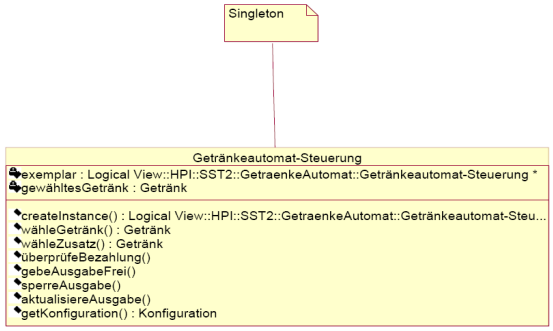
Klassendiagramm Detail (1/3)



Klassendiagramm Detail (2/3)



Klassendiagramm Detail (3/3)



Klassendiagramm
Getränkeautomat-Steuerung