Technische Hochschule Wildau
Technical University of Applied Sciences

Skript zur Vorlesung

Grundlagen der Wirtschaftsinformatik

Prof. Dr. Mathias Walther

Wintersemester 2021/22

Inhaltsverzeichnis

4 Softwareeinführung und Systementwcklung

4.1 Betriebsmodelle

Ausgangssituation

1

Software im Unternehmen:

- Betriebssysteme
- Infrastruktur (bspw. Authentifizierungs-Server, Backups)
- Generische Anwendungssoftware (Excel)
- Spezifische Anwendungssoftware (CRM, ERP,...)

Früher:
Unternehmen kaufen teure Hardware und lassen individuelle Software entwickeln

Heute:
Hardware ist günstig/austauschbar. Trend zu Standardsoftware + Customizing

Aufgaben

2

Wer ist verantwortlich für?

- (Weiter-)Entwicklung der Software?
- Bereitstellung der Hardware?
- Installation (+ Einspielen von Updates)?
- Backups?

Klassische Aufgaben einer IT-Abteilung:

- PC-Support und User-Help-Desk
- Netzwerkmanagement
- Rechenzentrumsbetrieb
- Anwendungsbetreuung (bspw. Backups/Updates)
- Anwendungsentwicklung (Individualsoftware)

$\Rightarrow$ On-Premise-Betrieb von Hard- und Software war bis ca. 2010 der Normalfall

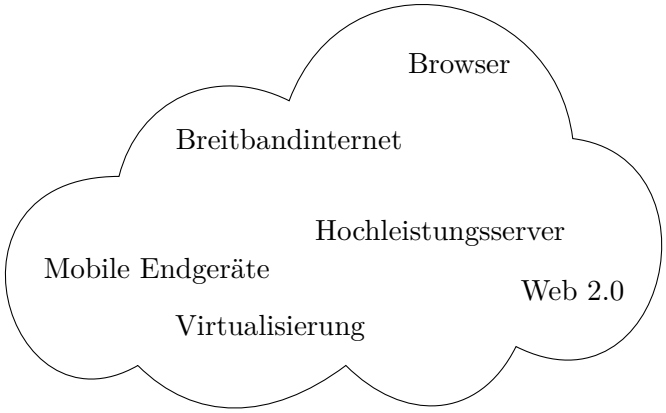
Software-as-a-Service (SaaS) 3

Bereitstellung lauffähiger Software als Geschäftsmodell

Definition:

- Bereitstellung lauffähiger Software durch einen Anbieter (meist der Softwarehersteller)
- Abgerechnet wird eine (meist monatliche) Miete, die alle Aufwände zum Betrieb abdeckt (bspw. Hardwarekosten, Lizenzen/Wartungsgebühren, Backups, Einspielen von Updates, User Help Desk...).

Cloud-Computing 4



There is no cloud. It‘s just someone else‘s computer

4.2 Lizenzmodelle

Lizenzen für Unternehmenssoftware 5

Lizenzmodelle für Standardsoftware orientieren sich an:

- Größe des Unternehmens
- Kosten vergleichbarer Lösungen im Markt
- Einsparpotentiale/Produktivitätsgewinne:
  - Nutzungsintensität
  - Anzahl der Nutzer
  - Installationen/Hardwarenutzung

  - Geschäftsvorfälle
- Umfang der Lösung (verfügbare Features)

Kosten für Individualsoftware orientieren sich typischerweise am Aufwand für Erstellung & Wartung

Softwarelizenzen

6

Grundlegende Bestimmungen:

- Das Urheberrecht bleibt immer beim Urheber!
- Der Urheber einer Software darf über die Verwendung bestimmen und die Nutzung lizensieren.
- Angestellte/Beamte lizensieren automatisch die Software an den Arbeitgeber/Dienstherr die im Rahmen der Arbeit programmiert wird (§69 UrhG)

Anderen werden Nutzungsrechte eingeräumt:

- Allgemeines Nutzungsrecht an der Software (ggf. beschränkt auf Nutzer, Geräte, Art der Nutzung usw.)
- Recht auf Verbindung („Linking“) der Binärdateien
- oder des Quellcodes mit anderer Software
- Recht auf Weiterverbreitung der Software (Binärdateien)
- Recht auf Veränderung der Binärdateien
- Recht auf Veränderung des Quellcodes, sofern vorhanden
- Recht auf Weiterverbreitung veränderter Versionen der Binärdateien oder des Quellcodes

Open-Source-Software-Lizenzen

7

Definitionen:
Laut der Open Source Initiative sind folgende Aspekte für Open Source Lizenzen geregelt:

„freely used, modified and shared“

Wichtige Eigenschaften:

- freie Nutzung/Verbreitung

⇒ keine Kosten, keine Einschränkungen bzgl. Nutzergruppen

- Zugang zu Quellcode (menschenlesbar)
- Veränderung und Weiterverbreitung erlauben

https://opensource.org/

Typen von Open-Source-Lizenzen

8

Permissive Licenses:

- Mach damit was du willst, aber der Urheber übernimmt keine Verantwortung
- *abgeleitete Software kann unter anderer Lizenz stehen*
- teilweise gibt es Werbeklauseln (erwähne den Urheber)
- z. B.: MIT-, BSD-, Apache-License

Copyleft-Licenses:

- abgeleitete Software muss unter der gleichen (oder einer kompatiblen) Lizenz stehen
- z. B.: GPL Lizenzfamilie

Open Source Software in Unternehmen

9

Nutzen als Anwender:

- Kostenfreie Alternative(n) nutzen, selbst Support leisten oder Support mieten
- z. B.: Linux-Distributionen, Firefox Webbrowser, Apache Webserver, ...

Nutzen als Teil/Basis des eigenen Produktes:

- Einbinden/Erweitern von Open-Source-Programmen
- z. B.: plan.io bietet eine SaaS-Projektmanagement-Lösung basierend auf dem Open-Source-Projekt redmine.org

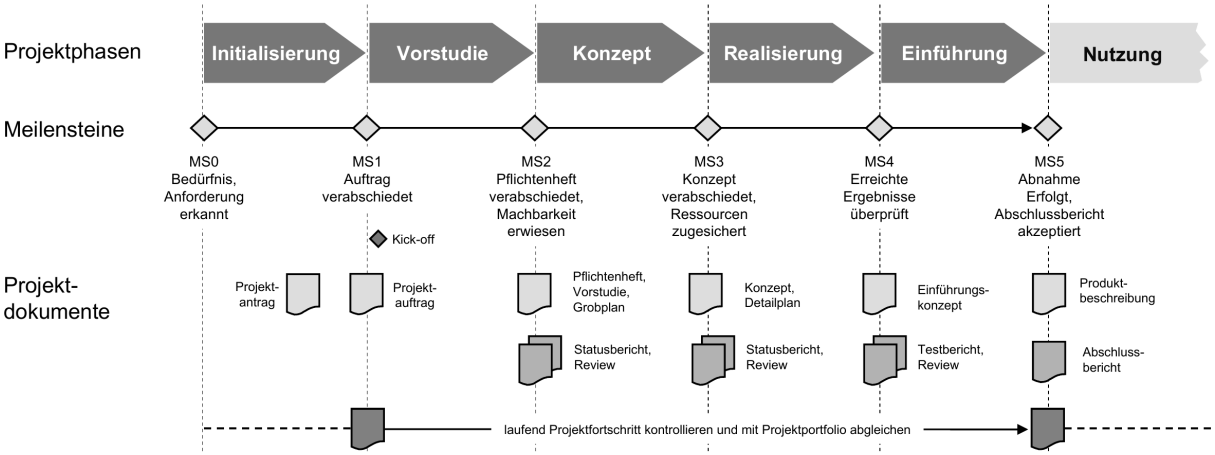
Eigene Produkte unter Open Source Lizenz stellen:

- Akzeptanz, Sichtbarkeit, Marktdurchdringung erhöhen
- z. B.: Camunda Services GmbH veröffentlicht seine Software als Open-Source, zahlende Kunden bekommen frühere Updates, mehr Features und 24/7-Support

4.3 Softwareprojekte

Ablauf eines Software-(Enwicklungs/Einführungs)-Projekts

10



Quelle: Kuster, Huber u. a., 2008

⚠ Genauer Ablauf und Inhalt der Phasen wird durch ein Vorgehensmodell festgelegt!

