

SOFTWARE ENGINEERING 2

00 - Motivation

Zur Person

- Stefan Betermieux
- Lehrgebiete → Software-Engineering, Software-Ergonomie
Seit 01. März 2014 an der HS Furtwangen
- Vorher:
 - ▶ Senior Consultant, IKS Hagen
 - ▶ Professor FH Dortmund
- Kontakt: best@hs-furtwangen.de
Raum: C 2.07
Sprechstunde: Dienstag, 13:00 - 14:00 Uhr



Organisatorisches

- Doppelstunde Vorlesung am Montag
 - ▶ Vorstellung der neuen Inhalte
 - ▶ Zeit für Fragen, Diskussionen
- Praktikum am Montag
 - ▶ praktische Aufgaben am Rechner
 - ▶ allgemein gehaltene Aufgaben → Ergebnisvielfalt
- auch lösbar mit eigenem Rechner, zentrale Infrastruktur wird bereitgestellt
- ab der zweiten Woche

Felix

- In Felix existiert ein Bereich für diese Veranstaltung:
 - ▶ Katalog → Fakultät IN → Software Engineering 2
- Vorlesungsfolien werden zeitnah veröffentlicht
- Diskussionsforum für ...
 - ▶ allgemeine Fragen
 - ▶ Ankündigungen
- Links zur bereitgestellten Infrastruktur
- Weitere Bereiche nach Bedarf
- Bitte heute als Student/in registrieren (und evtl. für eine Praktikumsgruppe eintragen)!

Klausur

- Klausurdauer: 90 Minuten
- Maximal 100 Punkte erreichbar
- Bestanden mit 50 Punkten
- Aufgaben vergleichbar mit Übungsaufgaben
 - ▶ lösbar mit Papier und Stift
 - ▶ Abfrage des Kenntnisstands, nicht jedes Komma wird bewertet
 - ▶ Transferaufgaben → Übertragung von gelerntem Wissen auf neue Problemstellungen

Notenskala

	sehr gut	1,0	95
	gut	1,3	90
		1,7	85
		2,0	80
		2,3	75
		2,7	70
	befriedigend	3,0	65
		3,3	60
		3,7	55
	ausreichend	4,0	50
	mangelhaft	5,0	0

Was können Sie erwarten?

- Verständliche Stoffvermittlung
 - ▶ Beispiele, Übungen, Kurzumfragen
 - ▶ Erfragen von Feedback, ob Inhalte verstanden worden sind
- Großes Interesse am Lernerfolg der Studierenden
- Intensive Betreuung im Felix
- Interessantes Projekt im Praktikum
- Erreichbarkeit meinerseits

Was von Ihnen erwartet wird?

- Bei der Vorlesung
 - ▶ Mobiltelefone leise
 - ▶ kein Tastaturgeklapper
 - ▶ keine Verspätungen, keine Gespräche, kein Frühstück!
 - ▶ das stört vor allem Ihre Kommiliton/innen
- Fragen, Anregungen und Feedback (z.B. im Felix)
- Hohe Eigenmotivation zur Vor- und Nachbereitung der LV

Voraussetzungen

- Programmierkenntnisse Java
- Modellierung mit UML
- Datenbanken und Informationssysteme
- Netzwerk- und Internettechnologie
- Grundkenntnisse in HTML, CSS und JavaScript
- Keine Angst: Alles wird kurz angerissen und wiederholt!

