
Bachelor of Science (BSc) in Informatik
Modul Advanced Software Engineering 1 (ASE1)

LE 03 - Requirements Engineering

2 System und Systemkontext abgrenzen

Institut für Angewandte Informationstechnologie (InIT)

Walter Eich (eicw) / Matthias Bachmann (bacn)

<https://www.zhaw.ch/de/engineering/institute-zentren/init/>

Agenda

- 2 System und Systemkontext abgrenzen**
 - 2.1 Systemkontext
 - 2.2 System- und Kontextgrenzen bestimmen
 - 2.2.1 Die Systemgrenze festlegen
 - 2.2.2 Die Kontextgrenze bestimmen
 - 2.3 Den Systemkontext dokumentieren
 - 2.4 Wrap-up

Lernziele

LZ 2.1 Systemkontext, System- und Kontextabgrenzung kennen

LZ 2.2 System- und Kontextgrenze bestimmen und anwenden

Systembegriff (1/4)

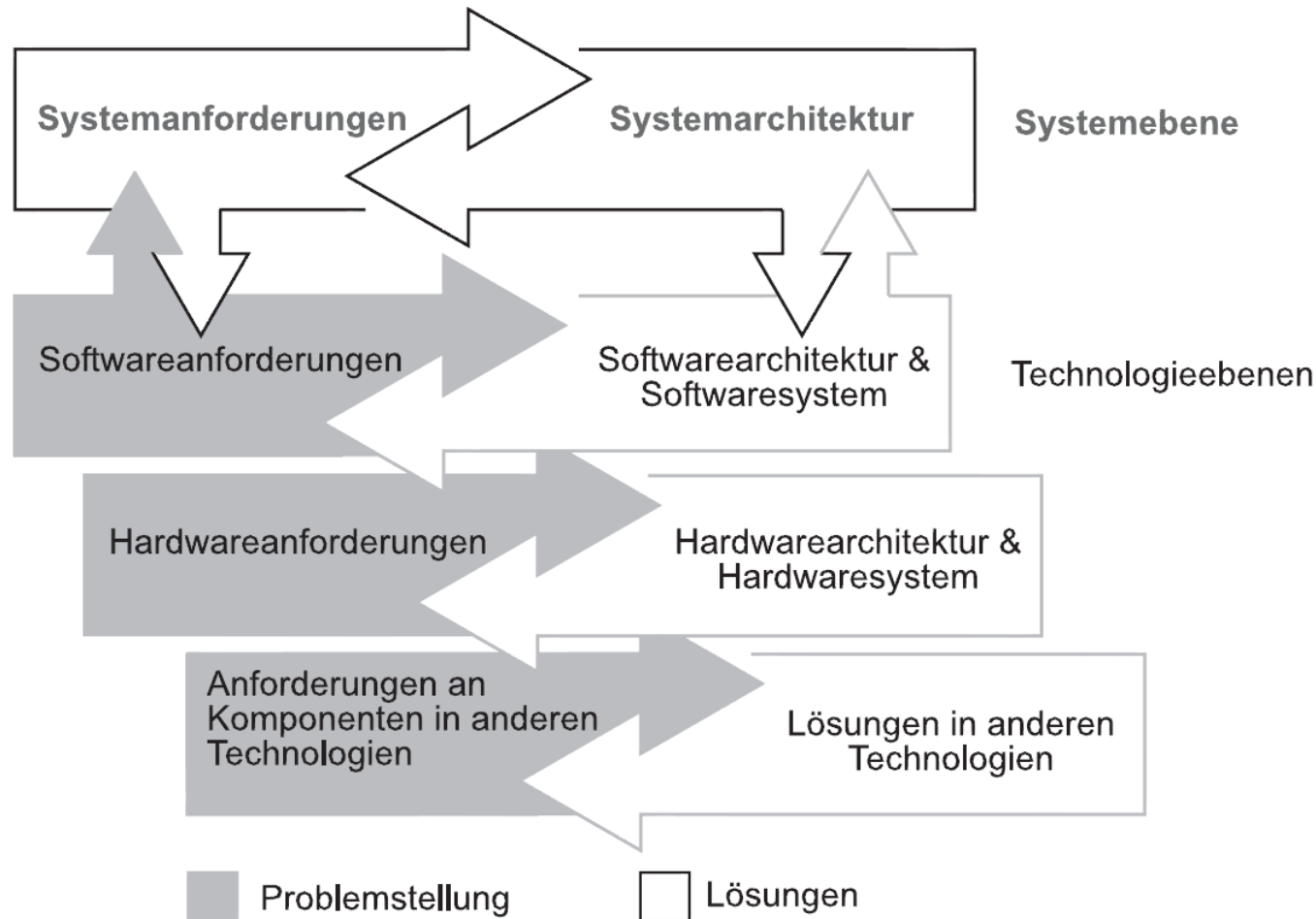
- Die **Unterscheidung von System- und Technologieebene** soll uns helfen, den Umfang des jeweils betrachteten Entwicklungsgegenstandes klar im Blick zu haben:
 - Auf der **Systemebene** betrachten wir **das Produkt oder das System**, das der Kunde von uns haben möchte, zum Beispiel einen Videorecorder, ein Flugsicherungssystem oder ein Mobiltelefon **in seiner Gesamtheit**.
 - Bei dieser **Gesamtsicht** spielt es **noch keine Rolle, welche Teile als Software- oder als Hardwarelösungen entwickelt werden**.
 - Hier erfassen Sie Forderungen an das Verhalten und die Eigenschaften des Gesamtsystems.