Initialisierungsphase

11

- meist unstrukturierte Phase
- Problemstellung kann entweder schon relativ konkret formuliert sein oder lediglich aus vagen Vermutungen bestehen
- Vorarbeiten sollten in Projektvereinbarung resultieren:
  - Globalziel und Projektpriorität
  - grobe Aufgabenstellung und Vorgehensweise
  - Verantwortlichkeiten und Rollen
  - Ressourcenzuteilung
- Erfassen des Projektauftrags:
  - Anforderungen aufnehmen: Was soll realisiert werden?
  - Projektorganisation festlegen
  - Stakeholder identifizieren und analysieren
  - Risiken identifizieren und Maßnahmen zur Reduktion der Risiken entwickeln
  - Projekt strukturieren und grob planen

Vorstudienphase

- entscheidende Phase ⇒ viel Zeit und Aufwand
- feststellen der realistische Durchführung der Problembearbeitung
- Analyse des Problems
  - Definition des Problems
  - Identifizierung der Ursachen
  - Spezifizierung der Lösung
  - Identifizierung der Anforderungen

  } *Systemanalyse*
- Machbarkeitsstudie
- Diskussion von Lösungsrichtungen
- Auswahl einer Vorgehensvariante
- weiteres Vorgehen planen: Projektorganisation, Terminplan, Ressourcen, Methoden
- wichtige Erkenntnisziele:
  - Zusammenhänge und Mechanismen im Problemfeld
  - Grenzen des Problems
  - Anforderungen an die Lösung
  - alternative Lösungsprinzipien
  - Projektrisiken

Konzeptphase

- Gesamtkonzept mit Lösungsvarianten auf der Basis des gewählten Lösungsprinzips bzw. Rahmenkonzepts aus der Vorstudie entwickeln
- Beurteilung von Zielerreichung, Funktionstüchtigkeit, Zweckmässigkeit und Wirtschaftlichkeit
- Aufgaben:
  - Rahmenplan (Meilensteinplan, Masterplan) für die nächsten Phasen erstellen
  - Einsparungsmöglichkeiten aufzeigen (Parallelisierung von Tätigkeiten)
  - Definition von Teilprojekten
  - Investitionsentscheidungen formulieren
- Detailplanung der Lösung (Grobkonzept ⇒ *Systementwurf*)

- einzelne Teillösungen so weit konkretisieren, dass sie anschließend umgesetzt werden können
- Schulungs- und Einführungskonzept erstellen
- Ergebnis der Konzeptphase ist die Entscheidung für eine Lösungsvariante

Realisierungsphase 14

- im weitesten Sinne Realisierung des Projekts
- läuft teilweise parallel zur Konzeptphase ab
- Arbeiten:
  - Software erstellen
  - benutzerfreundliche Dokumentation bzw. Bedienungsanweisung erstellen
  - organisatorische Regelungen (Information, Störungen usw.) festlegen
  - Wartungsorganisation, Instandhaltungskonzepte, usw. festlegen
  - Projektmarketing betreiben, d. h. Beteiligte und externe Stellen (z. B. Kunden) informieren
  - Sachmittel, personelle und finanzielle Mittel bereitstellen
  - Ausbildung der künftigen Benutzer planen
  - Projektablauf steuern
  - Plan/Ist-Vergleiche durchführen (finanzielle Führung, Controlling)
  - Abweichungen kommunizieren

