

Abschnitt 1.4

ZIELE, ANFORDERUNGEN UND SZENARIEN

Ziel

Bezeichnet einen in der Zukunft liegenden, gegenüber dem Gegenwärtigen im Allgemeinen veränderten, erstrebenswerten oder angestrebten Zustand.

Anforderung

1. Eine Bedingung oder Fähigkeit, die von einem Benutzer zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die ein System erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Fähigkeit nach (1) oder (2).

Wikipedia: <http://de.wikipedia.org/wiki/Ziel>

Ziele sind der Grund/die Ursache, warum ein Projekt durchgeführt/eine Veränderung angestrebt wird. „Warum machen wir das/wollen wir das?“

Ziele formuliert jemand im Allgemeinen, um für sich ein Bedürfnis zu erfüllen

Anforderungen sind die zu erreichenden Eigenschaften/Fähigkeiten, die das neue System haben muss, um die Ziele zu erreichen.

Etwas plakatives Beispiel: Ziel für eine neue Webseite: „Man findet leicht was man sucht“, daraus resultierende Anforderungen: „Suchfunktion“, „Seitenübersichten“

Beispiel

Bedürfnis: Mehr Geld verdienen

Ziel: Umsatz verdoppeln

Anforderung: E-Mails mit kundenspezifischen Einkaufsvorschlägen verschicken können.

- Ziel und Anforderung beschreiben beide etwas **zu Erreichendes**
- Begriffe sind aber nicht synonym
 - Ziel
 - Ein zu erreichender **Zustand**
 - Globalere oder abstraktere Sicht
 - Anforderung
 - Eine zu erreichende **Eigenschaft**
 - Konkreter
 - Meist eine zur Erreichung eines Ziels **notwendige Bedingung** oder **Eigenschaft**

Es kommt hinzu, dass Ziele auf einer Ebene oft (den Rahmen für) Anforderungen auf einer tieferliegenden Ebene vorgeben

Beispiel: Geschäftsebene -> Prozessebene -> Softwaresystemebene

[-> Ziele-Anforderungen-Kontinuum]

Vor nächster Folie: Tafelbild Arten von Zielen [->Arten-von-Zielen]

Frage: Wer kennt noch den Begriff Szenario?

Dann beschreiben: Szenarien + wie beschreibt man diese

Szenario

Beschreibt die Erfüllung oder Nichterfüllung von Zielen und somit die Stakeholder-Intentionen anhand eines konkreten Beispiels. Ein Szenario enthält typischerweise eine Folge von Interaktionsschritten mit dem System.

- Szenarien sind zentrales Element bei der Beschreibung von Geschäftsvorfällen (**Use-Cases**, ursprünglich aus Jacobsons OOSE-Ansatz)
- Heute zentraler Bestandteil von UML und RUP

[->Use-Case]

Wir unterscheiden im folgenden zwei Arten von Use-Cases: **Business Use-Cases** und **Produkt Use-Cases**. Die Use-Cases, die im Unified Process benutzt werden, sind Product Use-Cases. Für das Requirements Engineering ist es besser, **nicht sofort mit dem zu erstellenden System im Kopf an die Problemdomänen heranzugehen, sondern zuerst die zu erledigende Aufgabe konzeptuell**, also aus Sicht der Geschäftsdomäne, zu begreifen.

Use-Case Szenarien spielen eine große Rolle bei der Anforderungsanalyse und –beschreibung

Zusammengefasst:

Ziele: Was soll erreicht werden, oft eher abstrakt/visionär

Szenarien: Schon konkreter: Wie sehen die geschäftlichen Abläufe aus, anhand konkreter „Beispielabläufe“

Szenarien meint hier nicht die sehr formalisierte Beschreibung der Interaktion mit dem System, wie sie in Use-Cases verwendet wird, sondern einfach konkrete Geschichte, die beispielhaft erläutern, was benötigt wird.

Anforderungen: Konkrete und detaillierte Eigenschaften/Fähigkeiten/Verhalten, die nötig sind, um die Schritte des Szenarios zu realisieren.

Anforderungen werden oft auch präziser als atomare Anforderungen bezeichnet, da sie ein elementares Stück Funktionalität oder eine spezifische Eigenschaft/Fähigkeit beschreiben, die das System später haben muss.