Systembegriff (2/4)

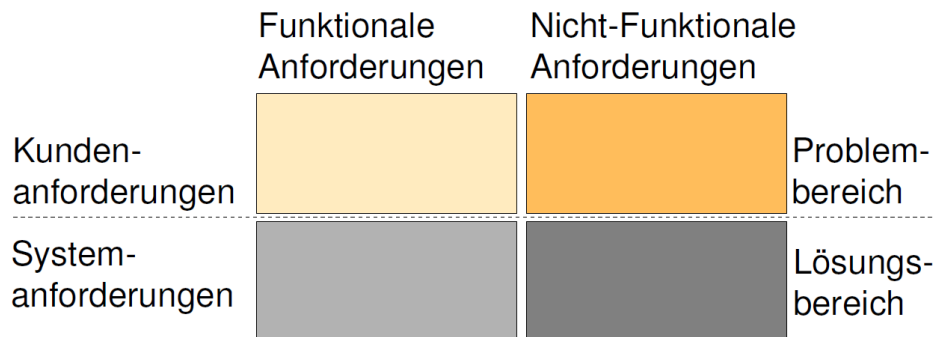
- Ein System ist ein **dynamisches Ganzes**,
 - das als solches **bestimmte Eigenschaften und Verhaltensweisen** besitzt.
 - Es **besteht aus Teilen**, die so miteinander verknüpft sind, dass **kein Teil unabhängig ist von anderen Teilen** und
 - das **Verhalten des Ganzen** beeinflusst wird vom **Zusammenwirken aller Teile**.
- Der Beweis, dass der Bankomat ein System ist, lässt sich wie folgt begründen:
 - Der Bankomat besitzt die **Eigenschaft**, dass er 24-Stunden zur Verfügung steht und den Kunden unter anderem Geld ausgibt sowie den Kontostand anzeigen kann.
 - Die **Dynamik** in diesem System besteht darin, dass Kunden Interaktionen mit dem System durchführen können.
 - Der Bankomat besteht aus **vielen einzelnen Teilen**, beispielsweise der Hard- und Software.

Systembegriff (3/4)



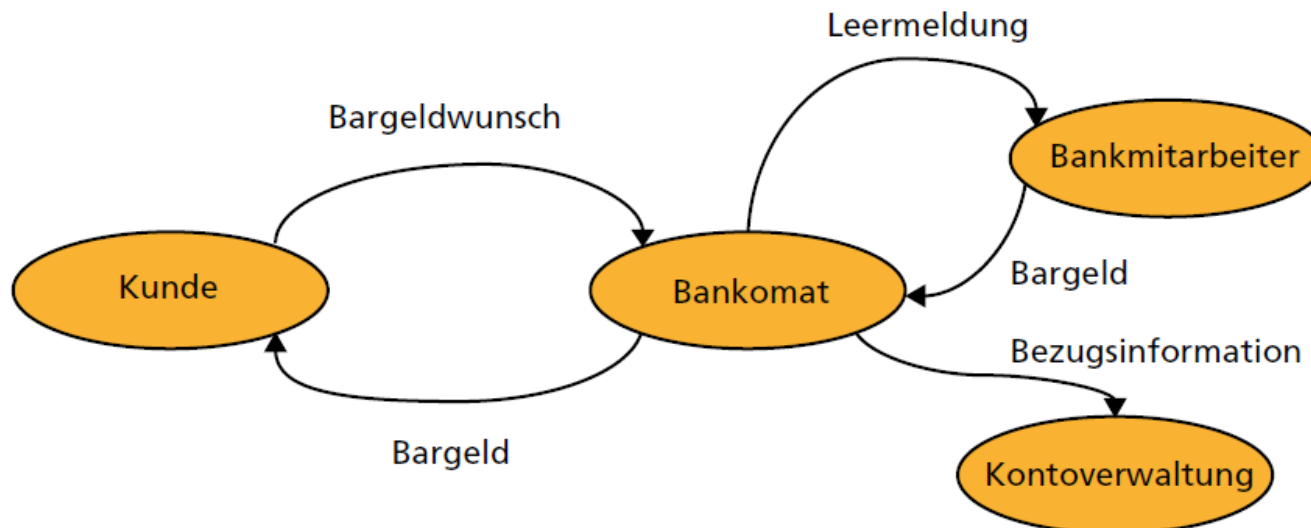
Systembegriff (4/4)

- Sie sollten bei Ihrer Entwicklung **Problemstellung und Lösung klar trennen**,
 - denn **die Problemstellung ist meist länger stabil, als die Lösungen es sind**,
 - und **zu einer Problemstellung finden Sie normalerweise mehr als eine mögliche Lösung**.
- Aus diesen Gründen streben wir die getrennte Formulierung von Problemstellung und Lösung auf jeder Ebene der Entwicklung an.



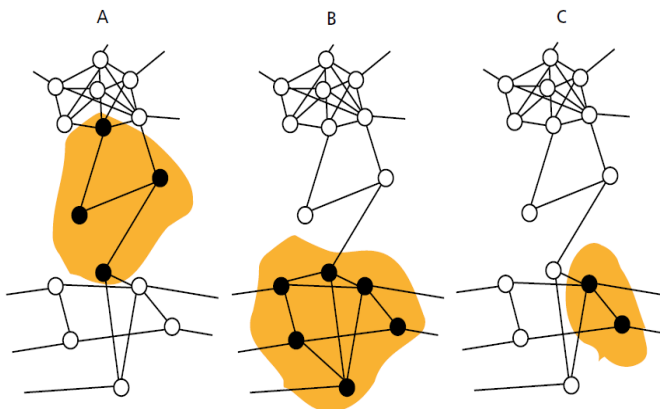
Offene Systeme

- Bei einem offenen System gibt es mindestens eine Interaktion mit einem anderen System.
 - Die Buchungsdaten werden vom System «Verkaufssystem» in das System «Buchhaltungssystem» übertragen.
 - Das System «Bankomat» übergibt einem System «Menschen» CHF 100.00.