Akronymitis

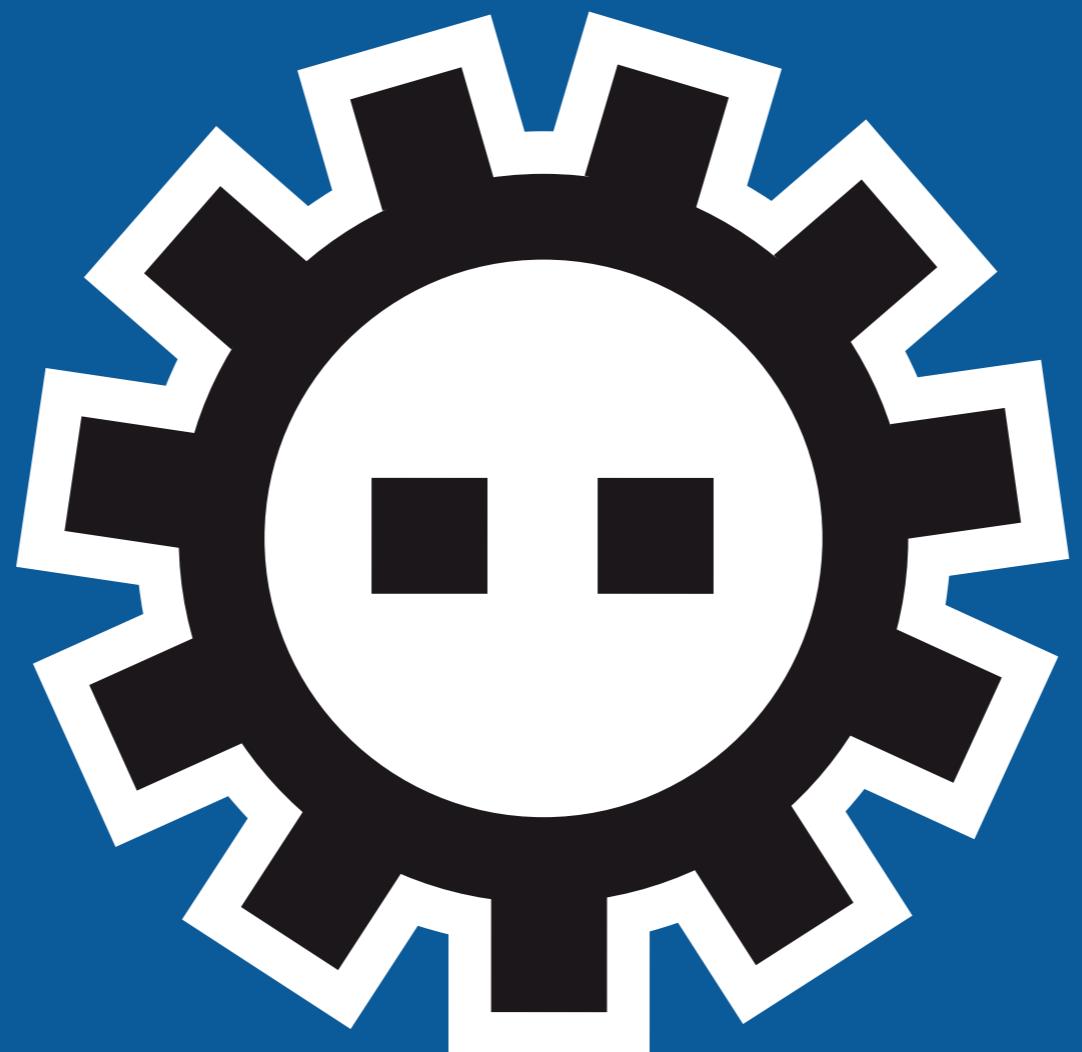
A word cloud visualization where the size and color of each word represent its frequency or importance in the context of web development. The words are arranged in two main vertical columns. The left column includes: IO, CSS, SSL, FTP, JPA, CSS3, O, URL, HTML, TCP/IP, HTML5, MVC, WSDL, WHATWG, DI, WOFF, P, SGML, WebMVC, CRUD, JMS, G, JSTL, WebGL, Spring, UDDI, PHP, JEE, S, JEE, URI, DTD, SAX, W3C, XML, RMI, CGI, JSF, and JSF. The right column includes: SOAP, XHTM, JSP, SMTP, XPATH, JSON, PNG, AJAX, DRY, XML, RMI, CGI, and JSF.

Fragen

Fragen zur Organisation der Lehrveranstaltung?

Erwartungen

- Viele Inhalte der Veranstaltung sind durch das Modulhandbuch und den Dozenten bereits vorgeben.
- Aber: **Was sind Ihre Vorstellungen und Interessen?**
- **Welche Themen im Bereich der Softwareentwicklung halten Sie für wichtig, interessant, betrachtenswert?**
- Anhaltspunkte: Persönliches Interesse, Wichtig im Berufsleben, Wichtig für allgemeines Informatik-Verständnis, ...
- Gruppenarbeit:
 - ▶ 3er Gruppen (jeweils 3 Personen die nebeneinander sitzen)
 - ▶ Bearbeitungszeit: 5 Minuten



MOTIVATION

Begriffe und Definitionen

- **1965:** Der Begriff der **Softwarekrise** etabliert sich in Industrie und Wissenschaft.
 - ▶ Fehler in Computersystemen sind fast immer auf Softwarefehler zurückzuführen.
 - ▶ Software wird oftmals nicht termingerecht bzw. zu deutlich höheren Kosten, als ursprünglich geschätzt, fertiggestellt.
 - ▶ Software entspricht oft nicht den konkreten Anforderungen der Benutzer
- **1968:** Der Begriff der **Software-Engineering** wird geprägt.
 - ▶ Er wird von F.L. Bauer im Rahmen einer Studiengruppe der NATO zum Thema Informatik geprägt.
 - ▶ Auslöser des Begriffs **Software-Engineering** war die **Softwarekrise**.

Definition (1)

- **Software Engineering**

The systematic approach to the development, operation, maintenance, and retirement of software.

Ansi 83: IEEE Standard Glossary of Software Engineering Terminology

- **Softwaretechnik (syn.: Software Engineering)**

Fachgebiet der Informatik, das sich mit der Bereitstellung und systematischen Verwendung von Methoden und Werkzeugen für die Herstellung und Anwendung von Software beschäftigt.

Hesse et al. 84: Ein Begriffssystem für die Softwaretechnik

Definition (2)

- Zielorientierte Bereitstellung und systematische Verwendung von
 - ▶ Prinzipien,
 - ▶ Methoden,
 - » Konzepte,
 - » Notationen und
 - ▶ Werkzeugen
- für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Software-Systemen.
- Zielorientiert bedeutet hierbei die Berücksichtigung z.B. von Kosten, Zeit, Qualität.

Software-Engineering vs. Programmierung

- In den ersten Semestern des Studiums haben Sie programmierintensive Veranstaltungen besucht
- Sie können Programmieren, aber können Sie auch gute Software entwickeln?
- Welches Wissen, welche Kenntnisse, welche Fähig- und Fertigkeiten benötigt man aus Ihrer Sicht, um gute Software zu entwickeln?

Vergleich mit der Architektur



Hundehütte

- kann von einer Person gebaut werden
- erfordert
 - ▶ minimalen Entwurf
 - ▶ einfachen Prozess
 - ▶ einfache Werkzeuge

Wolkenkratzer

- wird von einem Team erbaut
- erfordert
 - ▶ Entwurf (durch Architekt)
 - ▶ Analyse (durch Statiker)
 - ▶ Genehmigung (durch Bauaufsicht)
 - ▶ Realisierungsprozess
 - ▶ aufwendige Werkzeuge



Paradoxon

- **Hundehütte → Wolkenkratzer**

Wenn jemand mal eine Hundehütte gebaut hat, würden wir ihn vermutlich noch lange keinen Wolkenkratzer bauen lassen.

