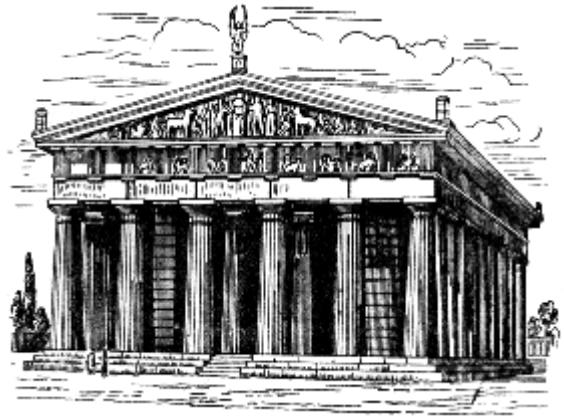
Softwarearchitektur

2. Ziele und Ergebnisse des Architekturentwurfs



Vorlesung Wintersemester 2005 / 06

Technische Universität München
Institut für Informatik
Lehrstuhl von Prof. Dr. Manfred Broy

Dr. Klaus Bergner, Prof. Dr. Manfred Broy, Dr. Marc Sihling

Inhalt

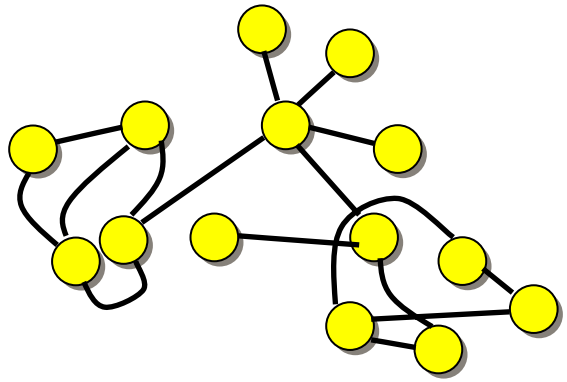
- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Kurze Wiederholung: die 2 Bedeutungen des Begriffs „Softwarearchitektur“

1. *„Eine **Softwarearchitektur** besteht aus einer Menge von Komponenten und ihren Beziehungen untereinander.“*

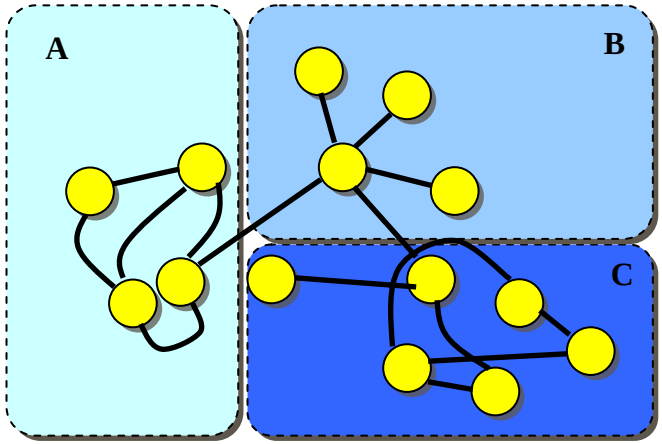
2. *„Die Disziplin **Softwarearchitektur** wird als Teil des Software-Engineerings verstanden und definiert Methoden, Techniken und Werkzeuge zur effizienten Entwicklung qualitativ hochwertiger Softwarearchitekturen.“*

Ein Beispiel:
wie kommt man zur Softwarearchitektur?