Systemgrenze und Umwelt

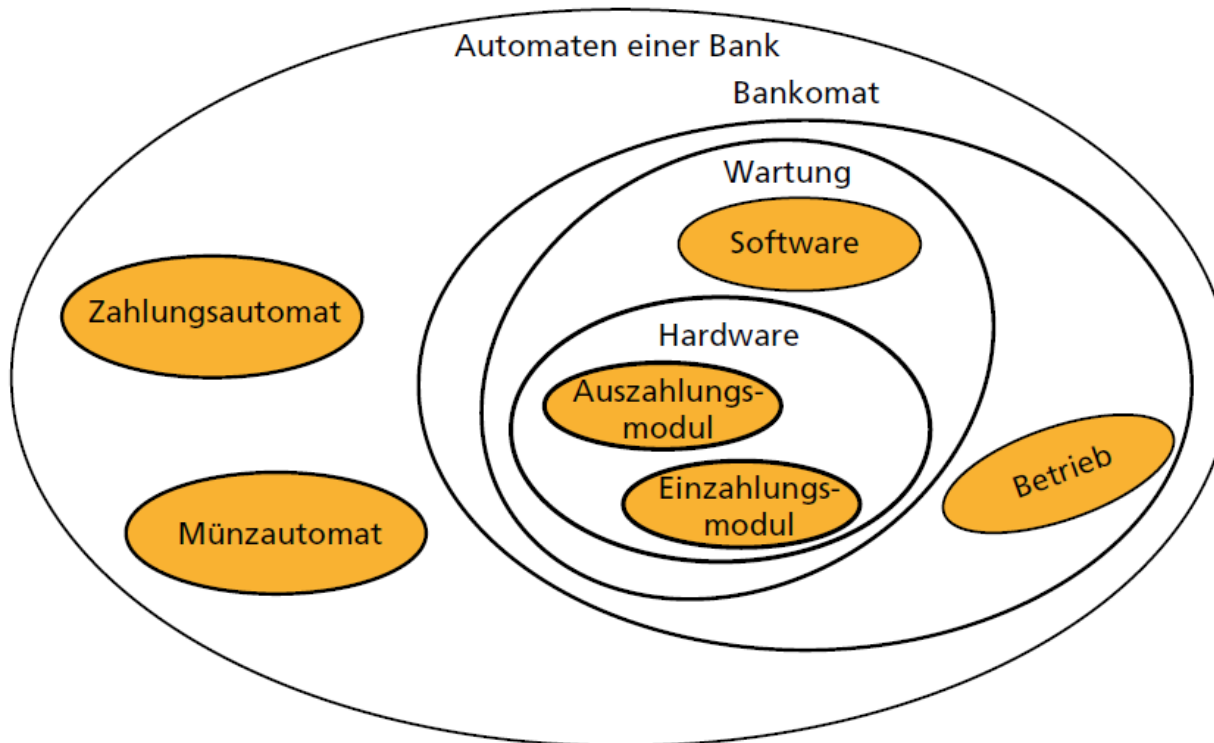
- Die **Systemgrenze** separiert das geplante System von seiner **Umgebung**.
 - Sie grenzt den im Rahmen des Entwicklungsprozesses **gestaltbaren und veränderbaren Teil der Realität** von **Aspekten in der Umgebung** ab, die durch den Entwicklungsprozess nicht verändert werden können.
 - Bei der Festlegung der Systemgrenze ist grundsätzlich darauf zu achten, dass die **Elemente innerhalb des Systems eine höhere Verbindungsichte** (Prinzip des Übergewichts der inneren Bindung) aufweisen als die Elemente am Rande.



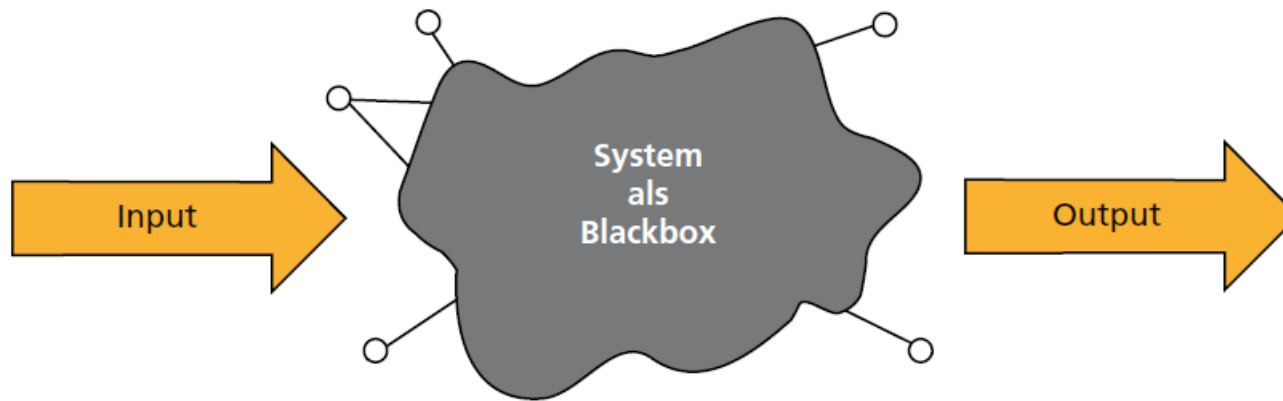
System	Anzahl Elemente im System	Anzahl Beziehungen im System	Anzahl Beziehungen zur Umwelt
A	4	3	7
B	6	9	6
C	2	1	6

Systemhierarchie

- Es können folgende Hierarchien unterschieden werden:
 - System, Element, Untersystem, Übersystem, Umsystem, Teilsystem