- **Programm → Softwaresystem**

Wenn jemand dagegen programmieren kann, lässt man ihn heutzutage aber oft bedenkenlos Software entwickeln ...

Ingenieurmäßiges Vorgehen

- Ingenieurwesen ist auf erfahrungsgeleitetes Vorgehen angewiesen
 - ▶ »Normales Vorgehen«
- Das Vorgehen wird immer irgendwo von vorherigen Erfahrungen abweichen
 - ▶ »Radikales Vorgehen«
- Ingenieurmäßiges Vorgehen besteht darin, solche Abweichungen zu vermeiden/gering zu halten
 - ▶ Radikales Vorgehen nur, wo unbedingt nötig!
- Ansätze für normales Vorgehen:
 - ▶ Was sind die typischen Probleme in der SW-Entwicklung?
 - ▶ Was sind die dazu passenden Lösungsansätze des SE?

Fallbeispiel: Die Tacoma Narrows Bridge (TNB)

- Am 1. Juli 1940 wurde bei Tacoma im US-Bundesstaat Washington eine neue Hängebrücke über dem Puget Sound eröffnet.
- Baubeginn: 1938
Chefberater: Leon Moisseiff (Experte für die Konstruktion von Hängebrücken)
Erfahrung: Manhattan Bridge, Bay Bridge, George Washington Bridge und die Golden Gate Bridge



Leon J. Moisseiff
(1872 - 1943)



Golden Gate Bridge
1970m Länge
1280m Spannweite



Bay Bridge (West)
2820m Länge
700m Spannweite



Tacoma Narrows Br.
1520m Länge
850m Spannweite

Quelle: Wikipedia

Technische Daten

- Die Tacoma Narrows Bridge hatte überwiegend ganz übliche Parameter für die Einzelteile:
 - ▶ übliche Pylonen
 - ▶ übliche Seile
 - ▶ eine übliche Verankerung
 - ▶ eine übliche Fahrbahn (2 Fahrspuren, 2 Gehwege)
 - ▶ einen weitgehend üblichen Unterbau
- unüblich war lediglich die Schlankheit:
 - ▶ Tragwerk nur 12m breit und 2,40m hoch
 - ▶ Das Verhältnis Tragwerkhöhe:Spannweite war 1:350 (üblich war eher bis zu 1:90)
 - ▶ Das Verhältnis Tragwerkbreite:Spannweite war 1:72 (Golden Gate Bridge: 1:47)
 - ▶ niedrige Steifigkeit und niedriges Gewicht

Spitzname »Gallopingle Gertie«



<http://www.youtube.com/watch?v=j-zczJXSxnw>

Ursachen

- Keine Ursache: Die Windstärke
Sie war mit ca. 75 km/h noch deutlich im zulässigen Bereich
- Unfallursache (Bedingungen):
Die Längsschwingungen, für die die Brücke nicht konstruiert und vorgesehen war
- Unfallherkunft:
 - ▶ das Rechenmodell von Moisseiff berücksichtigte nur Querschwingungen
 - ▶ Urgrund: Längsschwingungen waren vor TNB nie relevant gewesen, weil vergleichbar lange Brücken stets viel schwerer und steifer gebaut worden waren
- **Moisseiffs Konstruktion lag außerhalb des Bereichs, über den es gesicherte Erfahrungen gab!**