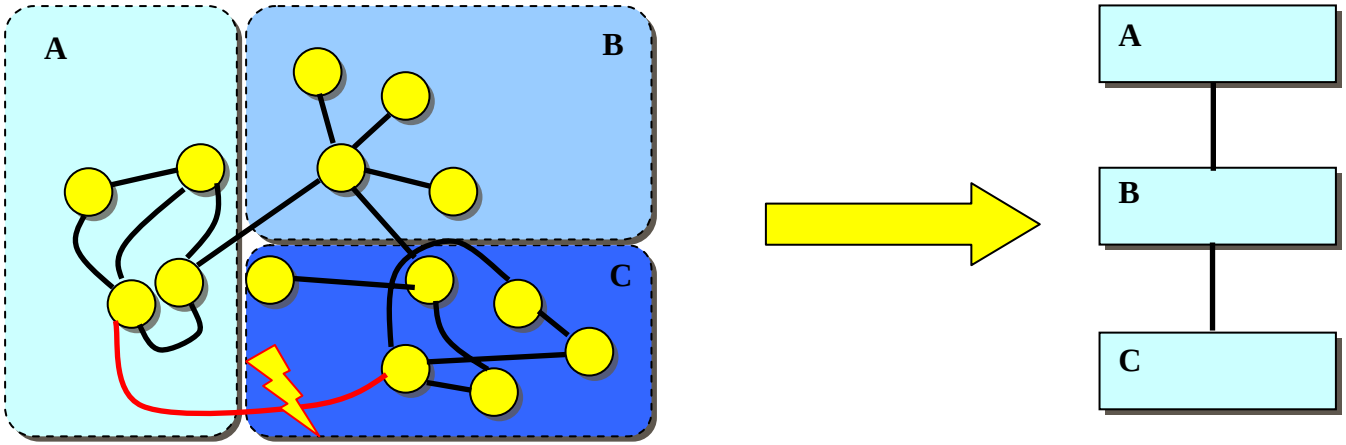
- Ein gängiger Weg, um die Komponenten einer Architektur zu ermitteln besteht in der Betrachtung der Abhängigkeiten

Ein Beispiel:
wie kommt man zur Softwarearchitektur?



- Ein gängiger Weg, um die Komponenten einer Architektur zu ermitteln besteht in der Betrachtung von Abhängigkeiten
- Komponenten „gruppieren“ Elemente mit starker Abhängigkeit

Ein Beispiel:
wie kommt man zur Softwarearchitektur?



- Ein gängiger Weg, um die Komponenten einer Architektur zu ermitteln besteht in der Betrachtung von Abhängigkeiten
- Komponenten „gruppieren“ Elemente mit starker Abhängigkeit
- Abhängigkeiten zwischen Komponenten werden zu Schnittstellen
- Ist die Architektur einmal festgelegt, gibt sie Richtlinien vor für die Einführung neuer Abhängigkeiten (hier: keine Verbindung zwischen den Komponenten A und C)