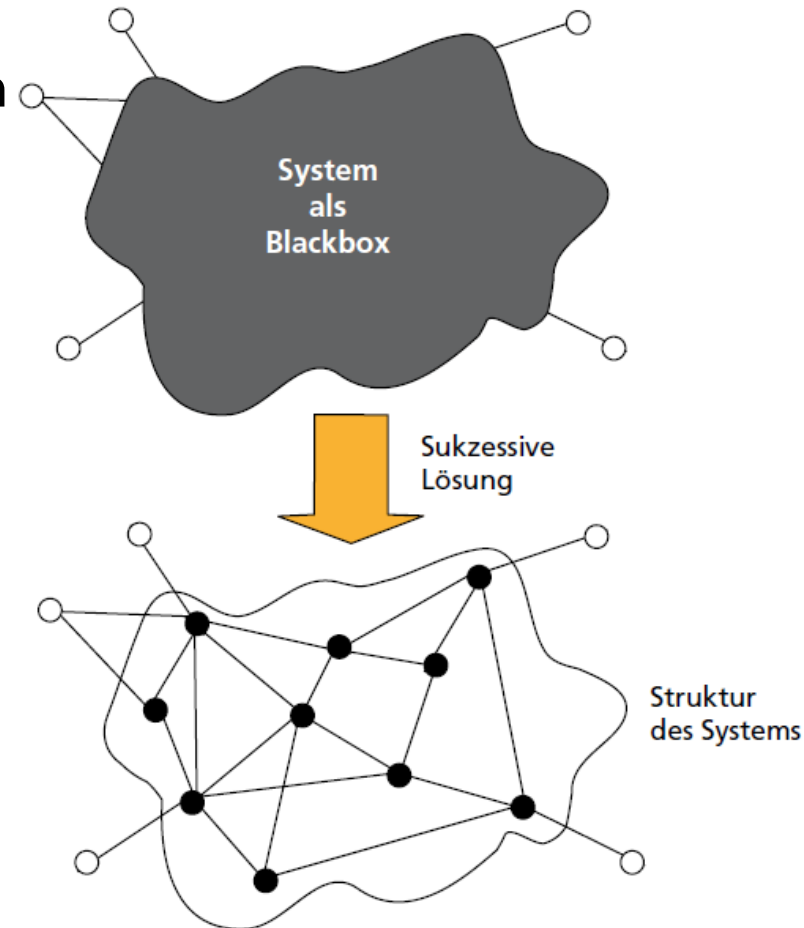
Systembetrachtung – 1. Wirkung nach aussen



- Bei der Betrachtung und der Gestaltung von Systemen kann es sinnvoll sein, **zunächst nur die Schnittstellen zwischen einem System und der Umwelt zu untersuchen**.
 - Das System wird dabei als **Blackbox** aufgefasst, und es werden nur die Schnittstellen zur Umwelt analysiert. Somit sind der **Input und der Output des Systems** ersichtlich.
 - Die Darstellung erfolgt in einem sogenannten Kontextdiagramm

Systembetrachtung – 2. Strukturorientiert

- Bei der **strukturorientierten Systembetrachtung** interessiert neben den umliegenden Systemen und deren Schnittstellen **auch das Innenleben**. (**Whitebox**) des Betrachtungssystems.
 - Somit wird das System inklusive aller Elemente und Beziehungen analysiert.
 - Mögliche Darstellung mittels Use Case Diagramm

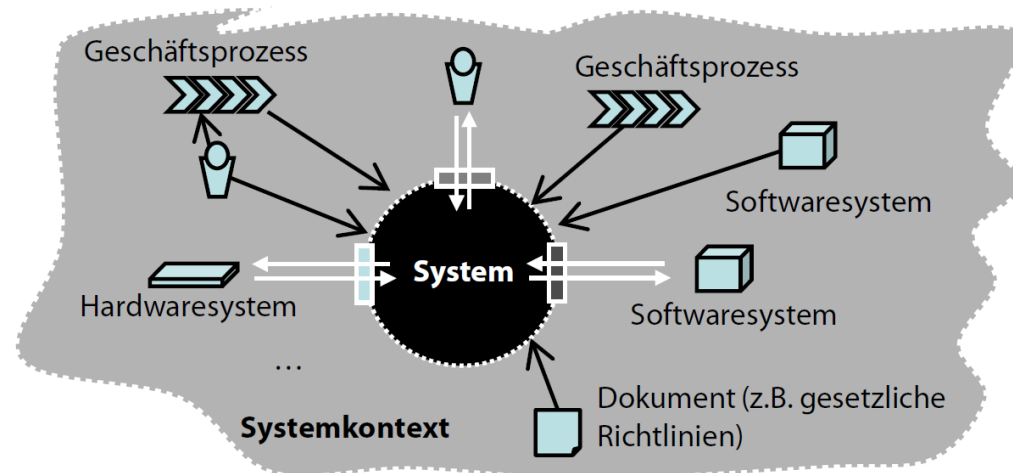


2.1 Systemkontext

- Identifikation der materiellen und immateriellen Aspekte, die eine Beziehung zum System haben
 - Die Beziehungen zwischen den einzelnen materiellen und immateriellen Aspekten mit dem geplanten System sollen identifiziert werden
- Der für die Anforderung des Systems relevante Ausschnitt der Realität wird als **Systemkontext** bezeichnet