Eine Menge von atomaren Anforderungen realisiert später einen Schritt eines Szenarios, mehrere solche Schritte zusammen bilden ein mögliches Szenario eines Use-Cases

Nach dieser Folie Beispiel 2CV [->Anforderung-2CV-Beispiel-für-unterschiedliche-Anforderungsartefakte, ->2CV-Anforderungen-was-ist-was]



26.10.2018

Software Engineering 2

27

Abschnitt 1.5

PRIORISIERUNG

Prioritäten setzen

- Einfache Priorisierungsstrategie [IEEE 830]:
 - **Muss:** Unverzichtbare Anforderung
 - **Soll:** Wichtig, aber bei zu hohen Kosten verzichtbar
 - **Wunsch:** Wäre nett, aber nicht essentiell

Priorisierung ist

- nötig bei
 - harten Kostengrenzen
 - Beschaffung
- wünschenswert bei
 - Festlegung der Inkremente bei inkrementeller Entwicklung
 - Releaseplanung bei der Weiterentwicklung bestehender Systeme

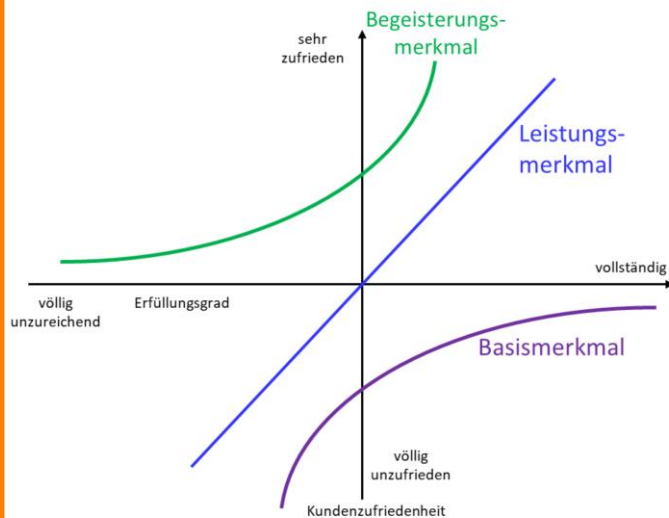
Institute of Electric and Electronic Engineers: IEEE Recommended Practice for Software Requirements Specifications (IEEE Std. 830-1998)

Die IEEE830 Klassifikation führt in der Praxis oft zu einer Häufung von Muss- und praktisch keinen Wunsch-Anforderungen. Kano daher eine bessere Alternative

Priorisierung wird im Kapitel zu Agile Methoden wieder eine Rolle spielen.

Nach dieser Folie: Kano-Modell an der Tafel machen [->Kano]

Entwickelt von Noriaki Kano 1978 an der Uni Tokio



Außerdem:

- **Unerhebliche Merkmale:**
Verursachen weder Zufriedenheit, wenn vorhanden, noch Unzufriedenheit, wenn nicht vorhanden
- **Rückweisungsmerkmale:**
Ablehnung des Produkts, wenn Merkmal vorhanden

Das Kano-Modell: Messung der Erwartungshaltung

- Messung der Erwartungshaltung mit Fragebogen
- Für jedes Produktmerkmal zwei Fragen: Einmal positiv (funktional) einmal negativ (dysfunktional)
- **Positive Frage:** Was würden Sie davon halten, wenn das Produkt über die Eigenschaft X verfügt?
- **Negative Frage:** Was würden Sie davon halten, wenn das Produkt NICHT über die Eigenschaft X verfügt?
- Beantwortung mit einer von fünf vordefinierte Antworten:
 - Das würde mich sehr freuen (1)
 - Das setze ich voraus (2)
 - Das ist mir egal (3)
 - Das nehme ich gerade noch hin (4)
 - Das würde mich sehr stören (5)

Beispiel:

Positiv: Was würden Sie davon halten, wenn der Aufzug über eine Funktion zur Priorisierung der Stockwerke verfügt?

Negativ: Was würden Sie davon halten, wenn der Aufzug NICHT über eine Funktion zur Priorisierung der Stockwerke verfügt?

Das Kano-Modell: Messung der Erwartungshaltung

Kundenanforderung		Dysfunktional				
		(1)	(2)	(3)	(4)	(5)
Funktional	(1)	W	Be	Be	Be	L
	(2)	R	U	U	U	Ba
	(3)	R	U	U	U	Ba
	(4)	R	U	U	U	Ba
	(5)	R	R	R	R	W

Be = Begeisterungsmerkmal

L = Leistungsmerkmal

Ba = Basismerkmal

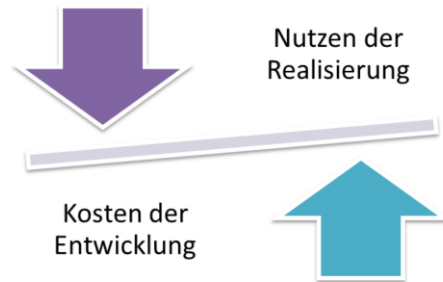
U = Unerhebliches Merkmal

R = Rückweisungsmerkmal

W = Widerspruch

Übung mit den Studierenden: Feature „Aufladung der Mensakarte über Web-Seite mit elektronischer Bezahlung (PayPal, EC-Abbuchung,...)“

Wertbasierte Priorisierung



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]