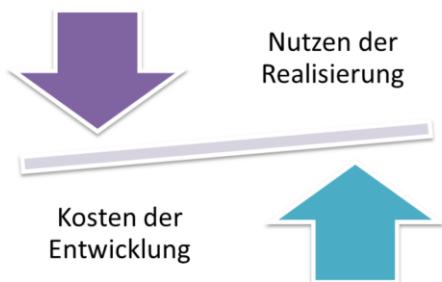


Wertbasierte Priorisierung



$$BW = \frac{Za}{(1 + Zi)^t}$$

Kosten und Nutzen **realistisch** rechnen:
• Kumulierter Nutzen über die Lebensdauer
• Kapitalkosten (inkl. Verzinsung) über die Entwicklungs- und Lebensdauer

Besonders geeignet, wenn viele wichtige aber nicht unverzichtbare Funktionen und Merkmale für ein System gefordert sind

M. Denne, J. Cleland-Huang: The Incremental Funding Method – A Data Driven Approach to Software Development, IEEE Software, 2004

[-> Beispiel-Barwert]

BW ist der Gegenwartswert einer zukünftigen Zahlung Za

Zi ist der Zinssatz, mit dem abgezinst wird

t ist die Zeit, zu der die Zahlung erfolgt (in der Regel in Jahren gezählt)

- Wertbasierte Priorisierung setzt präzise und absolute Werte für Kosten und Nutzen voraus
- Der **Cost-Value Approach** bewertet Requirements anhand von **Schätzungen** der **Kosten** und des **Nutzens**
- Cost-Value Approach verwendet Expertenschätzungen für Nutzen (durch Domänenexperten) und Kosten (durch erfahrene Entwickler)

Cost-Value-Approach: Vorgehen

Schritt 1: Nutzen bewerten

- Bewertung erfolgt durch paarweises Vergleichen aller n Requirements
- Die Vergleichswerte bilden eine Matrix $A = (a_{ij})$, wobei folgende Vergleichswerte verwendet werden:

Zahlenwert	Bedeutung in Worten
1	Gleich wichtig
3	Etwas wichtiger
5	Viel wichtiger
7	Sehr viel wichtiger
9	Uneingeschränkt wichtiger

- Die Werte 2,4,6,8 werden als Zwischenwerte eingesetzt, wenn Kompromisse nötig sind.
- Der zu a_{ij} umgekehrte Vergleich a_{ji} entspricht dem Kehrwert von a_{ij} , also $\frac{1}{a_{ij}}$

Eintrag in Zelle (i, j) heißt: Req_i ist „ (a_{ij}) “ als Req_j

Die entstehende Matrix wird in der Literatur zur Entscheidungstheorie und dem AHP als positive reciprocal matrix bezeichnet.

Die Spur der Matrix ist immer n . Im Idealfall (absolute konsistente Bewertungen) hat sie nur einen von 0 verschiedenen Eigenwert (Rang gleich 1). Dieser Eigenwert ist dann gleich n und der zugehörige Eigenvektor ist der gesuchte priority vector. Dieser Vektor ist der Perron-Vektor der Matrix.

Bsp: Wert $a_{ij} = 3$ heißt, Req_i ist „etwas wichtiger“ als Req_j

Cost-Value-Approach: Vorgehen

Schritt 1: Nutzen bewerten

- Aus der Vergleichsmatrix wird der **relative Wert** jedes Requirements bestimmt
- Der relative Wert eines Requirements ist der Anteil, den das Requirement am Gesamtwert der Requirementsmenge hat
- Der relative Wert der Requirements ergibt sich aus dem sogenannten **priority vector**
- Der priority vector ist der zum maximalen Eigenwert gehörende Eigenvektor

Schritt 1: Nutzen bewerten

- Den priority vector bestimmt man durch ein Näherungsverfahren:
 1. Alle Spalten normieren (Spaltensumme ist 1)
 2. Zeilensummen der normierten Matrix bestimmen
 3. Zeilensummen durch Anzahl der Requirements dividieren
 4. Der Ergebnisvektor ist der genäherte priority vector

Cost-Value-Approach: Vorgehen

Schritt 2: Konsistenz prüfen

- Bestimme **consistency index** und **consistency ratio**, um die innere Konsistenz der Vergleichsmatrix abzuschätzen
- Den consistency index berechnet man für eine Vergleichsmatrix A und priority vector p wie folgt:

1. Bilde Matrixprodukt $r = Ap$
2. Dividiere jedes Element von r durch das korrespondierende

$$\text{von } p: s = \begin{pmatrix} \frac{r_1}{p_i} \\ \vdots \\ \frac{r_n}{p_n} \end{pmatrix}$$

3. Bestimme maximalen Eigenwert von A : $\lambda_{max} = \frac{\sum_{i=1}^n s_i}{n}$
4. Der consistency index ist dann: $CI = \frac{\lambda_{max}-n}{n-1}$

Cost-Value-Approach: Vorgehen

Schritt 2: Konsistenz prüfen

- Die consistence ratio ist ein Maß für die Qualität der Matrix von paarweisen Vergleichen
- Zur Ermittlung benötigt man neben CI noch den passenden **random index**:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0,00	0,00	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

- Die consistency ratio ist dann: $CR = \frac{CI}{RI}$
- consistency ratio-Werte $\leq 0,1$ gelten als akzeptabel

Der random index für Matrixgröße n ist der mittlere consistence index für zufällig generierte reziproke Matrizen dieser Größe

In der Praxis kommen öfters auch Werte größer 0,1 vor, da muss man abwägen oder die Bewertungsmatrix verbessern.

