

Software Engineering - Motivation

Status quo:

- Software wirkt veränderlicher als physische Produkte, daraus resultiert häufig ein „quick&dirty“-Ansatz
- Modifikationen erfolgen häufig ad hoc
- da die Messlatte in der Historie tief lag, werden Bugs und schlechte Qualität häufig toleriert
- viele Amateure „trauten“ sich in die SW-Entwicklung, mit niedrigem Anspruch und geringer Berufsehre



Ziel:

- echte Ingenieurspraktiken bei der Softwarekonstruktion anwenden können und wollen



Software Engineering - Motivation

- sichere Software läuft 24/7 über viele Jahre
- Software kann viele Jahre lang bei hoher Qualität weiterentwickelt werden, zu bezahlbaren Kosten
- Kunden bleiben trotz anderer Produkte dem Anbieter treu
- Entwicklung großer Systeme ist langfristig nicht anders möglich
- Sichere Software basiert auf hoher Resilience: Widerstandsfähigkeit gegenüber Widrigkeiten

➤ Schrott ist keine Basis und keine Option!



Ablauf der Veranstaltung

➤ je 2 Schritte pro Themengebiet des Softwareengineering:

1. Dozent: Einführung ins Themengebiet, inklusive weiterführender Literatur
2. Studierende: Selbstständige Erarbeitung vorgegebener Unterthemen/ Begriffe (möglichst an einem Beispiel aus dem Semesterprojekt) und Präsentation in der nächsten Veranstaltung

- max. 5 Minuten zu einem Begriff des Themenbereichs
- Themenvergabe siehe jeweilige Themenliste des Themengebiets (s. Moodle)
- Quellenangaben nicht vergessen
- Präsentation oder Video sind zulässig

Inhalt:

- Begriff erläutern
- ein Beispiel für die Anwendbarkeit aus dem eigenen Semesterprojekt finden und erläutern, wenn möglich
- Einsatzgebiete erläutern

Kurzvorträge: Themengebiet RE

16. 1. 25, 13:00 Uhr s Moodle

- Englische Standardbegriffe werden englisch verwendet – bitte auch in Kurzvorträgen!
- Prüfung: als Programmentwurf
- Prüfungsschwerpunkte:
 - Requirementsdefinition (User stories) mit Akzeptanzkriterien, funktional (5) und nichtfunktional (3)
 - Testfälle (3 pro Requ.)
 - Vorkehrungen für Sicherung der Zuverlässigkeit/ Resilience (mind. 3)
 - Design (Einsatz von 2 Pattern)
 - Architektur (Abbildung eines Use cases mit UML)
- Vorgabe: > 50 Softwarebasierte Produkte

Prüfung: PE (benotet)

- 50 Produktvorschläge, 1 auswählen
- dafür nach vorgegebenem Bewertungsschema SW-Engineering nachweisen:

Bewertungskriterien		Punkte	Summe
RE	Beschreibung des zugrundeliegenden Problems und von 3 (UND-verknüpften) Akzeptanzkriterien im Problemraum, mit denen nach der Bearbeitung festgestellt werden kann, ob das Problem gelöst werden konnte.		0 (4)
	Darstellung von 3 Lösungsoptionen, Benennen und Begründen der bevorzugten Lösungsoption		0 (4)
	Aufstellung von 6 Stakeholdern der Lösungsoption		0 (3)
	Beschreibung von 5 funktionalen Requirements der ausgewählten Lösungsoption		0 (5)
Zuverlässigkt.	5 testbare, nichtfunktionale Requirements		0 (5)
Test	je 1 Testfall pro Requirement (funktionale und nichtfunktionale), automatisiert ausführbar (Beschreibung der Schritte)		0 (10)
Architektur	Begründung der Architekturauswahl anhand von 5 Kriterien, Dokumentation der Architektur mit mindestens 3 UML-Sichten		0 (8)
Design	Möglicher Einsatz von 3 Designpattern (nicht unbedingt aus der Vorlesung) inkl. Beschreibung des Verwendungsortes		0 (6)
			0 (45)

- Ausgabe der Themen (50 Produkte): 1. 3. 2025
- spätestes Abgabedatum: 25. 4. 2025



- vom Problem

... zur Festlegung der Lösung aus mehreren Optionen

... über deren Realisierung

... bis zum Liefer- und dann Nutzungsstop...