Kognitive Kompetenzen

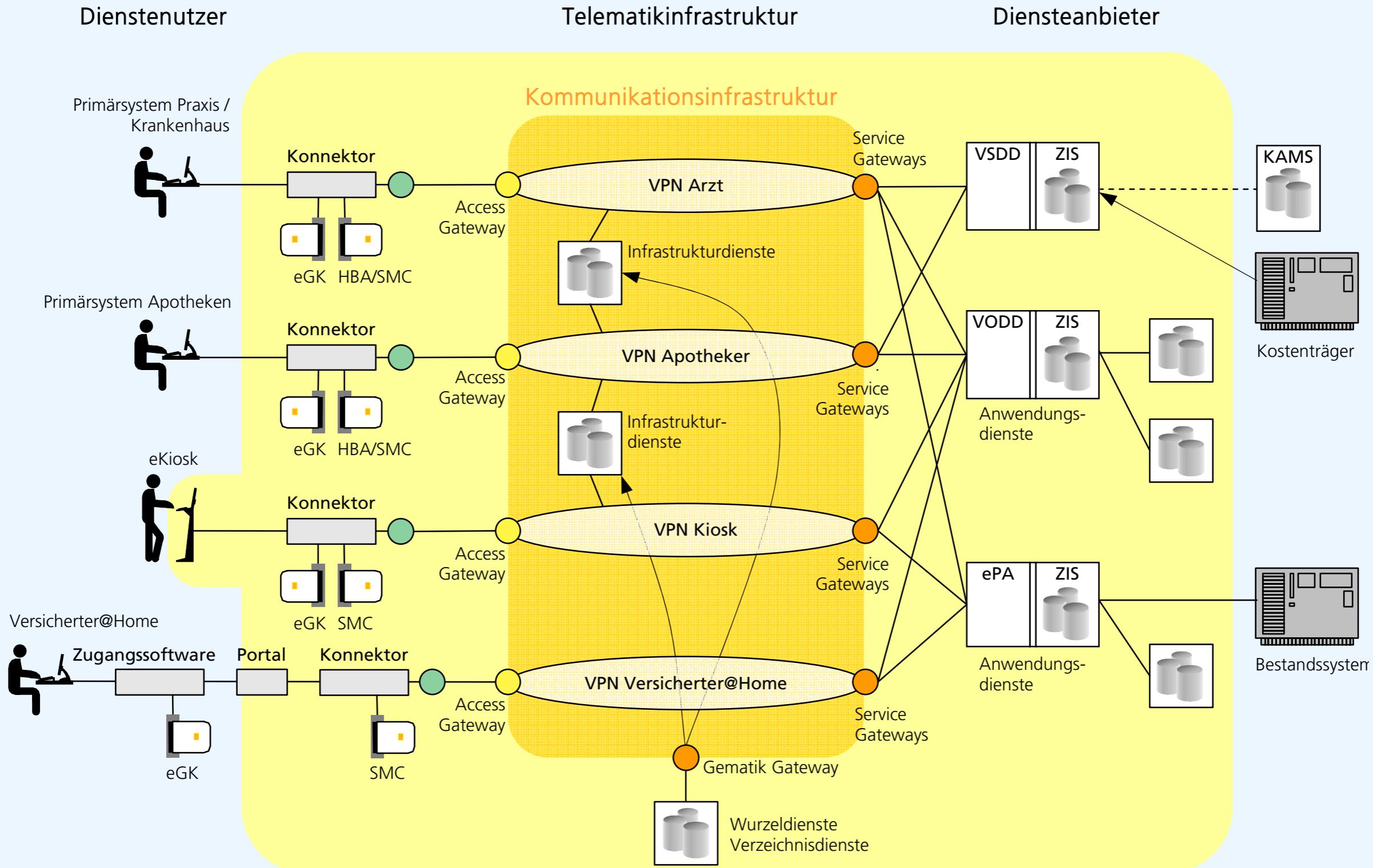
Erforderliche kognitive Kompetenzen für das Software-Engineering:

1. Kennen	Gleich wiedergeben	
2. Verstehen	Anders wiedergeben	nicht relevant
3. Anwenden	Folgern in konkreten Fällen	