Cost-Value-Approach: Beispiel

4 Requirements Req1 bis Req4 sollen priorisiert werden

Expertenbewertung lieferte folgende Vergleichsmatrix:

	Req1	Req2	Req3	Req4
Req1	1	1/3	2	4
Req2	3	1	5	3
Req3	1/2	1/5	1	1/3
Req4	1/4	1/3	3	1

Cost-Value-Approach: Beispiel

Normierung der Spalten ergibt:

	Req1	Req2	Req3	Req4
Req1	0,21	0,18	0,18	0,48
Req2	0,63	0,54	0,45	0,36
Req3	0,11	0,11	0,09	0,04
Req4	0,05	0,18	0,27	0,12

$$a'_{ij} = \frac{a_{ij}}{\sum_{l=1}^n a_{lj}}$$

Berechnung der Zeilensummen ergibt:

	Req1	Req2	Req3	Req4	Σ
Req1	0,21	0,18	0,18	0,48	1,05
Req2	0,63	0,54	0,45	0,36	1,98
Req3	0,11	0,11	0,09	0,04	0,34
Req4	0,05	0,18	0,27	0,12	0,62

Cost-Value-Approach: Beispiel

Für den priority vector ergibt sich:

$$\frac{1}{4} \begin{pmatrix} 1,05 \\ 1,98 \\ 0,34 \\ 0,64 \end{pmatrix} = \begin{pmatrix} 0,26 \\ 0,50 \\ 0,09 \\ 0,16 \end{pmatrix}$$

Der priority
vector

Das heißt, in unserem Beispiel repräsentiert Req1 26% des gesamten Werts der Requirements, Req2 50%, Req3 9% und Req4 16%

Cost-Value-Approach: Beispiel

Für λ_{max} ergibt sich:

$$\begin{pmatrix} 1 & 1/3 & 2 & 4 \\ 3 & 1 & 5 & 3 \\ 1/2 & 1/5 & 1 & 1/3 \\ 1/4 & 1/3 & 3 & 1 \end{pmatrix} \begin{pmatrix} 0,26 \\ 0,50 \\ 0,09 \\ 0,16 \end{pmatrix} = \begin{pmatrix} 1,22 \\ 2,18 \\ 0,37 \\ 0,64 \end{pmatrix}$$

$$\begin{pmatrix} 1,22/0,26 \\ 2,18/0,50 \\ 0,37/0,09 \\ 0,64/0,16 \end{pmatrix} = \begin{pmatrix} 4,66 \\ 4,40 \\ 4,29 \\ 4,13 \end{pmatrix}$$

$$\lambda_{max} = \frac{4,66 + 4,40 + 4,29 + 4,13}{4} = 4,37$$

Für den consistency index ergibt sich:

$$CI = \frac{4,37 - 4}{4 - 1} = 0,12$$

Für die consistency ratio ergibt sich:

$$CR = \frac{0,12}{0,90} = 0,14$$

Unsere Vergleichsmatrix ist also grenzwertig akzeptabel.

Cost-Value-Approach: Vorgehen

Schritt 3: Kosten bewerten

Bewertung der Kosten erfolgt auf gleiche Weise, wie die Bewertung des Werts der Requirements, nur auf der Basis einer paarweisen Vergleichsmatrix für die Herstellungskosten der Requirements, nicht deren Nutzen

Cost-Value-Approach: Vorgehen



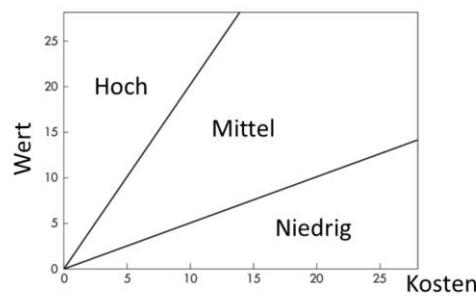
Schritt 4: Konsistenz prüfen

Ebenfalls auf gleiche Weise wie beim Wert der Requirements wird auch für die Vergleichsmatrix der Kostenwerte die consistency ratio bestimmt

Cost-Value-Approach: Vorgehen

Schritt 5: Ergebnisse darstellen

- Die relativen Werte und Kosten jedes Requirements werden in einem Kosten-Nutzen-Diagramm eingetragen
- Typischerweise entlang der x-Achse die relativen Kosten, auf der y-Achse der relative Wert



Cost-Value-Approach: Fortsetzung Beispiel

Expertenbewertung lieferte folgende Vergleichsmatrix
für die Kosten:

	Req1	Req2	Req3	Req4
Req1	1	7	1/2	4
Req2	1/7	1	1/5	2
Req3	2	5	1	5
Req4	1/4	1/2	1/5	1

Cost-Value-Approach: Fortsetzung Beispiel

Normierung der Spalten ergibt:

	Req1	Req2	Req3	Req4
Req1	0,29	0,52	0,26	0,33
Req2	0,04	0,07	0,11	0,39
Req3	0,59	0,37	0,53	0,42
Req4	0,07	0,04	0,11	0,08

Berechnung der Zeilensummen ergibt:

	Req1	Req2	Req3	Req4	Σ
Req1	0,29	0,52	0,26	0,33	1,41
Req2	0,04	0,07	0,11	0,39	0,39
Req3	0,59	0,37	0,53	0,42	1,90
Req4	0,07	0,04	0,11	0,08	0,30

Cost-Value-Approach: Fortsetzung Beispiel

Für den priority vector ergibt sich:

$$\frac{1}{4} \begin{pmatrix} 1,41 \\ 0,39 \\ 1,90 \\ 0,30 \end{pmatrix} = \begin{pmatrix} 0,35 \\ 0,10 \\ 0,48 \\ 0,07 \end{pmatrix}$$

Das heißt, in unserem Beispiel repräsentiert Req1 35% der gesamten Kosten der Requirements, Req2 10%, Req3 48% und Req4 7%

Cost-Value-Approach: Fortsetzung Beispiel

Für λ_{max} ergibt sich:

$$\begin{pmatrix} 1 & 7 & 1/2 & 4 \\ 1/7 & 1 & 1/5 & 2 \\ 2 & 5 & 1 & 5 \\ 1/4 & 1/2 & 1/5 & 1 \end{pmatrix} \begin{pmatrix} 0,35 \\ 0,10 \\ 0,48 \\ 0,07 \end{pmatrix} = \begin{pmatrix} 1,57 \\ 0,39 \\ 2,04 \\ 0,31 \end{pmatrix}$$

$$\begin{pmatrix} 1,57/0,35 \\ 0,39/0,10 \\ 2,04/0,48 \\ 0,31/0,07 \end{pmatrix} = \begin{pmatrix} 4,45 \\ 4,04 \\ 4,29 \\ 4,10 \end{pmatrix}$$

$$\lambda_{max} = \frac{4,45 + 4,04 + 4,29 + 4,10}{4} = 4,22$$

Cost-Value-Approach: Fortsetzung Beispiel

Für den consistency index ergibt sich:

$$CI = \frac{4,22 - 4}{4 - 1} = 0,07$$

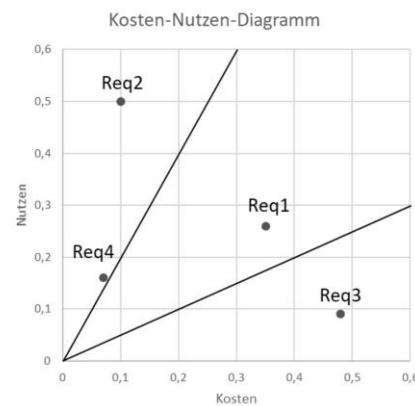
Für die consistency ratio ergibt sich:

$$CR = \frac{0,07}{0,90} = 0,08$$

Unsere Vergleichsmatrix ist also OK.

Cost-Value-Approach: Fortsetzung Beispiel

Es entsteht folgendes Kosten-Nutzen-Diagramm:



Abschnitt 1.6

VOLERE REQUIREMENTS ENGINEERING PROZESS

30.10.2018

Software Engineering 2

54

Was macht einen guten RE-Prozess aus?



Kundenorientierung



Vorgehen **zielgerichtet** und
methodisch klar



Einsatz **adäquater Mittel**



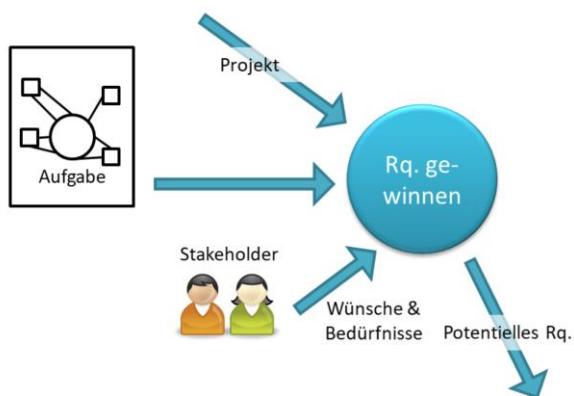
Integration von **Erstellung** und
Prüfung von Anforderungen