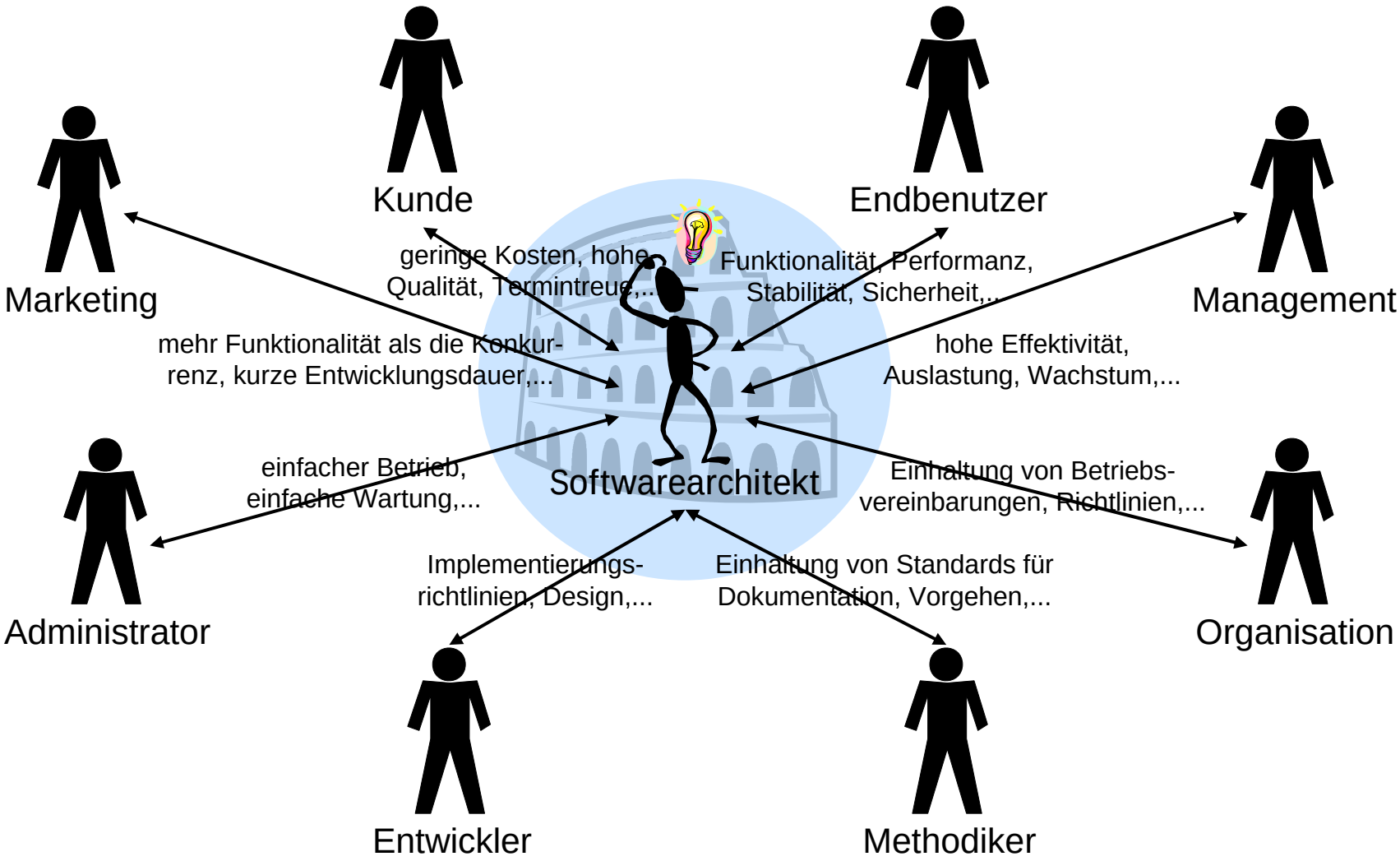
Begrifflichkeiten

- Ähnlich dem Begriff der Softwarearchitektur gibt es eine Fülle teils sehr unterschiedlicher Vorstellungen über den Begriff „Komponente“
- Um präzise über Komponenten reden zu können, definieren wir in der nächsten Stunde ein Systemmodell, das die beteiligten Konzepte formal beschreibt und zueinander in Beziehung setzt
- Bis dahin halten wir fest:
  - Eine Komponente ist ein Baustein des Systems
  - Eine Komponente kapselt ihr Inneres und kommuniziert mit anderen Komponenten über Schnittstellen
  - Komponenten können, müssen aber nicht tatsächlichen Modulen der Implementierung entsprechen. Insbesondere kann man auch Komponenten definieren, wenn es in der Implementierung kein entsprechendes Konzept hierfür gibt.

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

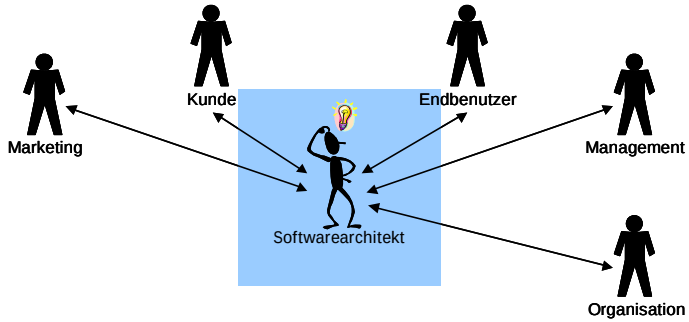
Spannungsfeld Architekturentwurf



Ziel des Architekturentwurfs

- Eine Softwarearchitektur besteht aus einer Menge von Komponenten und deren Beziehungen untereinander.
- Ziel des Architekturentwurfs ist:
  - Den Bauplan des Systems zu erstellen, nach dem sich dann Erstellung, Wartung, Pflege und Weiterentwicklung ausrichten
  - Eine vollständige und prägnante Architekturbeschreibung erstellen
  - Dabei die geforderten funktionalen und nicht funktionalen Anforderungen gewährleisten
- Die Architekturbeschreibung dient als
  - Kommunikations- und Diskussionsplattform
  - Design- und Implementierungsplan