4. Analysieren	Zerlegen; Verstehen der Struktur	
5. Synthetisieren	Neue, ähnliche Strukturen bilden	mindestens erforderlich
6. Beurteilen	Brauchbarkeit bewerten	

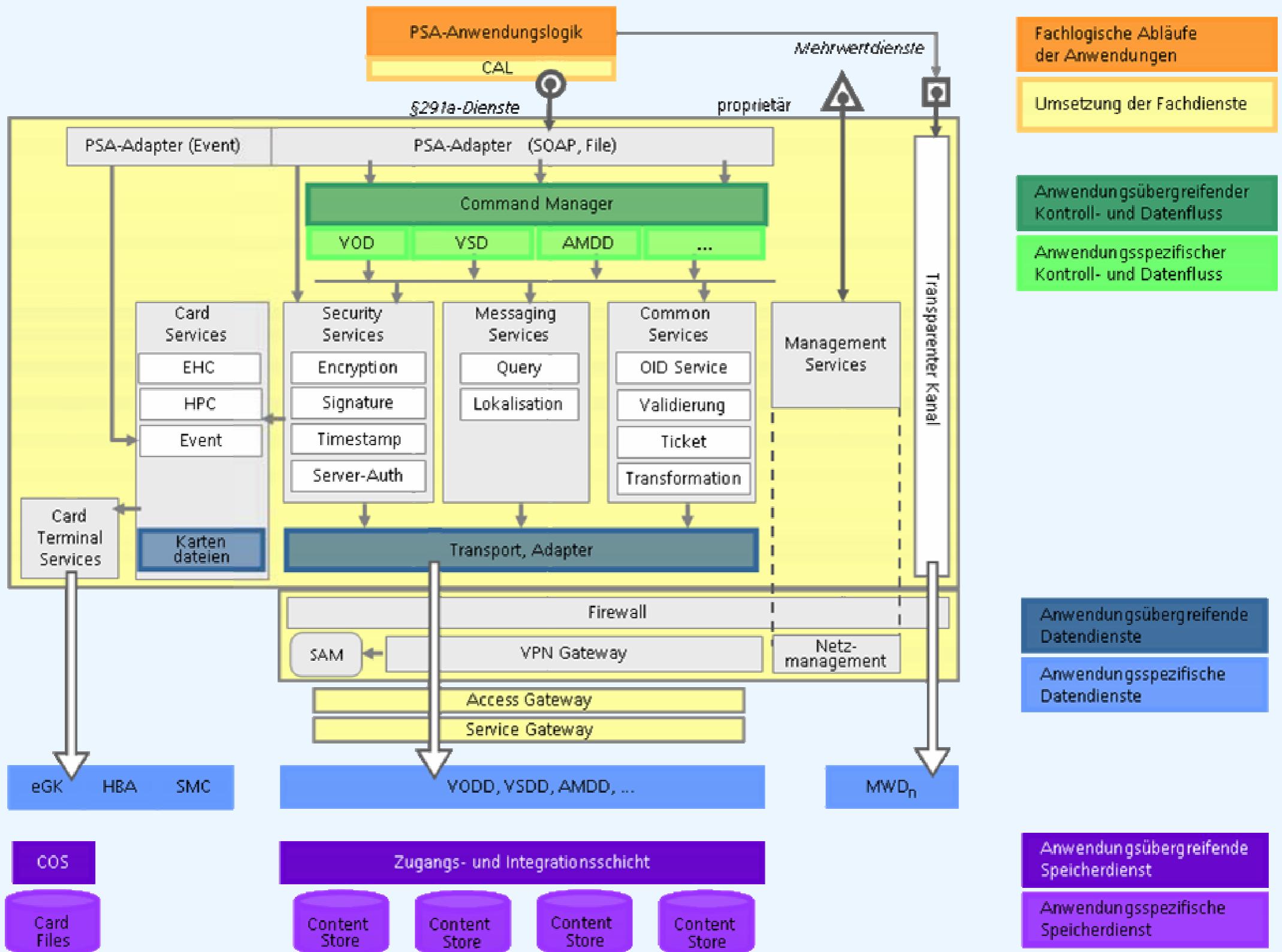
Ein Beispiel aus der Praxis: Gesundheitskarte