Das Volere Requirements Process Model I

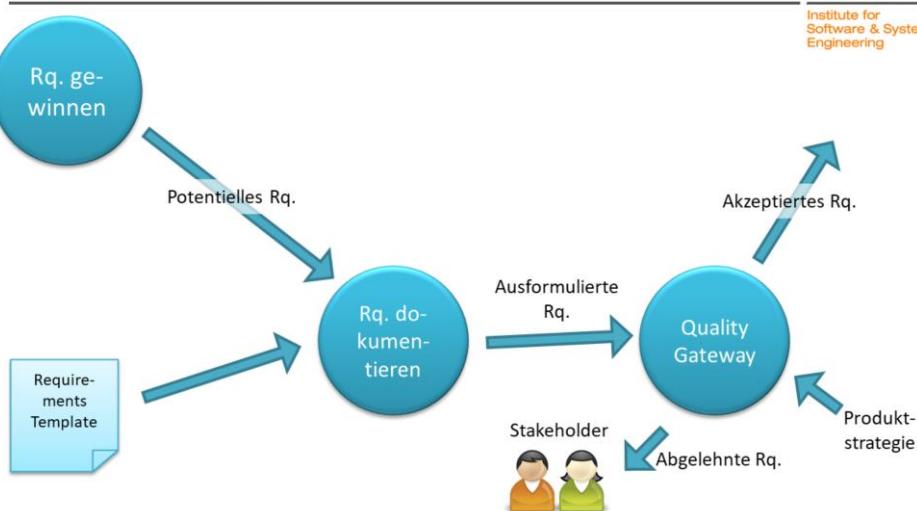


Volere Requirements Process: The Atlantic Systems Guild Ltd.
S. Robertson, J. Robertson: Mastering the Requirements Process, Addison-Wesley, 2006

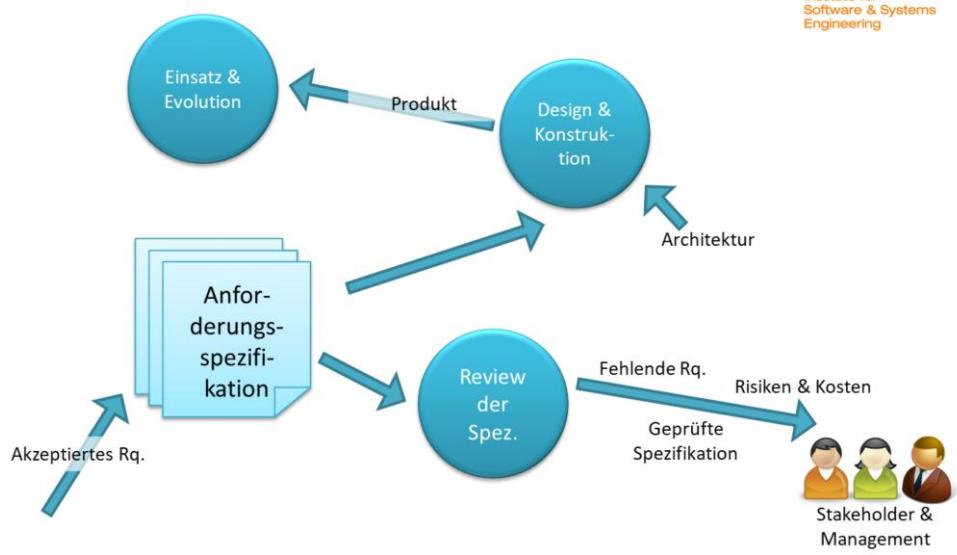
Das Volere Requirements Process Model II



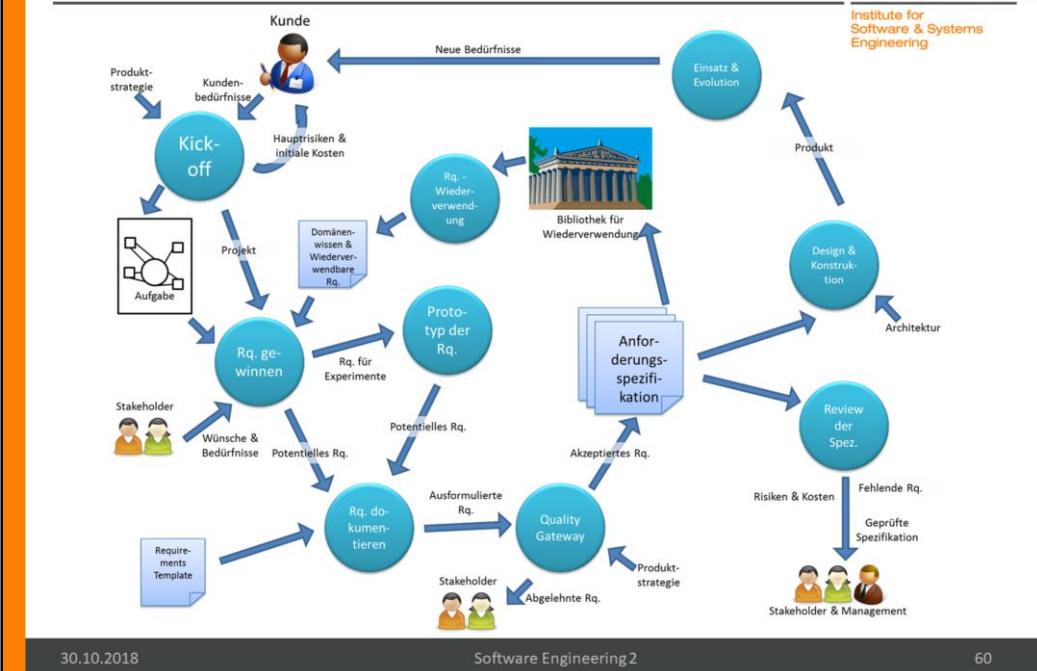
Das Volere Requirements Process Model III