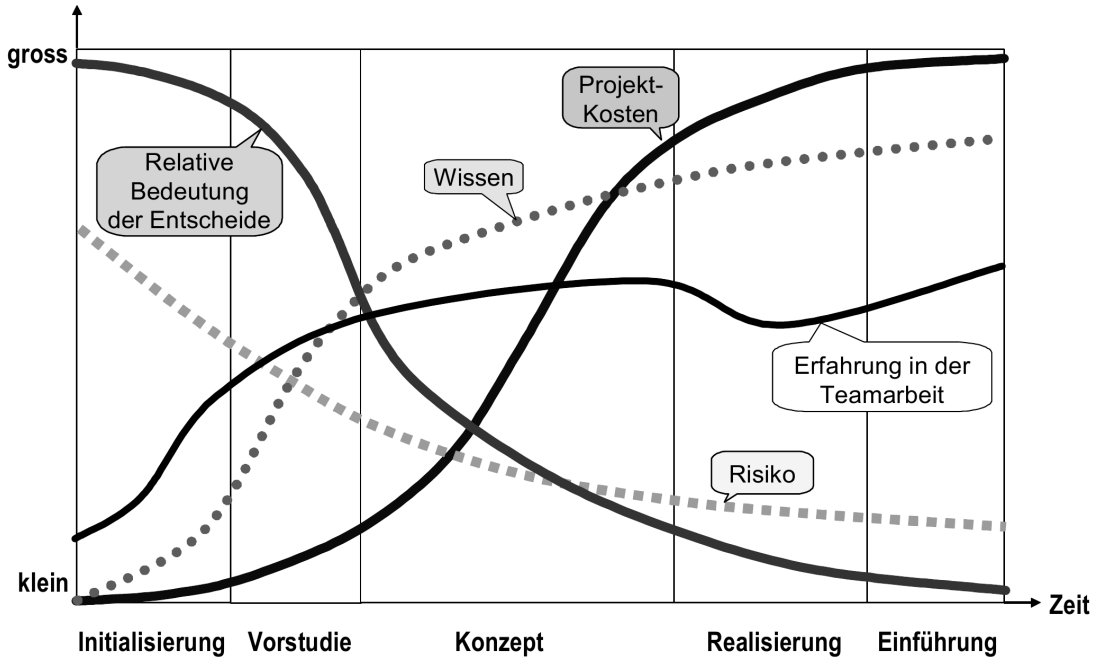
Einführungsphase 15

- Einführung:
  - nur relativ kleine und einfache Lösungen können ohne großes Risiko als Ganzes eingeführt werden
  - komplexe Systeme tendieren zu nicht kalkulierbaren Nebenerscheinungen
  - Einführungsphase oft langwierig
  - stufenweise Einführung
  - Big-Bang-Ansatz
- Übergabe:
  - Erfolg hängt davon ab, wie Wissenstransfer gelingt
  - Anwender oder Benutzer genügend schnell und umfassend schulen
- Abschluss

- Projektabschlussbericht: Ergebnisse, Aussagen zum Projektprozess
- administrative Abschlussarbeiten
- Schlussabrechnung
- Abnahmeprotokoll

Tragweite von Entscheidungen

16



Quelle: Kuster, Huber u. a., 2008

Kontrollfragen

1. Welche Aussagen sind richtig?

☐ Das Projekt abzubrechen kann eine Entscheidung eines Meilensteins sein.
☐ Große Projekte sollten immer mit dem Big-Bang-Ansatz eingeführt werden.
☐ Die Definition von Teilprojekten ist eine Aufgabe der Vorstudie.
☐ In der Vorstudie werden Projektrisiken ermittelt.
☐ An die Vorstudie schließt sich die Initialisierungsphase an.

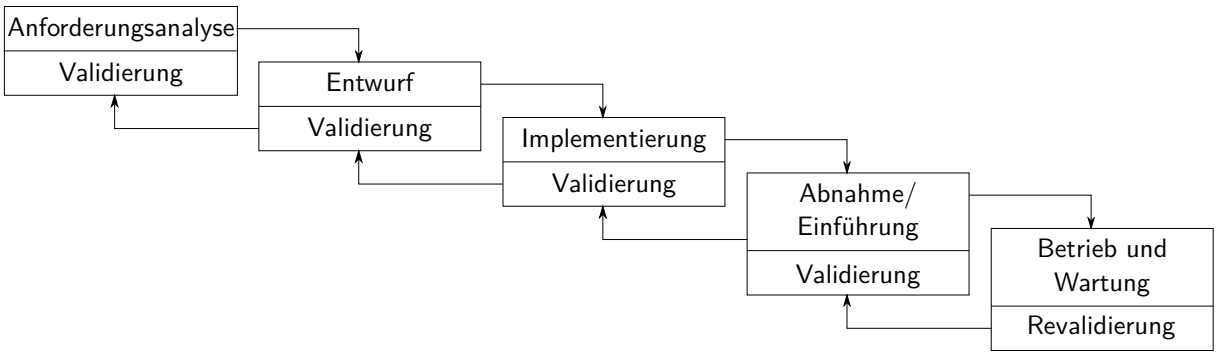
2. Welche Phasen sollten in welcher Reihenfolge bei einem Informatik-Projekt durchgeführt werden?