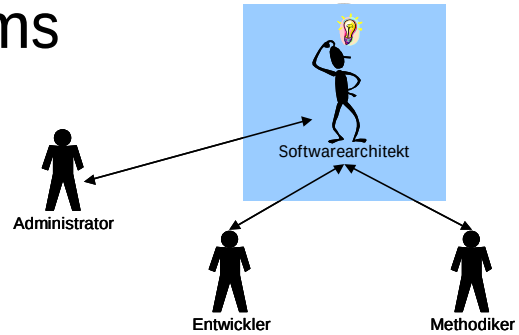
Architekturbeschreibung: Kommunikations- und Diskussionsplattform



- Abstrakte Beschreibung des Systems
- Beherrschbarkeit der Komplexität (divide et impera)
- Zuordnen, Priorisieren und Reflektieren von funktionalen und nicht funktionalen Anforderungen
- Integration von existierenden Lösungen und Systemen
- Abgleich mit den Anforderungen an Systemarchitektur und Hardware
- Erarbeiten, Evaluieren und Bewerten von alternativen Lösungsansätzen

Architekturbeschreibung: Design- und Implementierungsplan

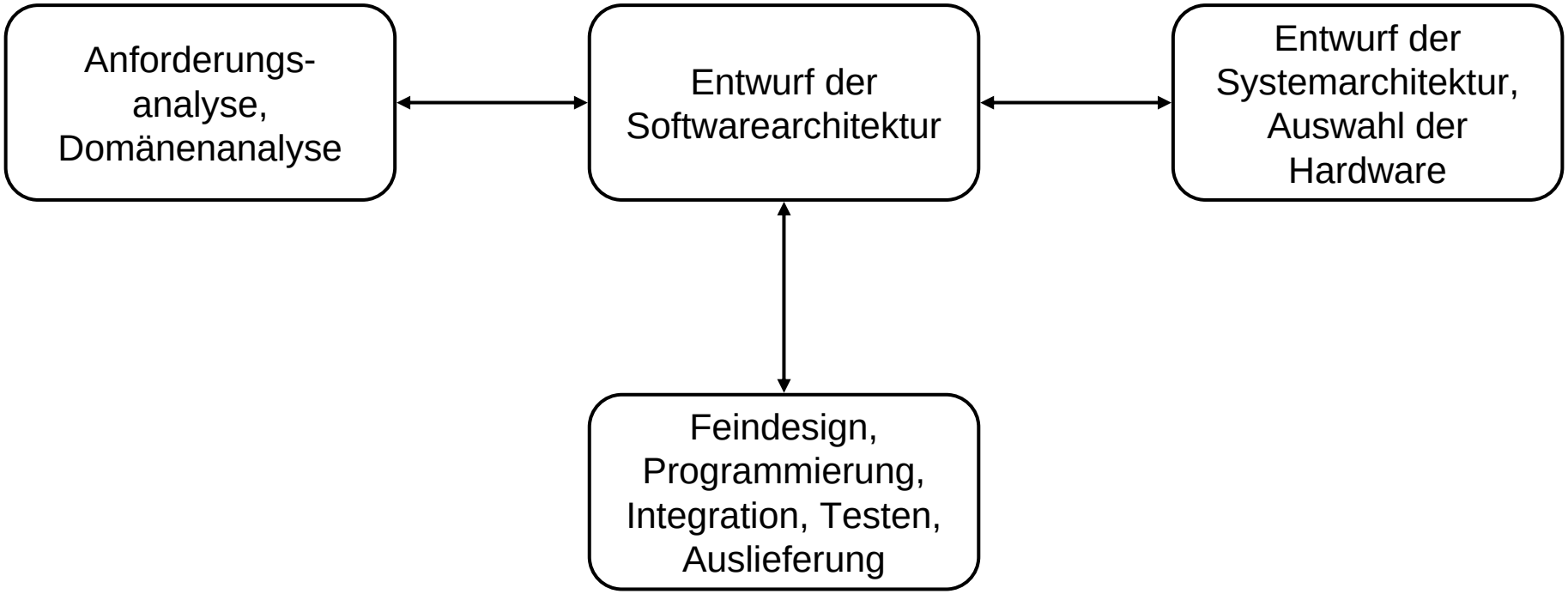
- Festlegung der Komponenten des Systems sowie deren Schnittstellen und Interaktionsmuster
- Integration von neuen Technologien und innovativen Lösungsansätzen
- Erarbeitung, Einführung, Schulung und Überprüfung von Programmierrichtlinien
- Erstellung von Prototypen und exemplarischen Teillösungen
- Wiederverwendung von existierenden Implementierungen und Teilimplementierungen
- Rahmen für die Projektorganisation und -planung