Das Volere Requirements Process Model IV



Das Volere Requirements Process Model



Abschnitt 1.7

PROJEKTSTART

30.10.2018

Software Engineering 2

61

Fallstudie „Eisfrei“



- Eisfrei ist ein System, das vorhersagt, wann sich auf welchen Straßen Eis bildet und Streufahrzeuge disponiert, um die betroffenen Straßen mit Auftausalz zu behandeln.
- Dies ermöglicht der Straßenmeisterei genauere Vorhersagen, präzisere Einsatzpläne und macht so die Straßen sicherer.



Projekt-Kickoff

- Projekt vorbereiten und **Durchführbarkeit** sicherstellen
- Zusammenkunft der zentralen Stakeholder
- Es werden genügend Fakten gesammelt, um sicherzustellen, dass das Projekt
 - einen klar **definierten Umfang** hat,
 - ein **lohnendes Ziel** hat,
 - **durchführbar** ist und
 - die **Unterstützung der Stakeholder** hat.

Zentrale Stakeholder: Kunde, Hauptbenutzer, leitender Requirementsanalyst, technische Experten, Experten der Problemdomäne, andere mit erfolgsbestimmendem Input

Ergebnisse des Kickoff



30.10.2018

Software Engineering 2

64

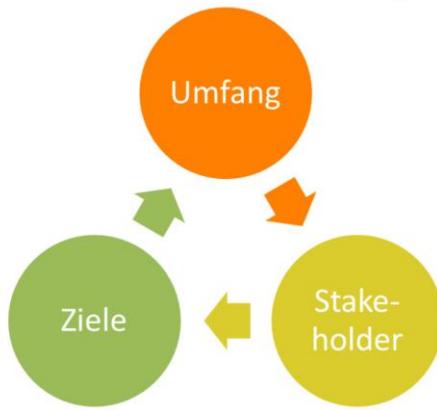
Constraints: Gibt es Randbedingungen/Lösungsvorgaben, die verwendet/beachtet werden müssen? Wie viel Zeit und Geld stehen zur Verfügung?

Terminologie: Begriffe und Bezeichner der Fachdomäne

Zusätzlich wird begonnen, relevante Fakten und Annahmen zu sammeln und u.U. wird ein einfacher low-fidelity Prototyp erstellt

Für das Projekt Eisfrei sieht es so aus: Salztechnik Systeme ist der Entwickler, Ziel ist ein weltweiter Absatz. Erster Kunde für das Projekt ist die Straßenmeisterei Ulm im Alb-Donau-Kreis. Vertreter dieser beiden Gruppen gehören zu den zentralen Stakeholdern.

- Drei Informationen sind entscheidend: Umfang, Stakeholder und Ziele



- Meistens beginnt man mit dem Umfang des Aufgabenbereichs

Der Aufgabenbereich (oder Arbeit (engl. „work“)) sind die Geschäftsaktivitäten, für die der Benutzer das Produkt braucht (kann kommerziell, wissenschaftlich oder technisch sein). Der Aufgabenbereich kann aktuell bereits automatisiert sein oder manuell oder eine Kombination von beidem, er kann auch im Moment noch gar nicht existieren.

Warum ist das wichtig? Nur wenn man den Aufgabenbereich verstanden hat, kann man ein Produkt bauen, das die Aufgabe unterstützen und dafür nützlich sein kann.

Der Umfang der Aufgabe steht in engem Zusammenhang mit einem zweiten wichtigen Konzept: Dem Kontext.

Kontextanalyse ist auch ein wichtiges Werkzeug, um den Aufbau des Domänenmodells/Glossars zu beginnen.

Abschnitt 1.8

KONTEXT

30.10.2018

Software Engineering 2

66





Vor nächster Folie: Beispiele für Kontext [->Enterprise-Application-Landscape, ->System-Architecture]

- Systeme stehen im Allgemeinen nicht allein. Sie sind eingebettet in eine (nicht-leere) Umgebung
- In der Umgebung befinden sich in der Regel Akteure, die für das System bedeutsam sind
- Die relevanten Akteure aus der Umgebung bilden den (System-)Kontext

Akteure können menschlich (z.B. Benutzer) oder technisch (anderes IT-System, Sensor,...) sein.

Software arbeitet oftmals in einem Kontext, der Rückwirkungen auf die Architektur hat. Dieser Kontext sind oft nicht nur andere Softwaresysteme, sondern z.B. auch Hardware oder mehr. Beispiel: Gesamtprojekt ist ein Strahlenmedizinzentrum, da gehört ein Zyklotronbeschleuniger dazu, Gesamtkosten 580 Mio. Euro, und es gilt herauszufinden, was von dem Ganzen für die Architektur wichtig ist.