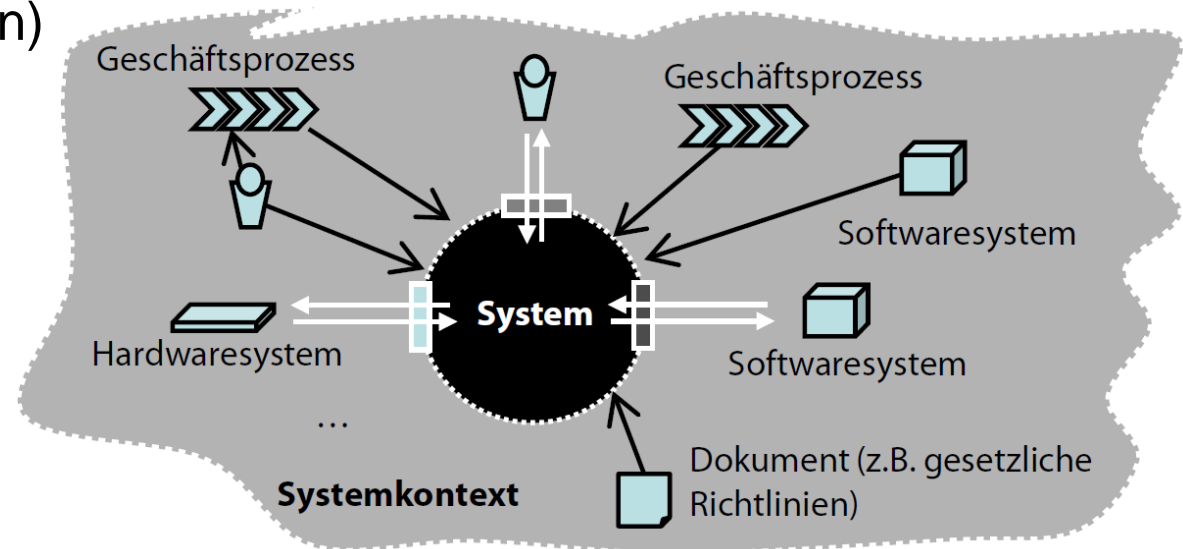
Definition Systemkontext:

Der Systemkontext ist der Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen des betrachteten Systems relevant ist.



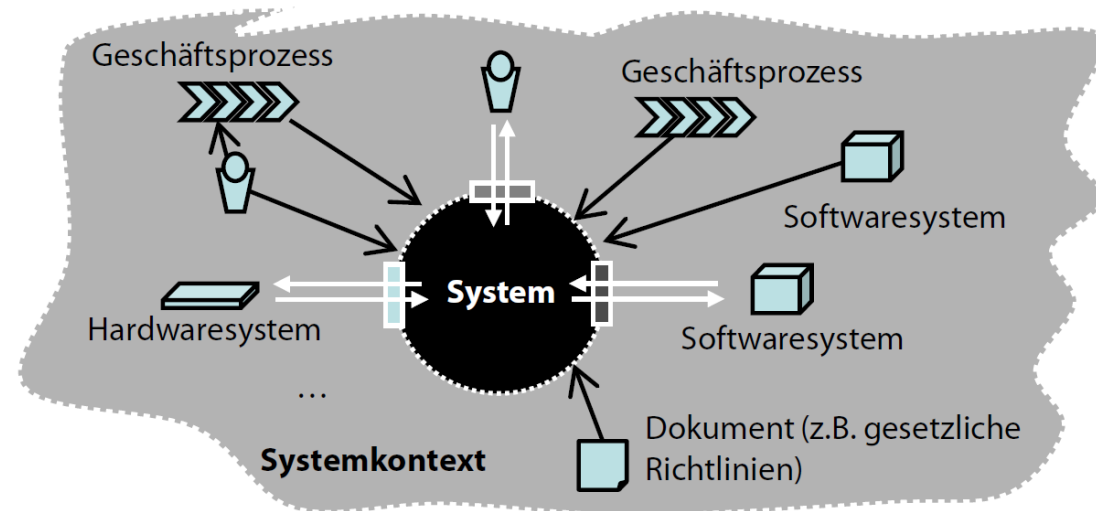
Aspekte im Systemkontext

- **Personen** (Stakeholder oder Stakeholdergruppen)
- **Systeme** im Betrieb (andere technische Systeme oder Hardware)
- **Prozesse** (technische oder physikalisch, Geschäftsprozesse)
- **Ereignisse** (technisch oder physikalisch)
- **Dokumente** (z.B. Gesetze, Standards, Systemdokumentationen)



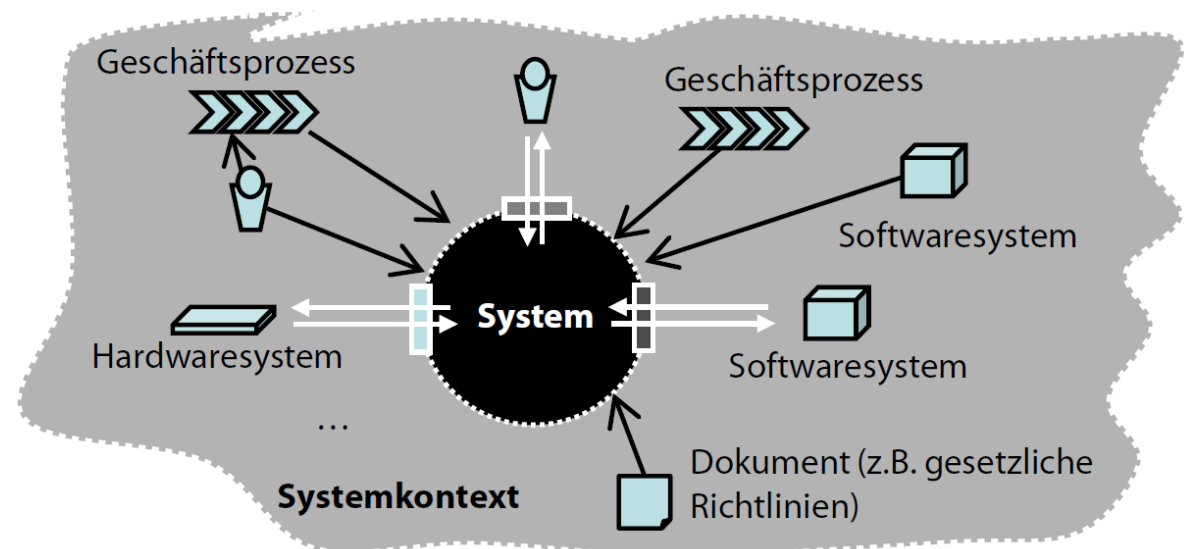
Konsequenzen eines fehlerhaften oder unvollständigen Kontextes

- Unvollständige oder fehlerhafte Systemkontexte führen zu fehlerhaften Anforderungen
- Entwicklung wird aufgrund von unvollständigen oder fehlerhaften Anforderungen durchgeführt.
- Diese Fehler bleiben bei der Überprüfung, ob das System die spezifizierten Anforderungen abdeckt, **unentdeckt**. Fehler treten erst später im Betrieb auf.