☐ Initialisierung, Vorstudie, Konzept, Realisierung
☐ Vorprojekt, Projektantrag, Vorbereitung zur Ausführung, Ausführung
☐ Vorstudie, Konzept, Initialisierung, Realisierung

Vorgehensmodelle allgemein

- organisieren einen Prozess der gestaltenden Produktion in verschiedene, strukturierte Abschnitte
- sind organisatorische Hilfsmittel, die für konkrete Aufgabenstellungen individuell angepasst werden können
- vereinheitlichen Maßnahmen
- dienen zur Ableitung konkreter Maßnahmen
- dienen dem Qualitätsmanagement
- Ausprägungen: Lineare Vorgehensmodelle, iterative Vorgehensmodelle und Prototyping Modelle
- Vorgehensmodelle zur Softwareentwicklung
  - Wasserfallmodell
  - V-Modell
  - Spiralmodell
  - Scrum

Wasserfallmodell



- klassisches sequentielles Vorgehensmodell in der Softwareentwicklung
- zu Beginn der Phase sind Anforderungen umfassend bestimmt
- zum Abschluss einer Phase werden die Ergebnisse verifiziert und validiert

Phasen (I)

Anforderungsanalyse:

- Aufstellen der fachlichen Anforderungen (Requirements Engineering)

- Ergebnis ist Lastenheft als Soll-Konzept mit funktionalen und ökonomischen Aspekten sowie Qualitätsaspekten
- Auftragnehmer erstellt daraus das Pflichtenheft

Entwurfsphase:

- fachlicher Entwurf beschreibt Objekte, Funktionen und Daten sowie deren Zusammenhänge
- technischer Entwurf berücksichtigt Hardware, Systemsoftware und Programmiersprache
- Ergebnisse sind logische und physische Datenstrukturen sowie Testfälle

Phasen (II)

21

Implementierung:

- Detaillierung des Systementwurfs
- Spezifikation von Klassen, Attributen und Methoden
- Umsetzung
- Test

Abnahme- und Einführungsphase:

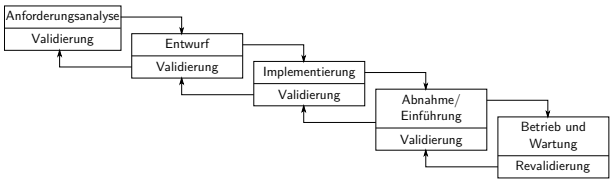
- vollständige oder modulweise Einführung der neuen Software?
- Prüfung, ob das Programm die Anforderungen des Pflichtenhefts erfüllt

Betrieb und Wartung:

- Anpassung des Programms
- Fehlerbeseitigung (Bug-Fixing)

Vor- und Nachteile des Wasserfallmodell

22



Vorteile:

- ⊕ dokumentengetrieben (Aktivitäten enden mit Dokument)
- ⊕ einfach, verständlich, wenig Aufwand
- ⊕ leichte Anpassbarkeit

Nachteile:

- ⊖ sequentielle Reihenfolge nicht immer sinnvoll
- ⊖ vollständige Durchführung nicht immer sinnvoll
- ⊖ Gefahr der Bürokratisierung
- ⊖ Risiken werden nicht berücksichtigt

4.4 Anforderungsmanagement

Warum ist Anforderungsmanagement wichtig?