Aufgabenbereich und Kontext bestimmen

- Durch das Festlegen des **Umfangs des Aufgabenbereichs** trennen wir
 - die Aufgabe, die wir studieren,
 - von
 - allen Aufgaben, die darum herum existieren.
- Die Grenze ergibt sich anhand von **Informationsflüssen**:
 - Fremde Information, die wir verarbeiten
 - Ergebnisse, die wir nicht selbst verarbeiten

Kontextmodell beim Kickoff durch Diskussion als erste Skizze am Whiteboard.

Warum ist das wichtig? Nur wenn man die Aufgabe verstanden hat, kann man ein Produkt bauen, das sie unterstützen und dafür nützlich sein kann.

Die Informationsflüsse, die in einen Aufgabenbereich hineinkommen, und die Informationsflüsse, die ihn verlassen, definieren den Umfang des Aufgabenbereichs. Der Aufgabenbereich ist letzten Endes die Transformation der eingehenden Informationen in die ausgehenden Informationen.

Die Festlegung des Umfangs bestimmt, welchen Aufgabenbereich wir im Folgenden studieren und welche Aufgabenbereiche wir nicht untersuchen.

Das spätere Produkt wird einen wesentlichen Teil des Aufgabenbereichs übernehmen (nicht notwendigerweise alles)

Die Aufgabenbereiche, die wir nicht untersuchen, liegen außerhalb des Umfangs unseres Aufgabenbereichs, dies sind die angrenzenden Systeme. Für diese genügt es, die Details der Verbindung zu unserer Aufgabe zu verstehen

Zur Fallstudie Eisfrei: Der Aufgabenbereich von Eisfrei ist, die Zeit vorherzusagen, wann sich Eis auf einer Straße bildet, so dass ein Streufahrzeug eingeteilt werden kann, um die Straße zu behandeln – möglichst zeitnah zum Gefrierzeitpunkt – und so die Eisbildung zu vermeiden. Um das zu tun, braucht Eisfrei Daten über das Wetter und die Bedingungen auf den Straßen. Es erhält Daten von Sensoren an einer Messstation für atmosphärische Bedingungen und Temperatur und der Wassermenge auf der Straßenoberfläche. Der Aufgabenbereich der Messstation ist wiederum mit einem anderen Aufgabenbereich verknüpft, dem Wetter. Das Wetter ist mit der Neigung der Erdachse während des Umlaufs um die Sonne verknüpft. Die Sonne ist mit dem Aufgabenbereich der Galaxis verknüpft.

Erster Schritt: Relevante Problemdomänen

- Domänen sind Fachgebiete
- Relevante Domänen sind Fachgebiete, über die wir etwas wissen müssen
- Zu Eisfrei erzählt uns der Kunde:

Straßen überfrieren im Winter und Eis auf der Straße verursacht Unfälle. Wir müssen in der Lage sein, vorherzusagen, wann sich Eis bildet, so dass ein Streufahrzeug eingeplant werden kann, um die Straße rechtzeitig zu behandeln. Wir erwarten, dass ein neues System genauere Vorhersagen der Eislage liefert. Dies ermöglicht zeitnäheres Anwenden von Aufbaumaterial als aktuell möglich, wodurch Unfälle reduziert werden. Außerdem wollen wir willkürliche Behandlungen von Straßen eliminieren, welche Auftausalz verschwenden und Umweltschäden verursachen.

Solche Fachgebiete können sehr groß sein: „Bankwirtschaft“, „Versicherungen“, „Wetter“. Typischerweise ist eine Domäne inhaltlich Teil einer größeren.

Die Subjekte im Beispieltext zeigen, welche Domänen relevant sind.

Für Eisfrei finden wir 4 relevante Domänen: Straßen, Planung/Scheduling, Fahrzeuge und Wetter

Als nächstes gilt es, die relevanten Domänen auf diejenigen ihrer Teile einzuschränken, die wir genauer analysieren müssen. Es ist nicht praktikabel und kosteneffizient alles bis zurück zum Ursprung des Universums zu untersuchen. Wir halten diese Einteilung als Kontextmodell fest, das unseren Aufgabenbereich zusammen mit ihren Verbindungen zu den angrenzenden Systemen zeigt.

Schritt zwei: Angrenzende Systeme in den relevanten Problemdomänen

- Alle relevanten Problemdomänen zu einem Teil unserer Aufgabe zu machen, ist nicht zielführend (und oft einfach unmöglich)
- Stattdessen suchen wir in jeder Problemdomäne nach physischen Entitäten (Personen oder Institutionen/Systeme) die Teilespekte der Problemdomäne repräsentieren
- Diese Entitäten sind potentielle angrenzende Systeme
- Für jeden Kandidaten ist zu prüfen, ob er Teil unseres Aufgabenbereichs ist, oder Teil einer anderen Aufgabe

Die gefundenen Entitäten sind zuerst nur potentiell angrenzende Systeme, dies muss näher untersucht werden.