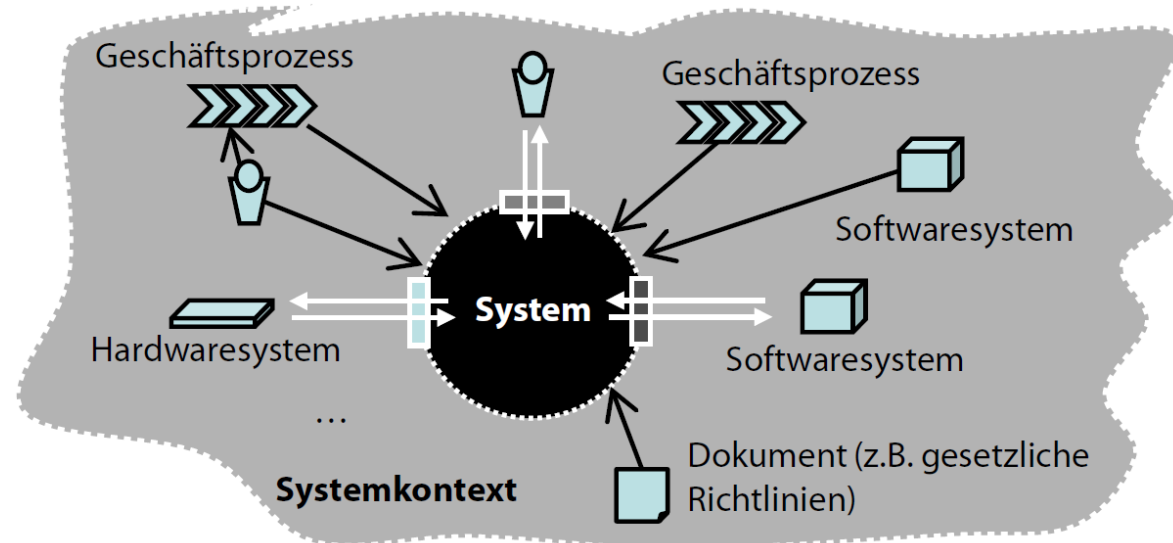
Arten von Aspekten

- **Quellen** und **Senken als Ausgangspunkt** der Abgrenzung
- Quellen und Senken: Stakeholder(gruppen) und existierende Systeme
- Schnittstellen: Interaktion zwischen System und Umgebung



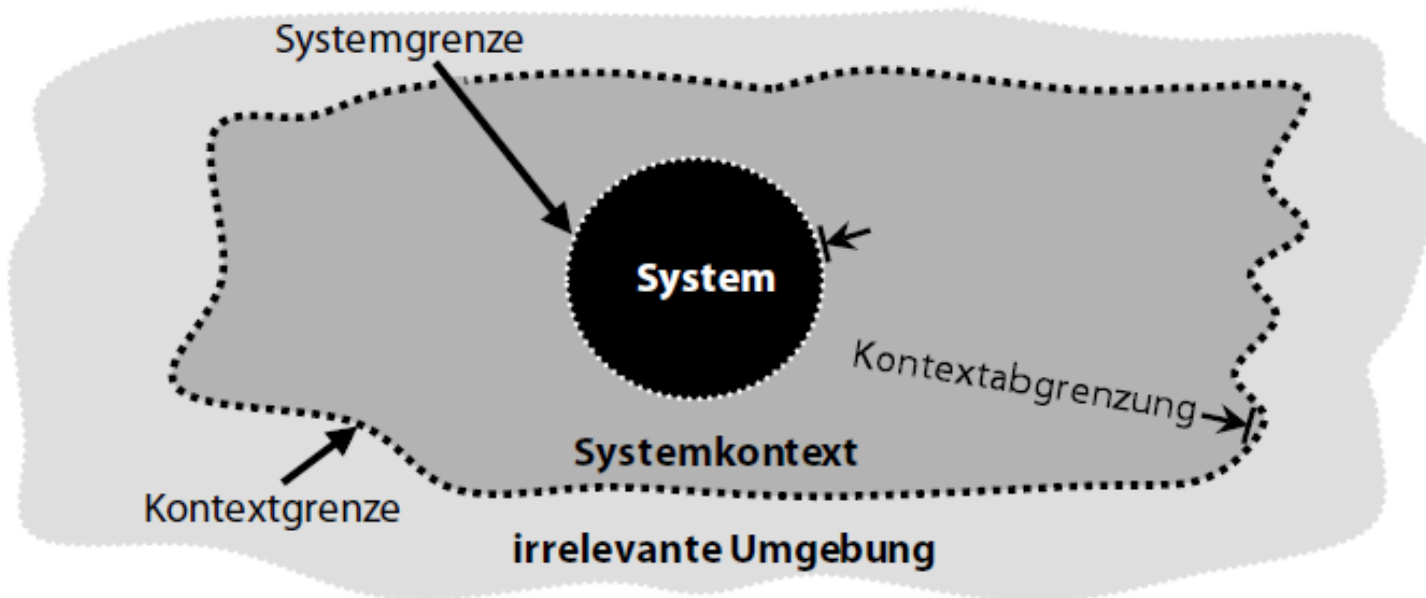
Systemkontext und Anforderungskontext

- Der **Ursprung der Anforderung** eines Systems liegt im **Systemkontext** des geplanten Systems
- Eine Anforderung ist in einem spezifischen Kontext definiert.
- Je vollständiger der Kontext einer Anforderung definiert ist umso geringer ist der Wahrscheinlichkeit der Fehlinterpretation.



