

# Requirements Engineering

# BEGRIFFE

**Anforderung** ist eine Bedingung oder Fähigkeit ...

- die von einer Person zur **Lösung eines Problems** oder zur **Erreichung eines Ziels** benötigt wird
- die ein System oder Systemteile erfüllen oder besitzen muss, um einen Vertrag zu erfüllen oder **einer Norm**, einer Spezifikation oder anderen, formell vorgegebenen Dokumenten zu entsprechen

## Funktionale Anforderungen

- **Was** soll ein Produkt tun
  - Funktionen, Verhalten, Strukturen (Daten, Abhängigkeiten in einem System)

## Qualitätsanforderungen

- **Wie gut** soll ein Produkt seine Leistung erbringen
- **non-functional-requirement**
- Performance, Verfügbarkeit, Zuverlässigkeit, ...

## Rahmenbedingungen

- können (von den Beteiligten) nicht verändert werden
- werden nicht umgesetzt, sondern schränken die Umsetzungsmöglichkeiten ein

## Stakeholder

- Projektbetroffener
- Quelle für Anforderungen
- direkt: Nutzer, Administratoren
- indirekt: Management, Hacker, Gesetze

# Requirements Engineering

- Hauptaufgaben
  - **Ermitteln** der Anforderungen
  - **Dokumentieren** der Anforderungen
  - **Prüfen** und Abstimmen der Anforderungen
  - **Verwalten** der Anforderungen
- Vorgehensweise
  - kooperativ, iterativ, inkrementell
  - während des **gesamten Lebenszyklus** des Systems

# ZIELE DES REQUIREMENTS ENGINEERING

- alle relevanten Anforderungen sind **bekannt** und **verstanden**
- alle Stakeholder **stimmen** allen Anforderungen **zu** (Übereinstimmung)
- Alle Anforderungen sind
  - standardkonform **dokumentiert**
  - standardkonform **spezifiziert**

## Warum gutes Requirements Engineering?

*60% der Fehler in  
Softwareentwicklungsprojekten  
entstehen bereits im Requirements  
Engineering*

— B. Boehm



# URSACHEN FÜR FEHLER

- Anforderungen fehlen
- Anforderungen sind unklar formuliert
- Fehlerhafte Anforderungen erscheinen (trotz der Fehler) subjektiv schlüssig (für den Entwickler) oder werden (vom Entwickler) unbewusst vervollständigt

# GEGENMASSNAHMEN

1. Grenzen kennen
2. Fallen vermeiden
3. Detektiv sein
4. gemeinsame Sprache entwickeln
5. strukturiert arbeiten
6. Qualitätsprüfung der Anforderungen
7. ein Rahmen für das Ganze