Architektur - organisatorisch

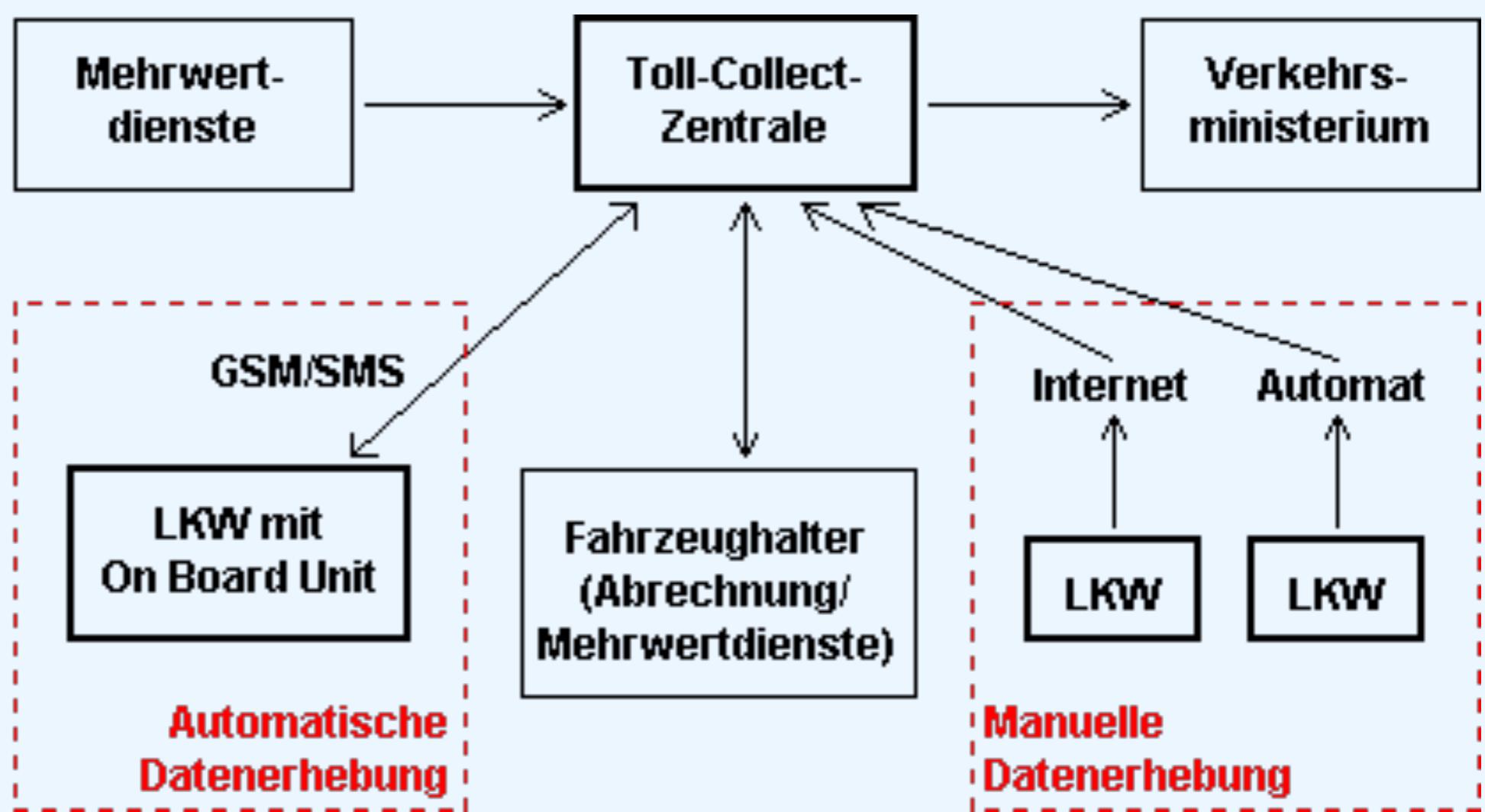


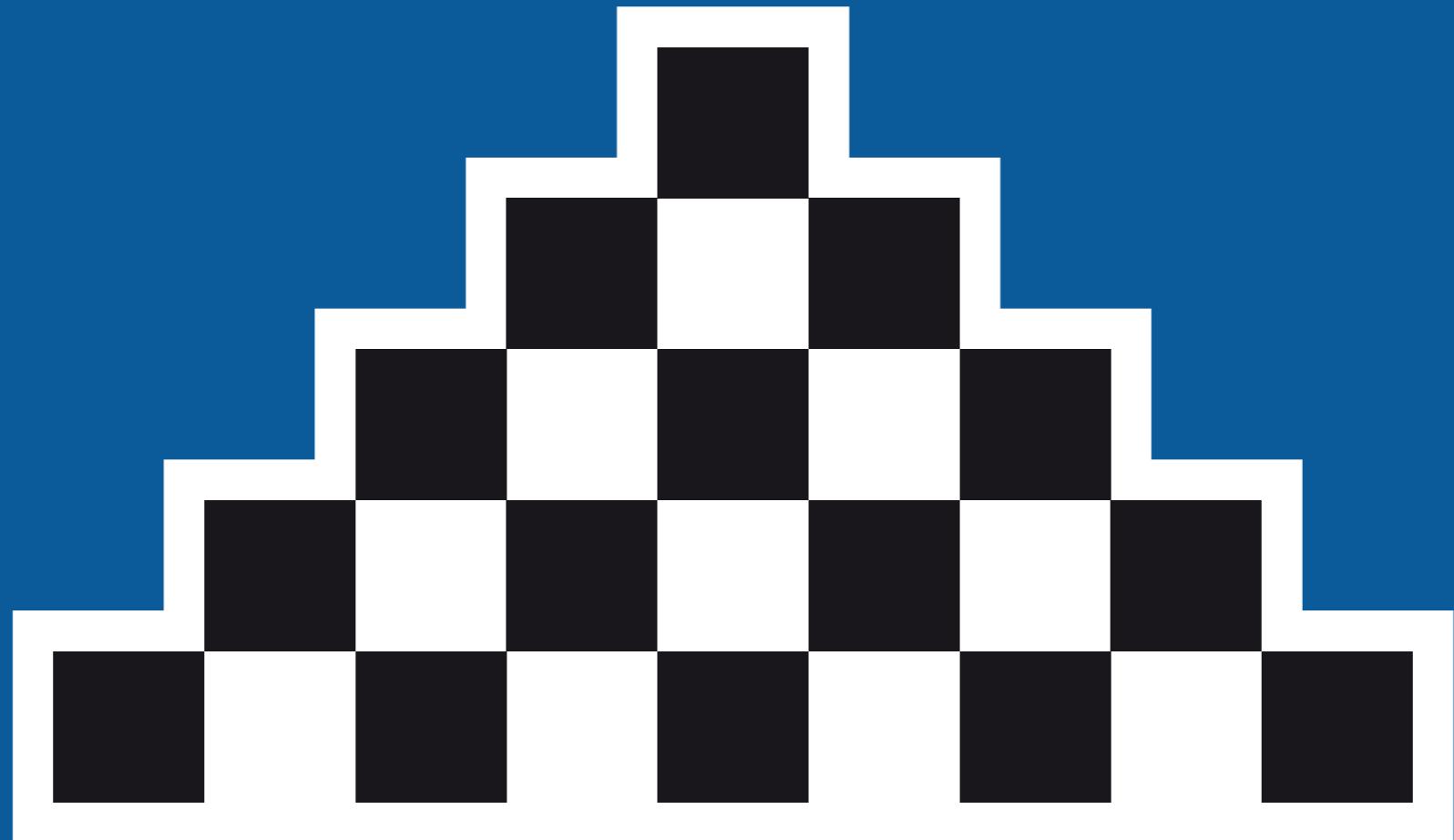
Architektur - technisch



Aus: Spezifikation der Lösungsarchitektur zur Umsetzung der Anwendungen der elektronischen Gesundheitskarte