Themengebiete

1. Requirements Engineering - Anforderungserhebung
Requirements Management – Anforderungsverwaltung
2. Dependability: Verlässlichkeit und Resilience
3. Testen
4. System-/ Software-Architekturfindung
5. Softwaredesign

Systems engineering (SE) ist ein interdisziplinärer Ansatz, um die Entstehung erfolgreicher (technischer) Systeme zu ermöglichen (Produktentstehung)" (INCOSE (International Council on Systems Engineering) 2012).

Systems engineering fokussiert sich auf

- das ganzheitliche und umfassende Verständnis der Bedürfnisse der Stakeholder;
- das Untersuchen von Chancen;
- das Dokumentieren von Requirements,
- und das Erzeugen, Verifizieren, Validieren und Weiterentwickeln von Lösungen in Betrachtung des Gesamtproblems;
- von der Erarbeitung des Systemkonzepts bis zur Systemübergabe

Im Zentrum des Systems engineering steht der Problemlösungsprozess:
die konstruktive Arbeit an der Lösung selbst wie auch die Organisation und Koordination des Lösungsprozesses.

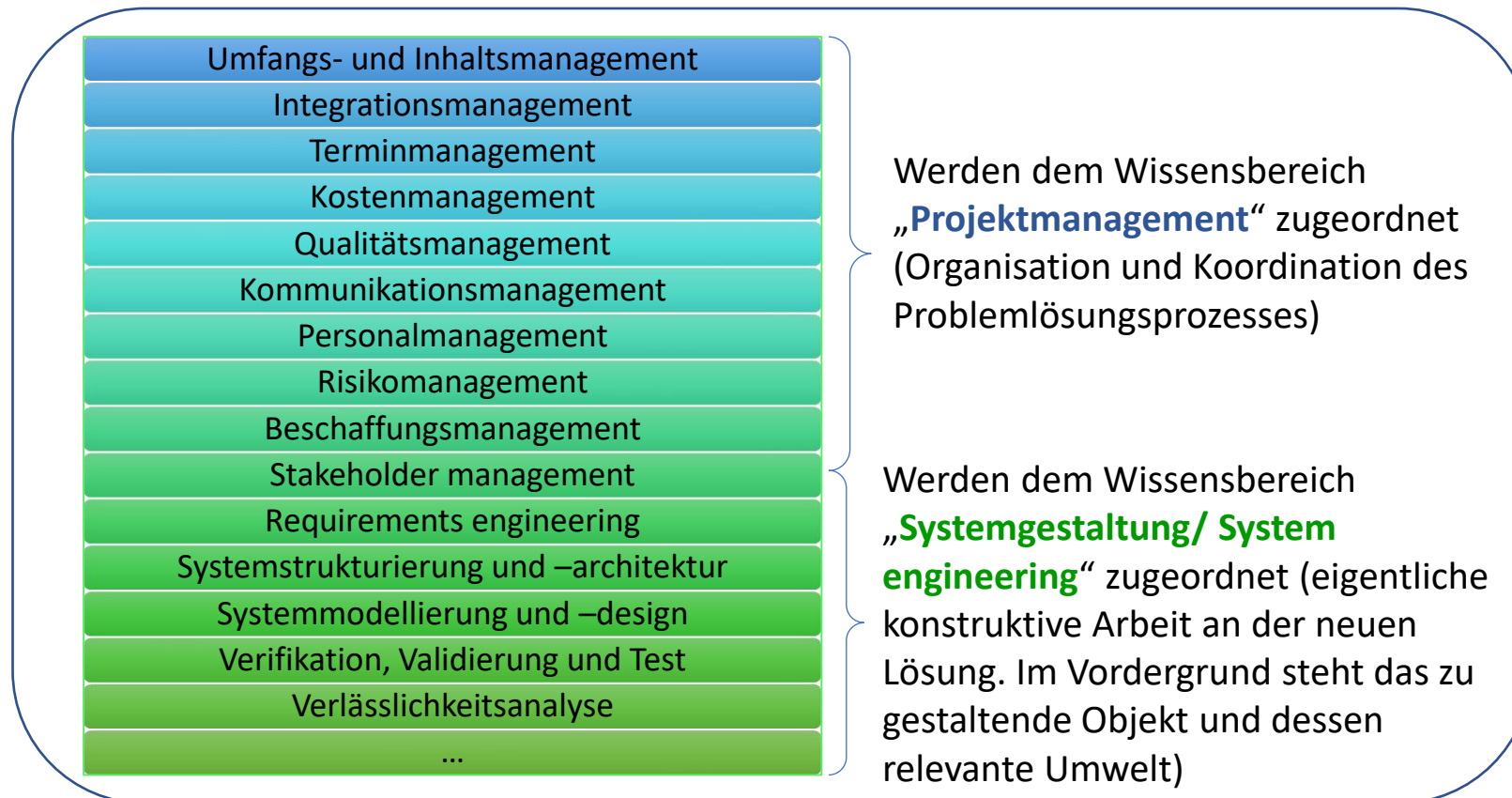
Der CEO der Firma, für die Sie als Entwicklungsleiter arbeiten, fragt Sie, warum die neue Enterprise-Software, für die er Millionen gezahlt hat, immer noch Fehler hat.