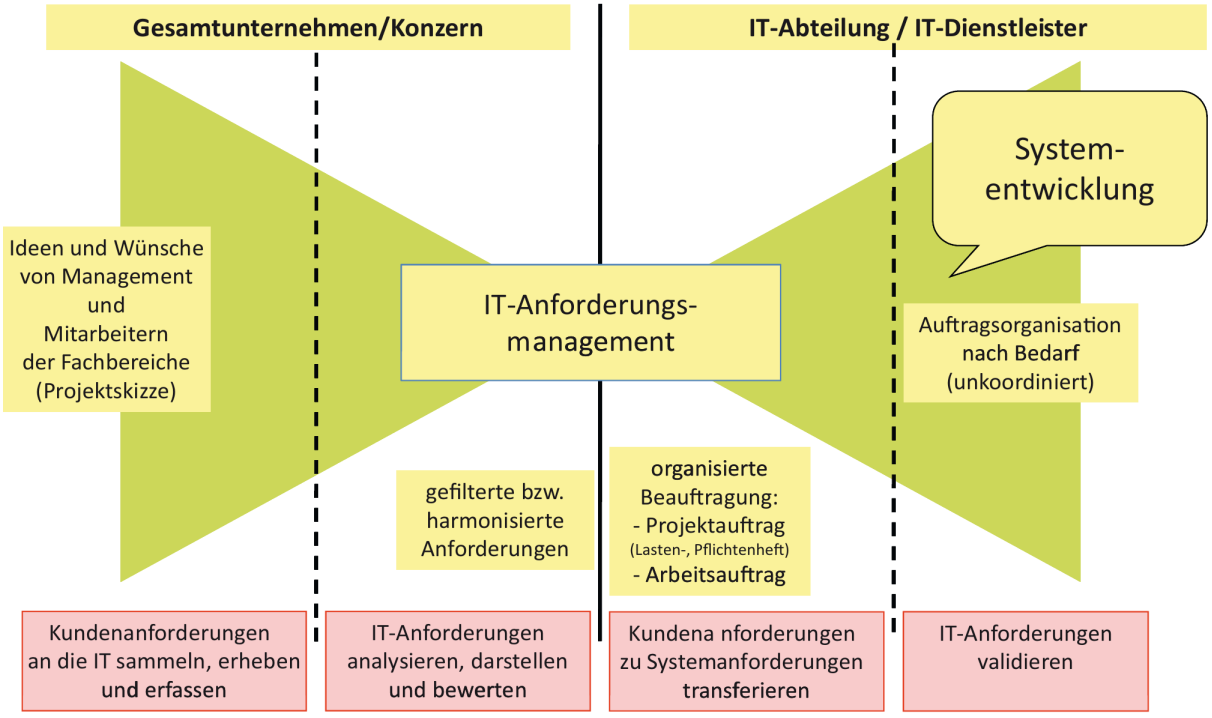
23

- IT-Systeme halten in immer mehr Bereiche Einzug und werden damit immer wichtiger für die Industrie
- Ansprüche an die Qualität von IT-Systemen steigen
- IT-Systeme werden immer komplexer
- IT-Systeme werden immer umfangreicher

Aufgabe des Anforderungsmanagements:

- bereits im Requirements Engineering potentielle Risiken identifizieren
- Fehler und Lücken in Anforderungsdokumenten erkennen, um spätere aufwändige Korrekturen zu vermeiden

Aufgaben/Prozesse



Quelle: Tiemeyer, 2017

Eine Anforderung ist ...

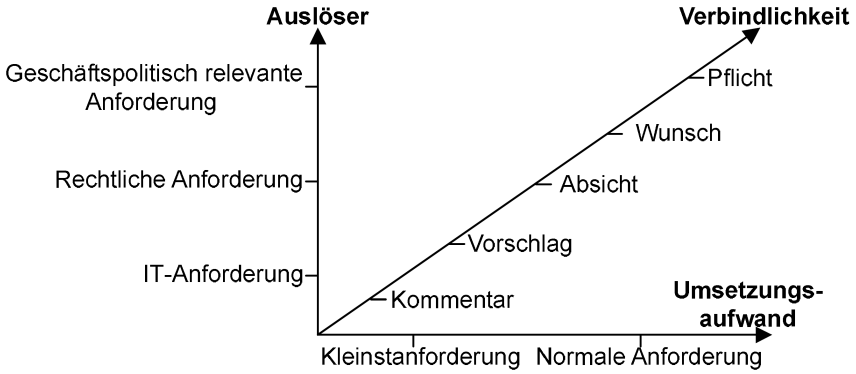
1. ... eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.