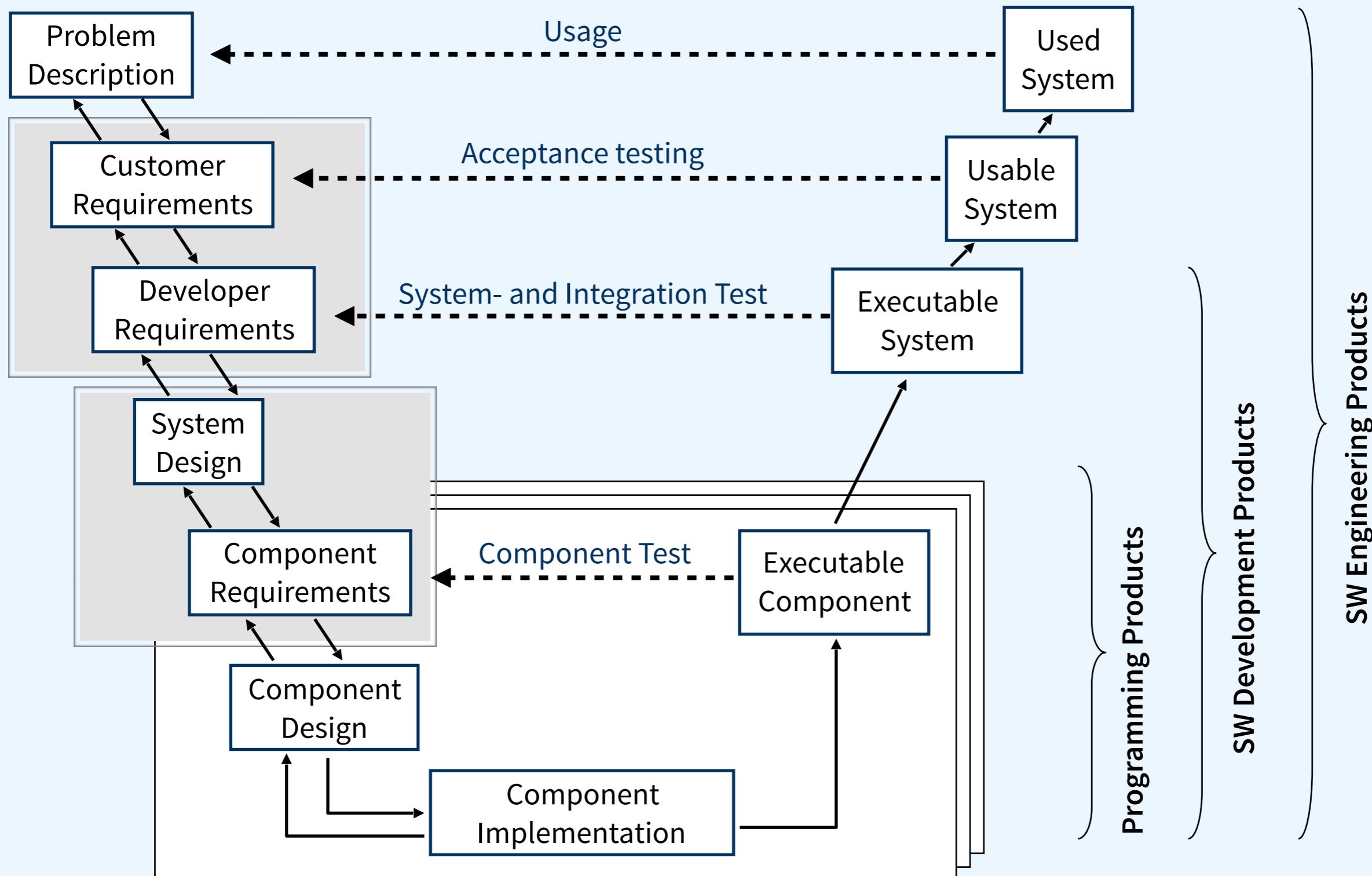
Toll Collect





GRUNDLAGEN

Software-Produkte: Überblick



Terminologie

- Software-Engineering
 - ▶ ist die ingenieurmäßige Erstellung eines Softwaresystems zur Lösung eines Anwenderproblems mittels auszuwählender T/M/W.
- Software-Entwicklung
 - ▶ ist die ingenieurmäßige Erstellung eines Softwaresystems nach vorgegebenen Anforderungen mittels vorgegebener T/M/W.
- Programmieren
 - ▶ ist die ingenieurmäßige Erstellung einer Software-Komponente nach vorgegebenen Anforderungen mittels vorgegebener T/M/W.

T/M/W = Techniken / Methoden / Werkzeuge

Definition von Software

- Software ist eine Sammlung von Computerprogrammen, Prozeduren, Regeln, zugehöriger Dokumentation und Daten (vgl. IEEE).
 - ▶ Programme als Teilmenge von Software.
 - ▶ Software beinhaltet Dokumente zur Beschreibung verschiedener Abstraktionsebenen für verschiedene Personengruppen (Benutzer, Entwickler, Management, ...).
 - ▶ Software kann für den freien Markt oder für einen bestimmten Kunden entwickelt werden.
- Software-Entwicklung ist mehr als nur die Entwicklung von Programmen.

Probleme vs. Lösungen (1)

Welche Probleme versucht die Softwaretechnik u.a. zu lösen?

- »Am Schluss hat sich rausgestellt, dass wir zur Hälfte Sachen gebaut hatten, die der Kunde nicht brauchte, aber dafür mit dem Rest nie genau das richtige.«
→ Anforderungsbestimmung
- »Die Software ist am Markt nicht sehr erfolgreich. Die Kunden behaupten meistens, sie können ihre Probleme damit nicht lösen.«
→ Anforderungsbestimmung, Software-Ergonomie

Probleme vs. Lösungen (2)