2.2 System- und Kontextgrenzen bestimmen

- **Systemabgrenzung**: welche Aspekte werden durch das geplante System abgedeckt
- **Kontextabgrenzung**: Grenze des Kontexts zur irrelevanten Umgebung



2.2.1 Systemgrenze festlegen

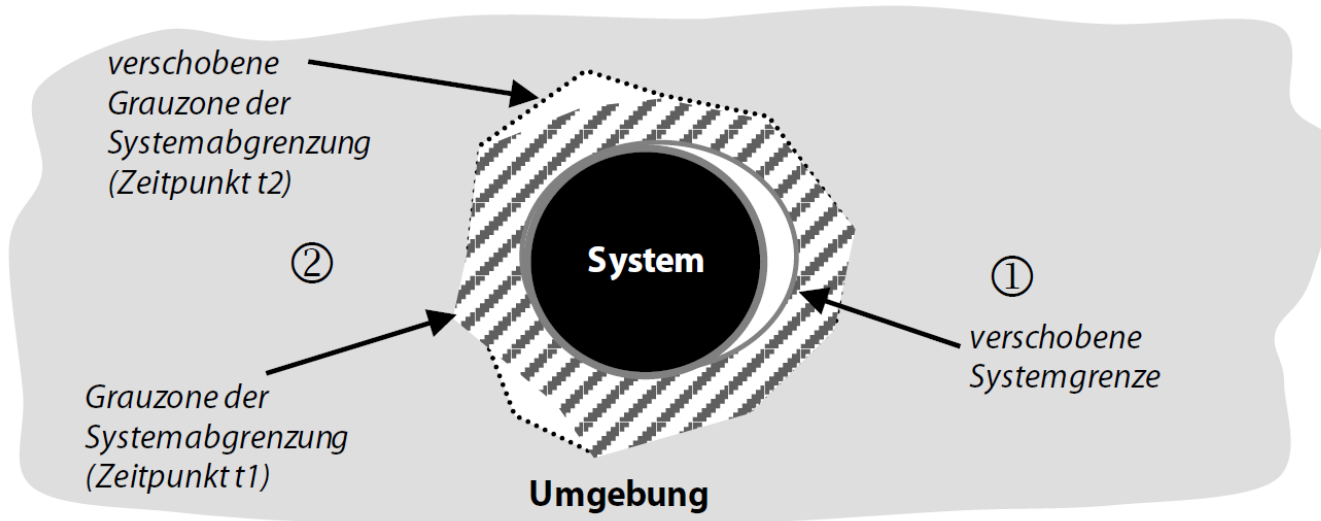
- Die **Systemgrenze** grenzt den Konstruktionsgegenstand zur Umgebung hin ab.
- Durch die Wahl der Systemgrenze wird festgelegt, **welche Aspekte das zu konstruierende System (Scope) abdeckt und was ausserhalb der Systems liegt.**
- **Sämtliche Aspekte die innerhalb der Systemgrenze liegen, können somit verändert bzw. gestaltet werden.**

Definition Systemgrenze:

Die Systemgrenze separiert das geplante System von seiner Umgebung. Sie grenzt den im Rahmen des Entwicklungsprozesses **gestaltbaren und veränderbaren Teil von Aspekten** der Umgebung ab, die durch den Entwicklungsprozess nicht verändert werden können.

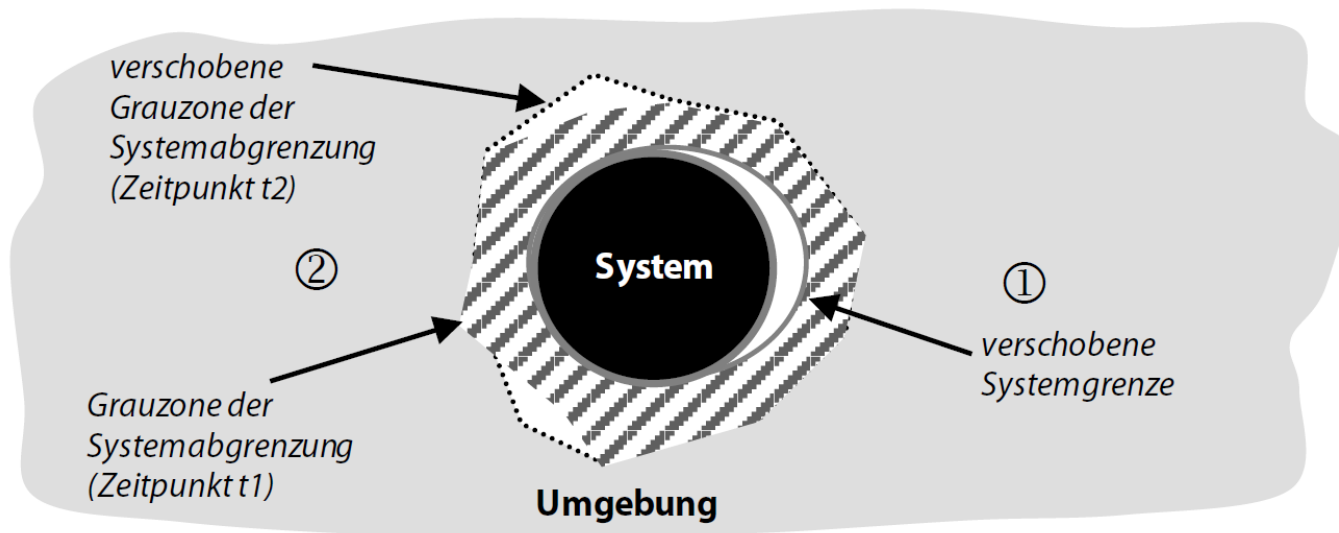
Grauzone der Systemabgrenzung

- Die **Systemgrenze** ist oft erst gegen Ende des Requirements Engineering Prozesses präzise festgelegt.
- Zu **Beginn** sind z.B. einige oder mehrere Schnittstellen nur **unvollständig** bekannt.
- Es entsteht zu bestimmten Zeitpunkten im Requirements Engineering Prozess **eine Grauzone** in der Systemabgrenzung (z.B. werden bestimmte Funktionen gebraucht oder nicht).