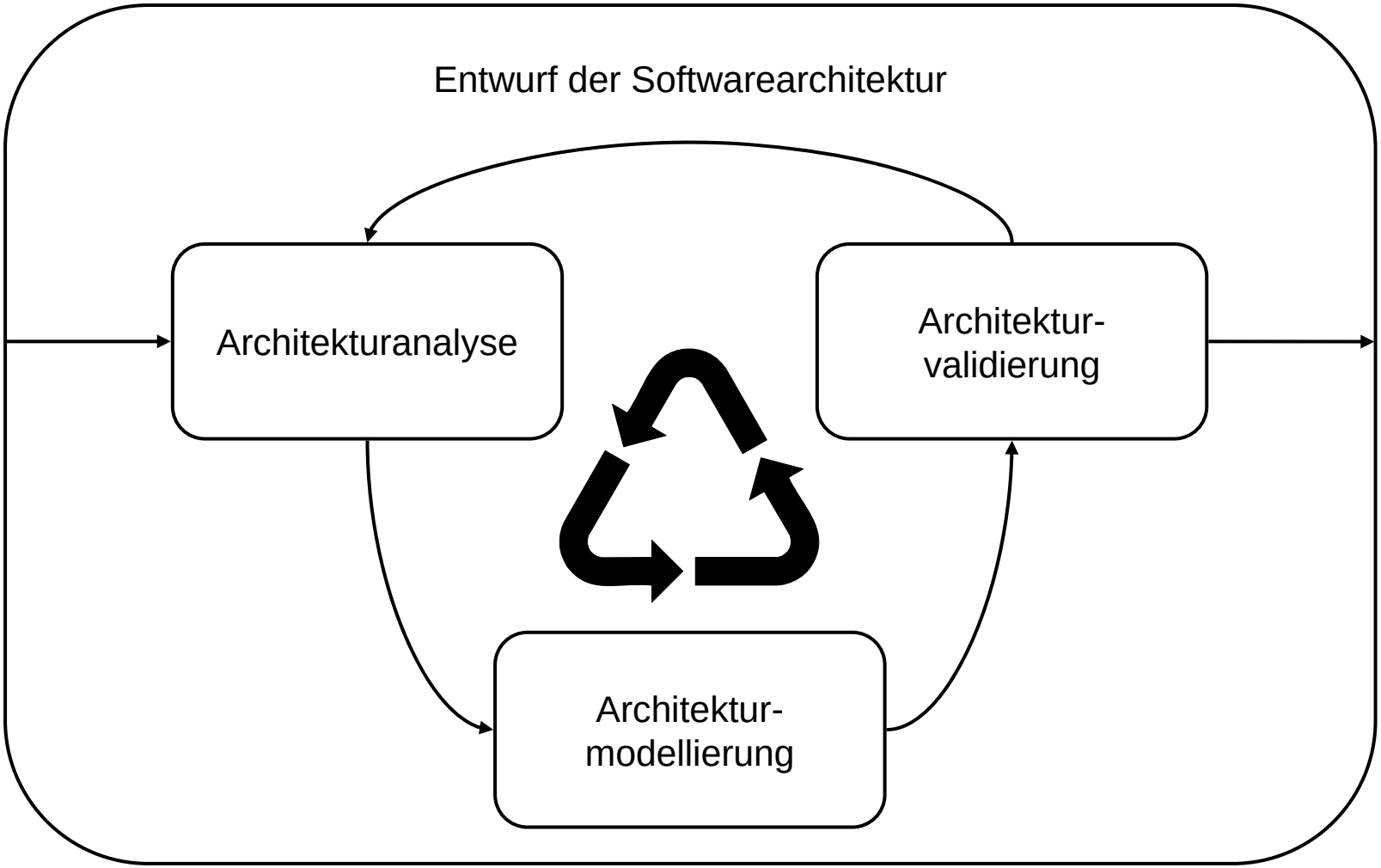
Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