# REQUIREMENTS ENGINEERING

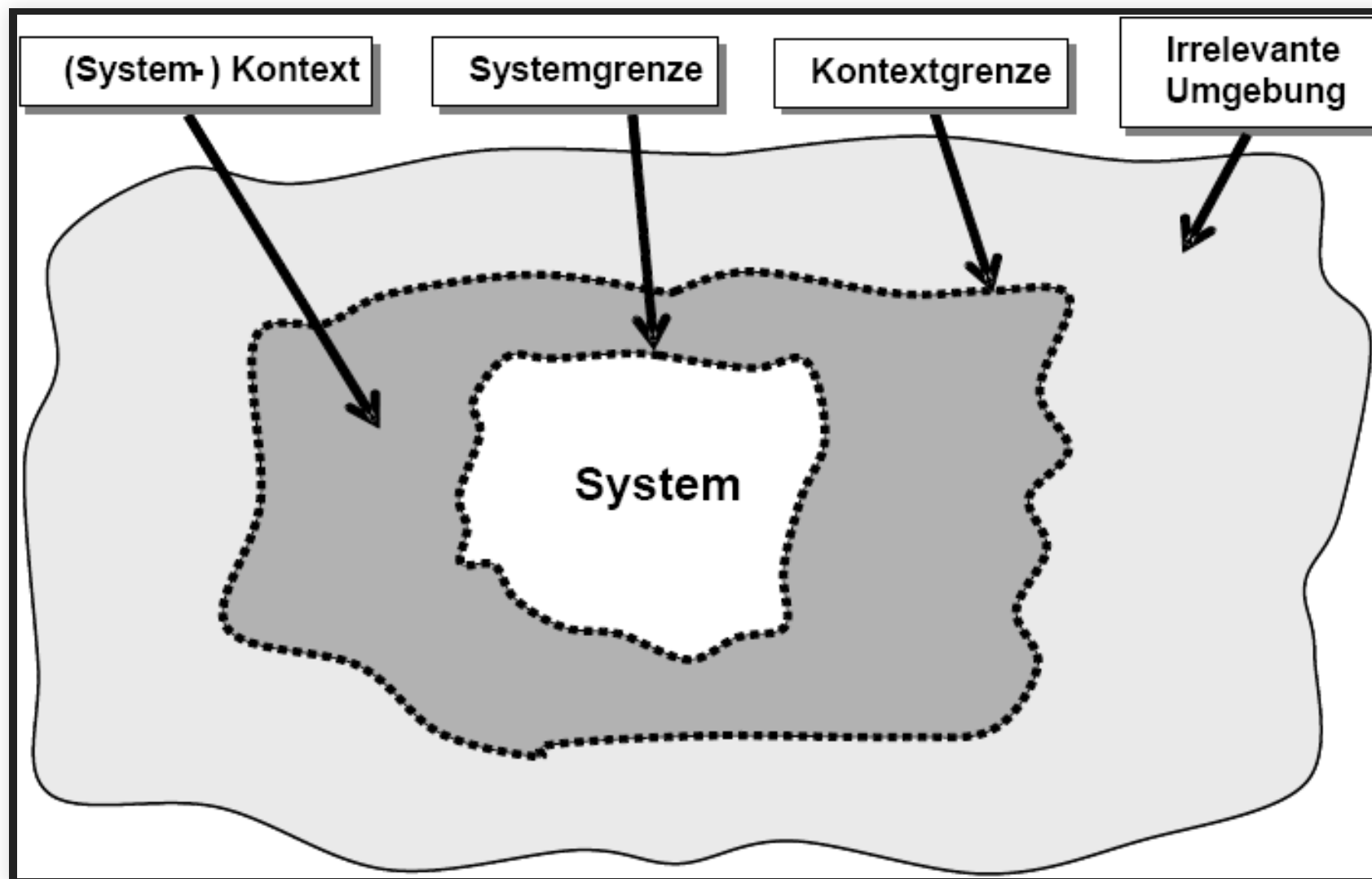
# GRENZEN KENNEN

## Systemkontext

Alle Aspekte, die eine Beziehung zu dem System haben

- Personen
- Systeme (Hardware oder andere Software)
- Prozesse, Geschäftsprozesse
- Ereignisse
- Dokumente (Gesetze, Standards)

# SYSTEM- UND KONTEXTGRENZEN



# FALLEN VERMEIDEN

- Stakeholder systematisch identifizieren und einbeziehen
- aus Projektbetroffenen sollen Projektbeteiligte werden
  - Individuelle "Verträge"



# FALLEN VERMEIDEN - KANO

Welche Bedeutung hat ein Anforderung für die Zufriedenheit eines Stakeholders?

- Unterscheidung: unterbewusst, unbewusst, bewusst
- mit der Zeit werden **Begeisterungsanforderungen** zu **Leistungsanforderungen** und später zu **Basisanforderungen**



# DETEKTIV SEIN

- Kommunikations-Geschick im Umgang mit dem Stakeholder
- Auswahl der richtigen Ermittlungstechnik
  - Befragungstechniken (Interview, Fragebogen)
  - Kreativitätstechniken (Brainstorming, Brainstorming Paradox, Perspektivenwechsel, Analogietechnik/Bisoziation)
  - Beobachtungstechniken (Feldbeobachtung, Apprenticing)

# GEMEINSAME SPRACHE

- Erstellung eines Glossars
  - Fachbegriffe, Abkürzungen, Synonyme
  - alltägliche Begriffe, die im Kontext eine andere Bedeutung haben
- Verwalten des Glossars
  - **ein** Verantwortlicher
  - zentral zugänglich

# STRUKTUR & DOKUMENTATION

Was muss dokumentiert werden?