Nicht alle Domänen haben eine physische Präsenz. Insbesondere Domänen, die Policies/Geschäftsstrategien repräsentieren. Diese müssen dennoch im Kontextmodell vertreten sein.

Für Eisfrei relevant ist das Wetter. Wolken und Regen sind physische Entitäten dieser Domäne, mit denen wollen wir uns aber nicht auseinandersetzen, da zu weit weg von unserem Aufgabenbereich. Stattdessen gibt es einen [Wetterdienst](#), der uns mit Informationen zum Wetter versorgen kann. Die Wettervorhersage ist aber nicht präzise genug, um Eisbildung adäquat vorherzusagen. [Wetterstationen](#), liefern solche genaueren Daten und können diese an Eisfrei übermitteln. Wetterstationen sind aber teuer, daher kann die Stadt Ulm nicht tausende von ihnen aufstellen. Von der Straßenmeisterei Ulm erfahren wir, dass [Anbieter von Temperaturkarten](#) Informationen über die Temperaturunterschiede auf jedem Meter Straße zwischen den Messstationen liefern können.

Die Kundenvertreter sind wesentlich für das Identifizieren der wichtigen Entitäten in den relevanten Fachdomänen.

Jeder dieser Entitäten könnte ein angrenzendes System sein. Dies muss aber genau untersucht werden. Im Fall des Beispiels wollen wir nicht selber das Wetter vorhersagen, daher betrachten wir das als außerhalb unseres Aufgabenbereichs. Die beiden anderen Entitäten werden ebenfalls als angrenzende Systeme betrachtet, da Salztechnik Systeme, der Hersteller von Eisfrei, sie nicht besitzt und sie auch nicht verändern oder anpassen möchte.

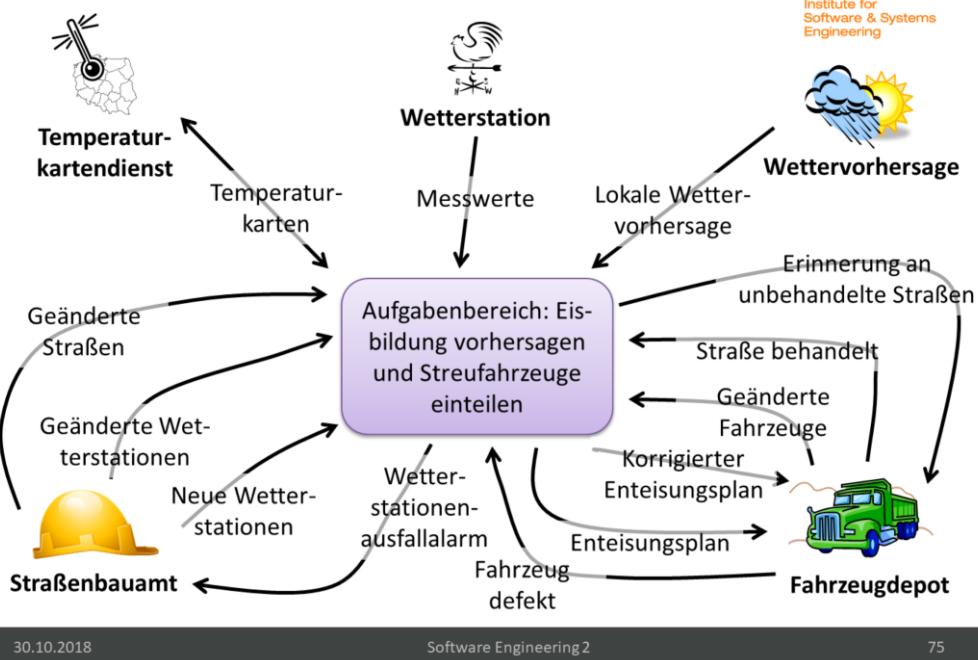
Aus den weiteren relevanten Problemdomänen finden wir weitere angrenzende Systeme:

- Die Domäne Straßen wird für uns durch das **Straßenbauamt** repräsentiert. Dieses baut und wartet Straßen und kann uns Informationen über Straßen liefern.
- Von Vertretern des Kunden erfahren wir außerdem, dass das **Fahrzeugdepot** Streufahrzeuge wartet und als Basis für deren Einsatz dient. Dies ist für uns ein angrenzendes System, das die Domäne Fahrzeuge repräsentiert.

Es ist unwahrscheinlich, dass wir den Wunsch oder die Notwendigkeit haben, Einfluss auf den Bau von Straßen zu nehmen, daher ist dies naheliegenderweise ein angrenzendes System und nicht Teil unserer Aufgabe.

Eine der Hauptaufgaben von Eisfrei ist das Erstellen von Einsatzplänen für Streufahrzeuge, daher erwarten wir zurecht Informationsflüsse vom Fahrzeugdepot zu unserem Aufgabenbereich (z.B. Fahrzeug kaputt) und in die andere Richtung (neuer Einsatzplan).