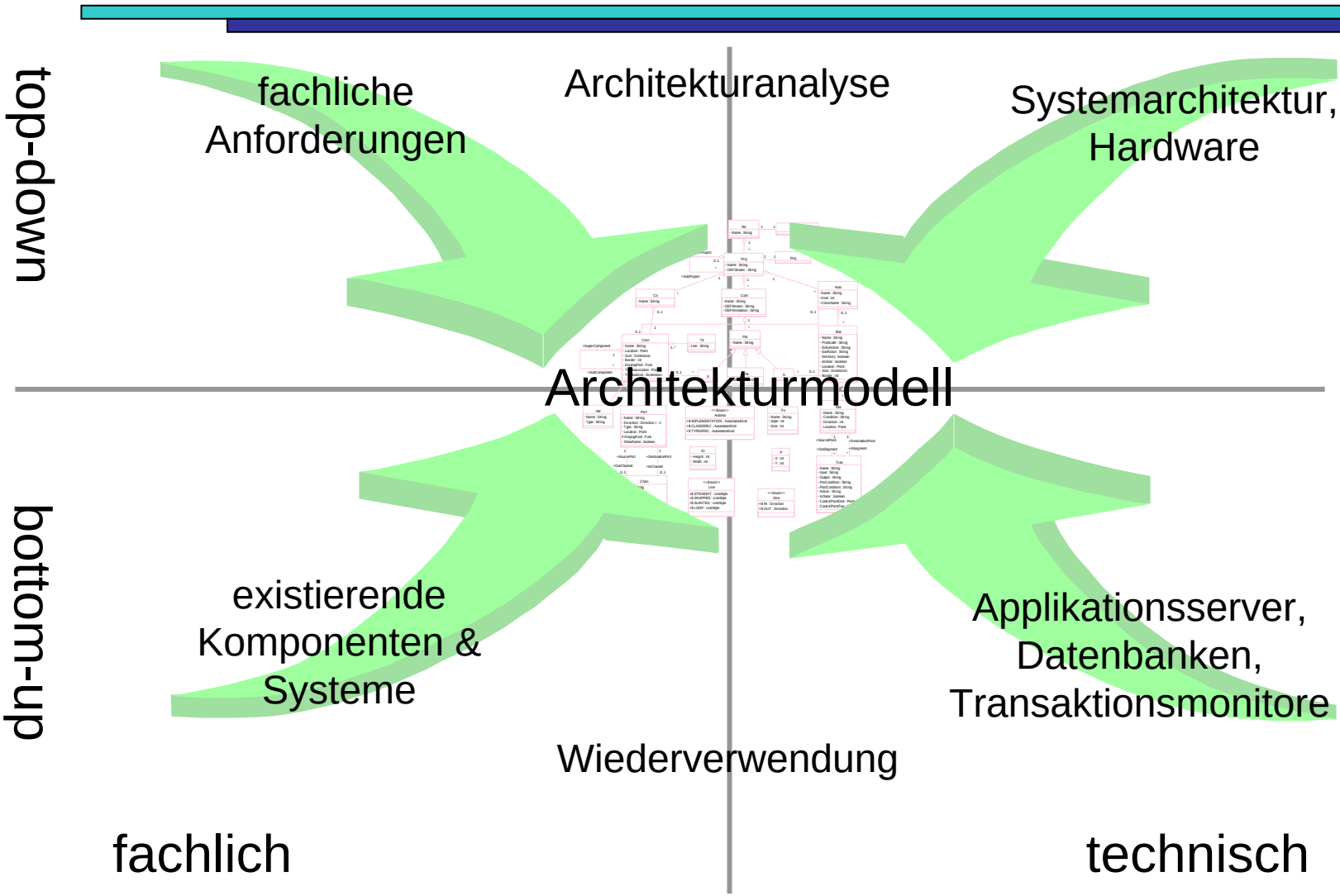
Architekturentwurf im Kontext der SW-Entwicklung