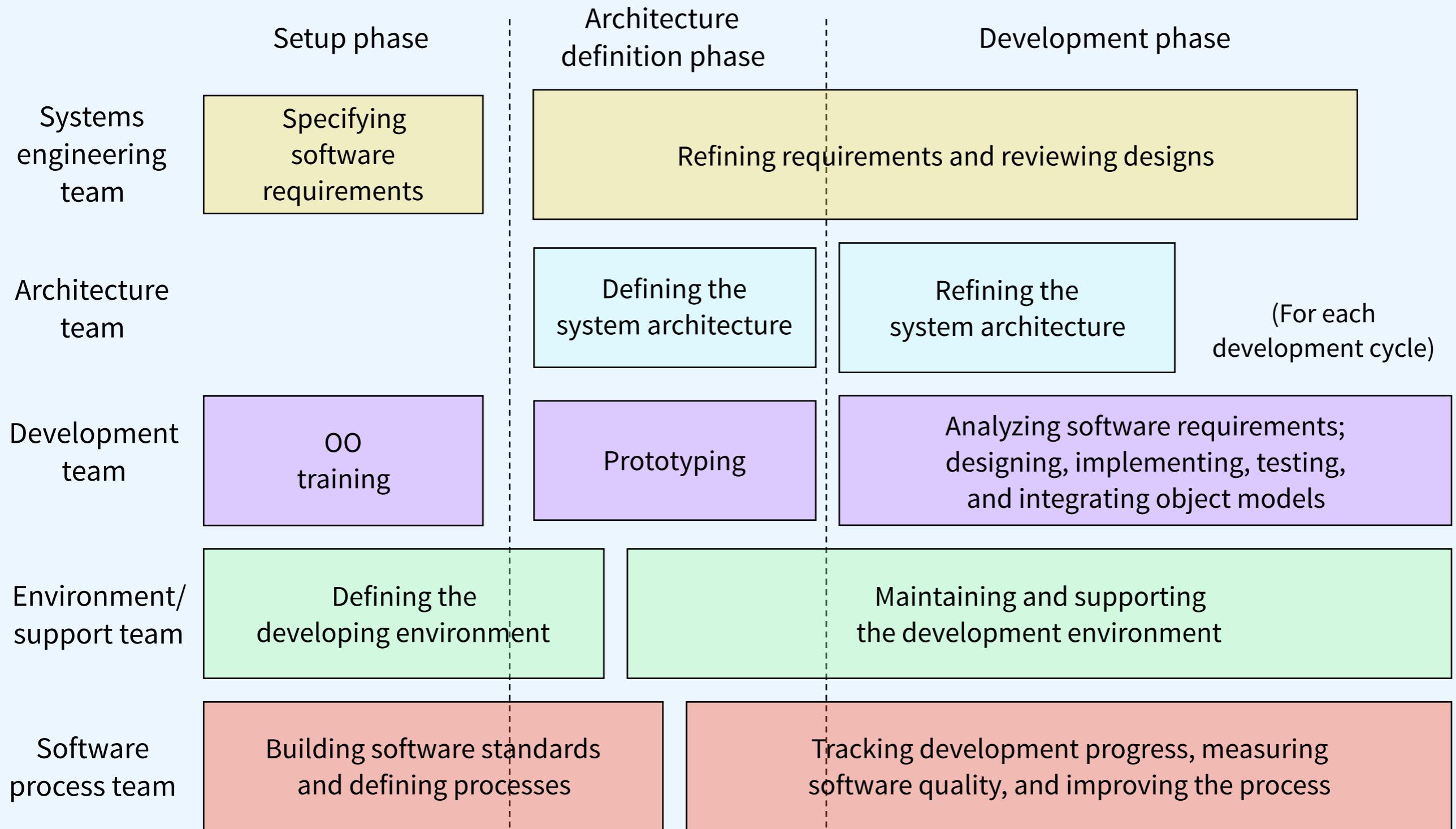
Welche Probleme versucht die Softwaretechnik u.a. zu lösen?

- »Die Fehler, die uns am meisten Zeit und Aufwand gekostet haben, haben wir alle schon ziemlich früh gemacht.«
→ Anforderungsbestimmung, Software-Entwurf
- »Die Hälfte der Zeit beim Implementieren geht dafür drauf, dass die Leute irgendwas untereinander abstimmen.«
→ Software-Entwurf

Softwaretechnische Kompetenz

- Sie werden nach dem Studium nicht dafür bezahlt, dass ...
 - ▶ Sie einen Hochschulabschluss haben
 - ▶ Sie etwas wissen
 - ▶ Sie etwas tun
 - ▶ Sie irgendwelche Probleme lösen
- sondern dafür
 - ▶ dass Sie die Probleme lösen, bei denen der Nutzen der Lösung höher ist als die Kosten der Lösung
- Dazu müssen Sie lernen:
 - ▶ Probleme und ihre Wichtigkeit zu verstehen
 - ▶ Probleme zu lösen
 - ▶ Kosten und Nutzen von Technologien und Methoden abzuschätzen

Strategic Development Program





BEISPIELE

übergreifendes Beispiel für Vorlesung und Praktikum

Veranstaltungsprojekt

Motivation

- Einzelne Vorlesungen sind nicht unabhängige »Informationshappen«
- Roter Faden durch das gesamte Semester:
 - ▶ Wie entwickle ich eine komplexe Web-Anwendung?
- Kommentar aus einem Grundlagenkurs:
 - ▶ "Warum lernen wir nicht, Anwendungen wie Facebook zu erstellen?"
- Facebook ist nicht besonders geeignet, aber...

Veranstaltungsprojekt

- Erstellung einer Web-Anwendung zur Veröffentlichung von Kurznachrichten
- Einfache Anforderungen:
 - ▶ Nutzer veröffentlichen Kurznachrichten
 - ▶ Kurznachrichten sind öffentlich
 - ▶ Nutzer können anderen Nutzern »folgen«

nicht angemeldet

NACHRICHTEN

REGISTRIERUNG		
NACHRICHTEN		
NEUE NACHRICHT		
ANMELDEN		
	Heike Schmidt @heike	vor 4 Minuten
	Zuhause ist da, wo man sein WLAN-Passwort kennt	
	Karl Müller @karl	vor 8 Minuten
	Toleranz ist gut. Aber nicht gegenüber Intoleranten. (Wilhelm Busch)	
	Stefan Schneider @stefan	vor 1 Stunde
	Die Lage ist hoffnungslos, aber nicht ernst. (Paul Watzlawick)	
	Karl Müller @karl	vor 1 Tag
	Hab mich im Lidl auf die Obstwage gelegt und den 82kg/714€ Etikett auf eine Traube geklebt. Gesicht der Kassiererin ist jeden Cent Wert!	
	Heike Schmidt @heike	16 September 2014
	Knetradierer nicht mehr neben offene Haribo-Tüte legen!	
	Karl Müller @karl	27 August 2014
	Das Homeoffice ist der Tod der Körperpflege.	
	Stefan Schneider @stefan	28 Juli 2014
	Hier fuhr gerade das Auto von Ilona und Hans vorbei. Ohne den entsprechenden Aufkleber hätten wir das nie erfahren.	
	Stefan Schneider @stefan	18 Juni 2014
	Wenn man jeden Tag eine Stunde joggt, lebt man 2 Jahre länger. Dumm nur, dass man dann 4 Jahre seines Lebens gejoggt ist.	

DANKE