Erster Entwurf eines Kontextdiagramms



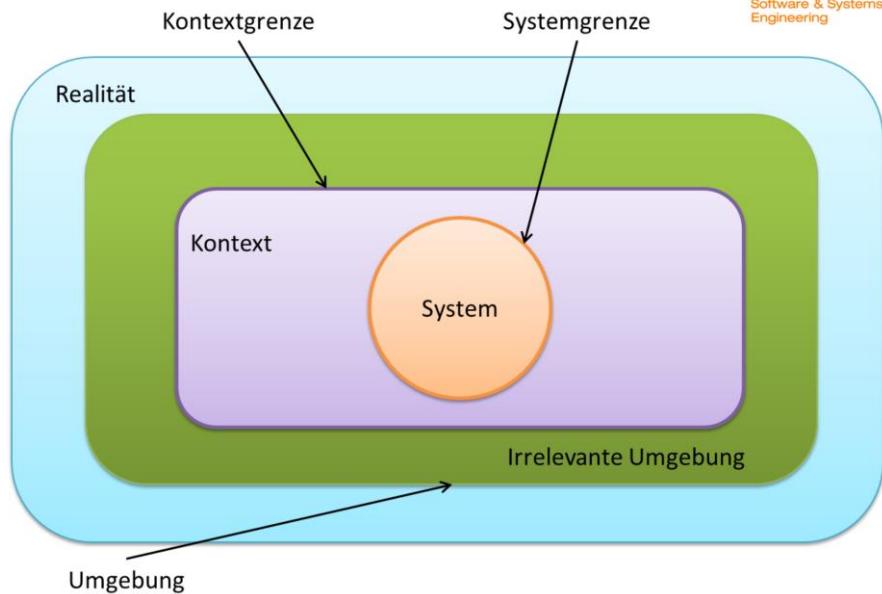
30.10.2018

Software Engineering 2

75

Es ist ein Fehler, jetzt schon ein rein produktzentriertes Kontextmodell zu erstellen, das nur das intendierte/antizipierte Produkt enthält. Es geht jetzt aber um das Verständnis der gesamten Aufgabe, so dass später entschieden werden kann, welcher Teil davon automatisiert (ein Teil des Produkts) wird und welcher nicht. Wenn man sich jetzt schon auf das konzentriert, was man denkt, dass es das Produkt werden wird, liefert man möglicherweise am Ende ein weniger nützliches Produkt, als es möglich wäre. Wenn sich im Aufgabenbereich keine Menschen mehr befinden, ist der Umfang des Aufgabenbereichs wahrscheinlich zu eng gefasst.

Kontext- und Systemgrenze



Realität

Alle materiellen oder immateriellen Dinge der Welt.

Umgebung

Der Teil der Realität, der auf ein in diesem Bereich eingesetztes System Einfluss haben könnte.

Kontext

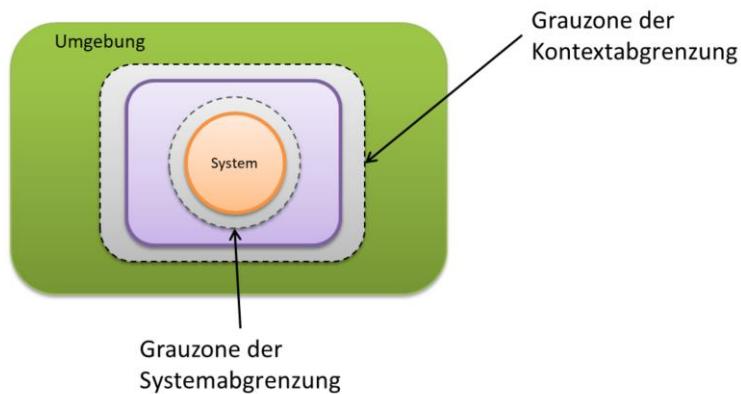
1. Der umgebende Teil einer sprachlichen Einheit. 2. Der inhaltliche (Gedanken-, Sinn)zusammenhang, in dem eine Äußerung steht. 3. (In der Informatik) Die Komponenten eines Anwendungsbereichs (Umgebung), die mit einem System interagieren, aber selbst nicht Teil des Systems sind.

Akteur

Ein **Mensch**, der **Ziele** hat und diese verfolgt oder ein **Element der Umgebung**, das zur Erreichung eines Ziels dient und hierzu handelt oder Informationen verarbeitet.

Dudenverlag: Duden Das Fremdwörterbuch, 2007

Kontext- und Systemgrenze können sich im Laufe des Requirements Engineerings verschieben



In der früheren Phase des Requirements Engineering ist insbesondere die **Systemgrenze noch gänzlich unbekannt**. Wir versuchen unsere Aufgabe als ganzes zu begreifen und erst später für die verschiedenen **Business Use Cases** zu entscheiden, welcher Teil des Use Cases automatisiert wird (und somit zu einem Teil des Systems) und welcher Teil nicht durch unser Produkt abgedeckt wird.

Beispiel Online-Datenbank: Im Laufe des RE stellt sich heraus, dass Datenschutzbestimmungen nicht beachtet werden müssen, da die vom System verarbeiteten Daten anonymisiert sind. Was wird durch diese Erkenntnis beeinflusst? Warum?