- Stakeholder
- Systemkontext
- Glossar
- Nutzer und Zielgruppen
- Annahmen
- Alle Anforderungen

# QUALITÄTSKRITERIEN

## Anforderungsdokument muss

- Eindeutig und Konsistent sein
  - jede einzelne Anforderung
  - **kein Widerspruch** zwischen den Anforderungen
  - **identifizierbar** (Dokument & jede Anforderung)
- Klare Struktur haben
- Modifizierbar und Erweiterbar sein
- Vollständig
- Verfolgbar sein

# WIE DOKUMENTIEREN ?

## Natürliche Sprache

- "Transformation im Kopf" findet immer statt
- ggf. Satzschablonen verwenden
- Kurze Sätze, kurze Absätze
- nur eine Anforderung pro Satz
  - Aktiv formulieren, nur ein Prozesswort (Verb)

# NATÜRLICHE SPRACHE - BEISPIEL

Zur Anmeldung des Benutzers werden die Login-Daten eingegeben

Das System soll dem Benutzer ermöglichen,  
seinen Usernamen und sein Passwort  
über die Tastatur  
am Terminal einzugeben.

# SATZSCHABLONEN - BEISPIEL

Als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen>

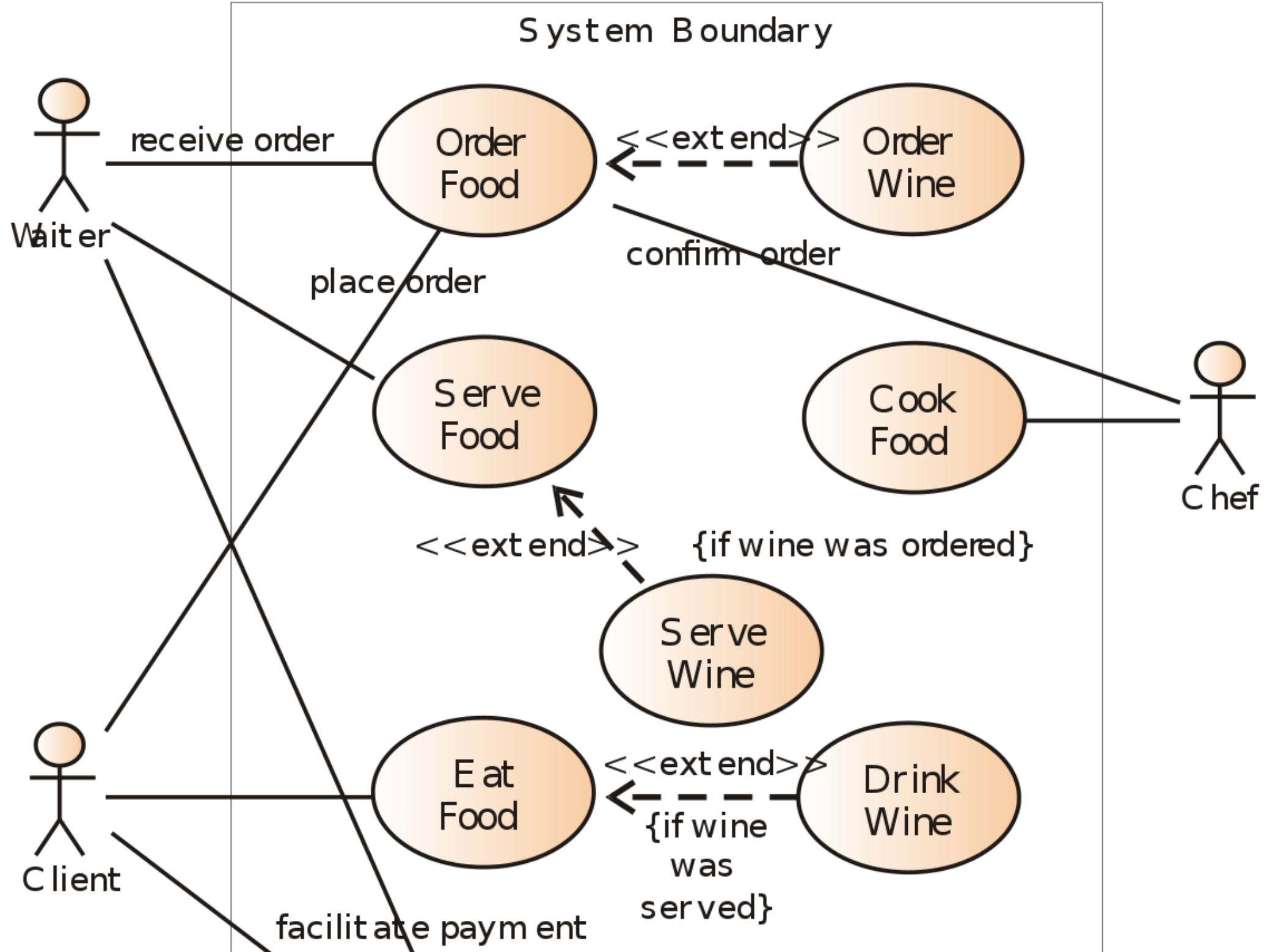
# WIE DOKUMENTIEREN ?