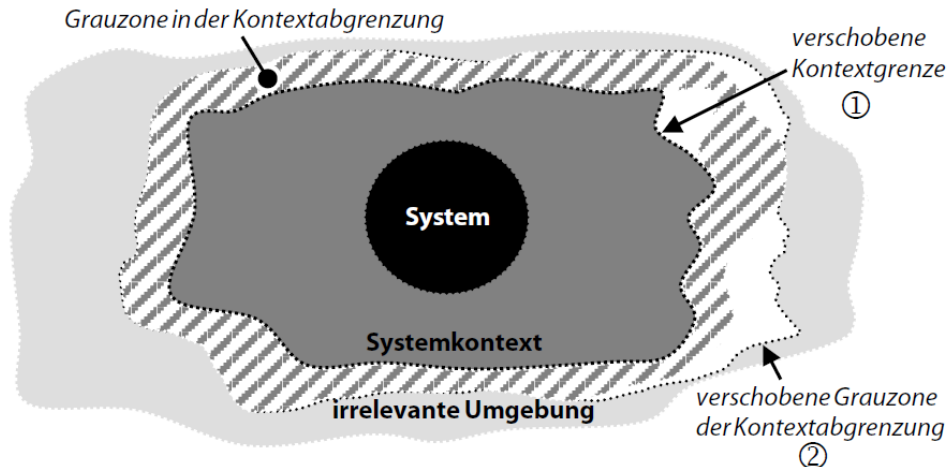
Verschiebung der Grauzone

- Neben einer **Verschiebung der Systemgrenze** in der der Grauzone kann sich im Verlaufe des Requirements Engineering Prozesses die Grauzone des Systemabgrenzung selber verschieben.
- Z.B. Aspekte die vorher zum Systemkontext gehörten sollen im Rahmen der Entwicklung nun doch verändert werden sollen.



2.2.2 Kontextgrenze bestimmen

- Die Kontextgrenze differenziert in der Umgebung des geplanten Systems zwischen Kontextaspekten und Aspekten die nicht relevant sind.



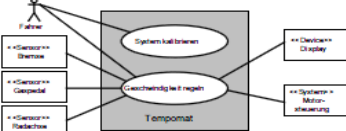
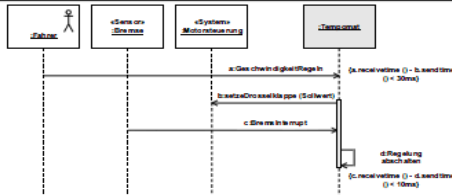
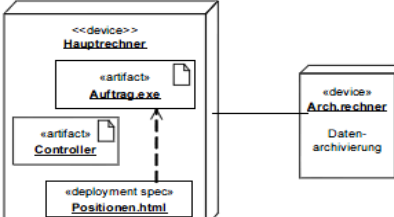
Definition Kontextgrenze:

Die Kontextgrenze separiert den relevanten Teil der Umgebung eines geplanten Systems vom irrelevanten Teil, d.h. dem Teil der Umgebung, der keinen Einfluss auf das geplante System und damit auch keinen Einfluss auf die Anforderungen dieses Systems hat.

2.3 Systemkontext dokumentieren

- Zum Dokumentieren des Systemkontextes (insbesondere der System und Kontextgrenzen) werden oftmals Use Cases oder Datenflussdiagramme eingesetzt.
 - Bei **Datenflussdiagrammen** werden oft Quellen und Senken modelliert (Datenflüsse, Materialflüsse, Energieflüsse, Geldflüsse).
 - In **Use Case Diagrammen** werden oft die Akteure (Personen, Rollen, Systeme) in der Umgebung des Systems und die Nutzungsbeziehung modelliert.

2.3 Arten den Systemkontext zu dokumentieren

Art der Kontextbildung	Schnittstelle	Diagrammart	
logisch	unspezifiziert	UML Use-Case-Diagramm	
	Ein-/Ausgaben	Kontextdiagramm der Strukturierten Analyse	
	Ein-/Ausgaben	UML Klassendiagramm	
	Nachrichten	UML Sequenzdiagramm	
physikalisch	Kanäle	UML Verteilungsdiagramm	

Wrap-up

- Der Ursprung und damit auch die **Rechtfertigung der Anforderungen** eines Systems **liegen im Systemkontext** des geplanten Systems.
- Zu den möglichen Aspekten im Systemkontext gehören u.a.: **Personen, Systeme im Betrieb, Prozesse, Ereignisse, Dokumente.**
- Aufgabe der **Systemabgrenzung** ist es festzulegen, welche Aspekte durch das geplante System abgedeckt werden (Scope), und welche Aspekte Teil der Umgebung dieses Systems sind.
- Die **Systemgrenze** ist häufig **erst gegen Ende des RE-Prozesses präzise** festgelegt.
- Deshalb gibt es eine **Grauzone**, in der die mögliche Systemgrenze liegt.
- Zur Dokumentation des Systemkontexts (insbesondere der System- und Kontextgrenzen) werden oftmals **Use-Case-Diagramme** und **Datenflussdiagramme** eingesetzt.