a) Was antworten Sie ihm, wenn die Qualität eigentlich sehr hoch ist?

Der CEO der Firma, für die Sie als Entwicklungsleiter arbeiten, fragt Sie, warum die neue Enterprise-Software, für die er Millionen gezahlt hat, immer noch Fehler hat.

b) Und was, wenn die Qualität wirklich erbarmungswürdig ist?

Wissensgebiete der Produktentstehung



Wissensgebiete der Produktentstehung

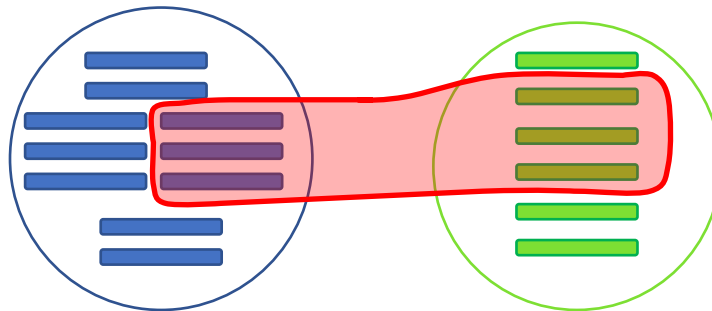
Projektmanagement



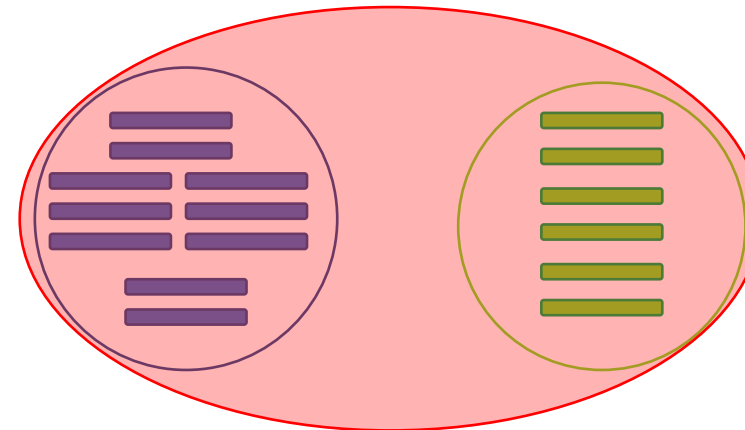
Systemgestaltung



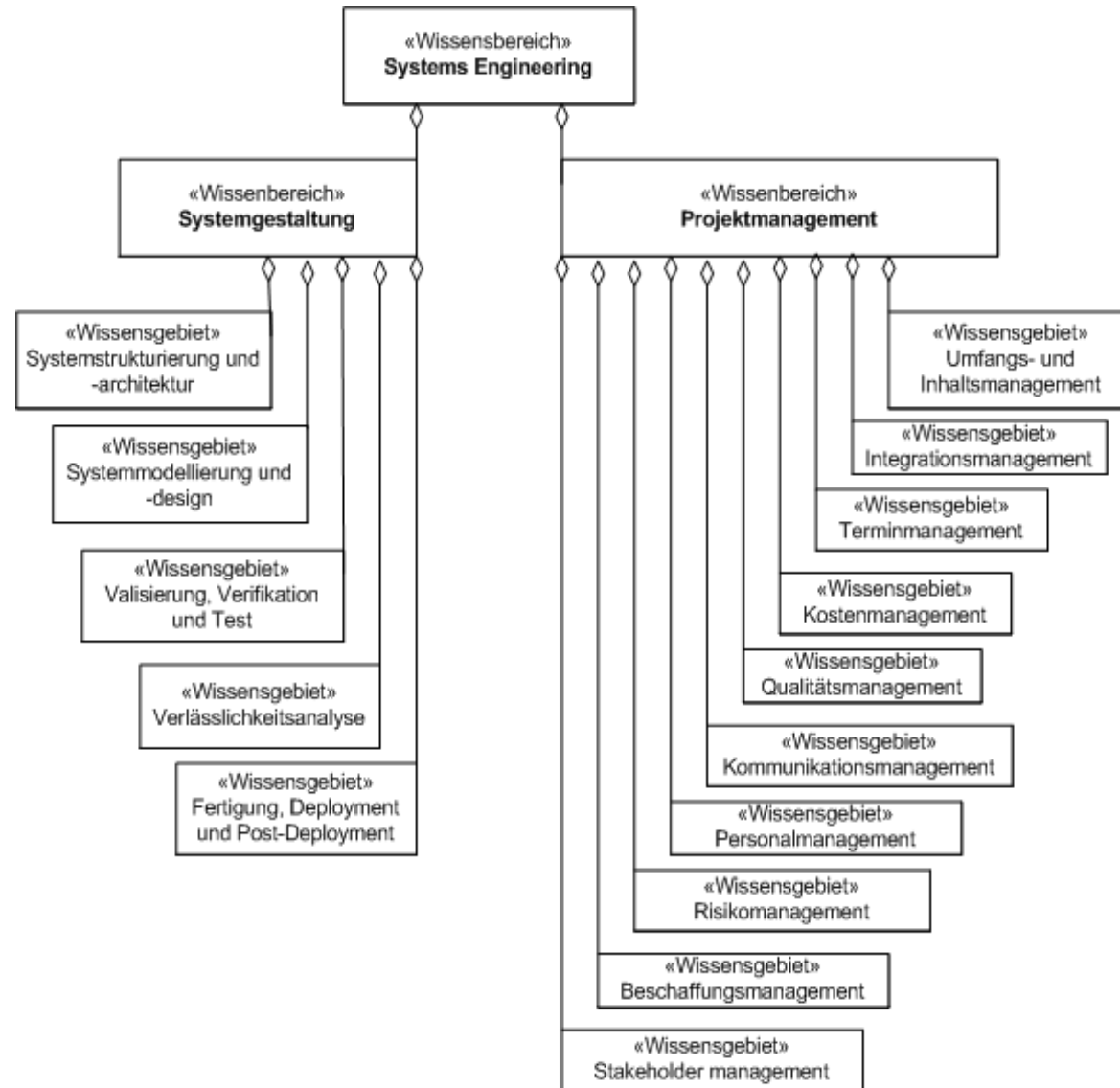
Systems Engineering für Projekt A



Systems Engineering für Projekt B



Wissensgebiete der Produktentstehung



Systems engineering roles

Requ. engineer

Systemarchitekt

Entwicklungsteam

Testmanager

Projektmanager

Anforderungsanalyse

Anforderungsverwaltung

Systemstrukturierung und –architektur

Systemmodellierung und –design

Verifikation, Validierung und Test

Verlässlichkeitsanalyse

Fertigung, Deployment und Postdeployment

Stakeholder management

Qualitätsmanagement

Terminmanagement

Kommunikationsmanagement

Personalmanagement

Kostenmanagement

Risikomanagement

Integrationsmanagement

Umfangs- und Inhaltsmanagement

Beschaffungsmanagement