## Modellbasiert

- UML
  - Use-Case-Diagramme
  - Datenflussdiagramme
  - Aktivitätsdiagramme



# uc Use Cases



# DATENFLUSSDIAGRAM

# AKTIVITÄTSDIAGRAM



# QUALITÄTSKRITERIEN FÜR ANFORDERUNGEN

# REQUIREMENTS-MANAGEMENT ALS RAHMEN

Verwalten und Kontrollieren aller Aufgaben des Requirements Engineering während der komplette Lebenszeit des Produktes.

- **Attributierung** der Anforderungen
- **Priorisierung** der Anforderungen
- **Verfolgbarkeit** der Anforderungen
- **Versionierung** der Anforderungen

# ATTRIBUTIERUNG BEISPIEL

<i>Attributname</i>		<i>Belegung des Attributs (Attributwert)</i>	
Identifikator	Name		
Req-10	Dynamische Stauumfahrung		
Anforderungsbeschreibung			
Das System soll beim Auftreten von Verkehrsbehinderungen, die einen konfigurierbaren Kritikalitätswert übersteigen, selbständig eine Ausweichroute berechnen.			
Stabilität	Verantwortlicher	Quelle	Autor
gefestigt	P. Müller	Produktmanagement	B. Wagner

# VERFOLGBARKEIT

*Eine Anforderung ist nachvollziehbar, wenn sowohl deren Ursprung als auch deren Umsetzung und die Beziehung zu anderen Dokumenten nachvollziehbar ist.*

Andere Dokumente: Commit-Historie, Testplan, Testprotokoll

# VORTEILE VERFOLGBARKEIT

- Nachweisbarkeit
- Identifikation von Goldrandlösung
- Auswirkungenanalyse
- Zuordnung von Entwicklungsaufwänden



# QUELLEN BILDER

- Kontextabgrenzung <http://docplayer.org/docs-images/24/4428614/images/7-0.png>
- Kano Modell <http://smallthingsmatter.ch/kano/>
- UseCase Diagram [https://en.wikipedia.org/wiki/Use\\_case\\_diagram](https://en.wikipedia.org/wiki/Use_case_diagram)
- Datenflussdiagramm <http://www.ritz-dv.de/beratungsangebot/systemanalyseabb.php>
- Aktivitätsdiagramm <https://de.wikipedia.org/wiki/Aktivitätsdiagramm>
- Attributierung von Anforderungen "Basiswissen Requirements Engineering" - Pohl, Rupp