2. ... eine Bedingung oder Fähigkeit, die ein System oder Teilsystem 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 Eigenschaft gemäß 1) oder 2).

Klassifikation

26



Quelle: Schiermann, 2002

Arten von Anforderungen

27

Funktionale Anforderungen:

- legen die Funktionalität des geplanten Systems fest
- Anforderung bezüglich des Ergebnisses eines Verhaltens, das von einer Funktion des Systems bereitgestellt wird

Nichtfunktionalen Anforderungen:

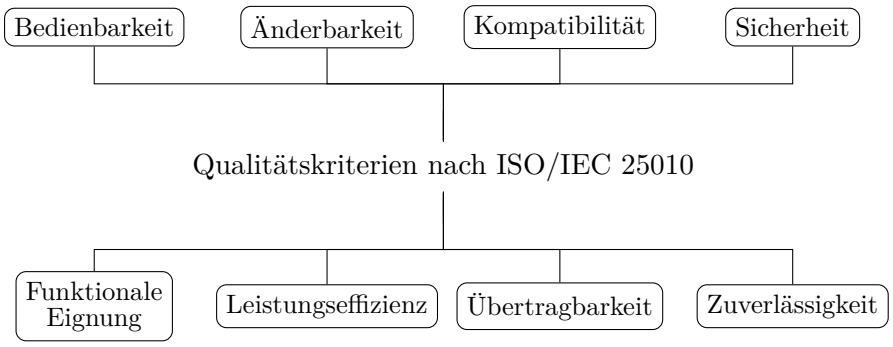
- beschreiben, wie gut das System die Leistung erbringen soll
- werden vielfach als Randbedingungen und Qualitätseigenschaften verstanden

$\Rightarrow$ Qualitätsanforderungen + Randbedingungen = *Nichtfunktionale Anforderungen*

Qualitätsanforderungen

28

- Anforderung, die sich auf ein Qualitätsmerkmal bezieht
- wird in der Regel nicht in einer Funktion umgesetzt
- Aussehen, Handhabung, Benutzbarkeit
- Leistung und Effizienz
- Betrieb und Umgebungsbedingungen
- Sicherheitsanforderungen



Randbedingungen 29

- nicht von den Projektbeteiligten beeinflussbar
- können sich beziehen auf
  - das System selbst
  - die Entwicklung des Systems
- werden nicht umgesetzt
- schränken den Lösungsraum weiter ein als was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen

Literatur

Fachbücher

Kuster, J., C. Bachmann, E. Huber, M. Hubmanna, R. Lippmann, E. Schneider, P. Schneider, U. Witschi und R. Wüst (2019). *Handbuch Projektmanagement Agil – Klassisch – Hybrid*. 4. Aufl. Berlin: Springer Gabler. ISBN: 978-3-662-57877-3.

Kuster, J., E. Huber, R. Lippmann, A. Schmid, E. Schneider, U. Witschi und R. Wüst (2008). *Handbuch Projektmangement*. Berlin: Springer. ISBN: 978-3-540-76431-1.

Rupp, C. und K. Pohl (2010). *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering – Foundation Level*. 2. Aufl. Heidelberg: dpunkt. 192 S. ISBN: 978-3-89864-708-3.

Schiermann, B. (2002). *Kontinuierliches Anforderungsmanagement. Prozesse – Techniken – Werkzeuge*. München: Addison-Wesley. ISBN: 3-8273-1787-8.

Tiemeyer, E. (2017). *Handbuch IT-Management - Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis*. 6. Aufl. München: Hanser. ISBN: 978-3-446-44347-1. DOI: 10.3139/9783446445376.