Entwurf der Softwarearchitektur



Architekturmodellierung



Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Ausgangsprodukttypen des Entwurfs: Anforderungen

- Anforderungskategorien
  - Funktionale Anforderungen: Verwaltung von Kunden, Verwaltung von Verträgen, etc.
  - Nicht funktionale Anforderungen
    - Qualitätsanforderungen (IEEE 1061): Benutzbarkeit, Stabilität, Wartbarkeit, Erweiterbarkeit, Performanz, etc.
    - Randbedingungen
      - System / Produkt: Vorgaben zur Verteilung, physikalische Anforderungen, Umweltanforderungen, Materialanforderungen, etc.
      - Technologie: Vorgaben zur Hardware, Vorgaben zur Software, Vorgaben zu Standardtechnologie, etc.
      - Prozess: Vorgaben zum Vorgehensmodell, Anforderungen bzgl. Einführung und Migration, etc.
      - Organisation / Management: Vorgaben bzgl. Zeit und Budget, Vorgaben bzgl. Personal, Vorgaben bzgl. Ressourcen

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Ergebnistypen der Architekturanalyse

- Beschreibung der Architekturanforderungen
  - Identifizieren, Einordnen und Beschreiben der Anforderungen
  - Charakterisierung der Anforderungen bzgl. Flexibilität und Änderbarkeit
  - Auswirkungsanalyse von Anforderungsänderungen
- Beschreibung der Architekturtreiber
  - Identifizieren der zentralen Architekturtreiber und zuordnen von Anforderungen
  - Entwicklung von (alternativen) (Teil-) Lösungen und Strategien bzw. Strategiebausteinen
  - Identifizieren von Strategien und Lösungen die in Beziehung stehen

Beschreibung der Architekturanforderungen

<Name der Anforderung>	
Kategorie:	<Kategorie der Anforderung: funktionale Anforderung, nicht funktionale Anforderung (Qualitätsanforderung), Rahmen-bedingung System/Produkt, usw.; Gegenebenenfalls Verweis auf das Anforderungsanalyse- bzw. Domänenanalyse-dokument>
Beschreibung:	<Kurze Beschreibung der Anforderung>
Priorität:	<muss | soll | kann>
Flexibilität und Änderbarkeit:	<Beschreibung der Flexibilität der Anforderung, d.h. in wie weit akzeptiert der Anforderungsträger, dass diese Anforderung verändert werden. Beschreibung der Änderbarkeit der Anforderung, d.h. in wie weit sind Änderungen dieser Anforderung durch den Anforderungsträger abzusehen.>
Auswirkungen:	<Auswirkungen auf andere Anforderungen und auf die Architektur bzw. auf Teile der Architektur bei Änderungen dieser Anforderung>

Beschreibung der Architekturtreiber

<Name des Architekturtreibers>	
Beschreibung:	<Kurze Beschreibung des Architekturtreibers>
Anforderungen:	<Liste der Anforderungen, die der Architekturtreiber adressiert.>
1. Testszenario:	<Beschreibung eines Anwendungsszenarios um eine korrekte Realisierung des Architekturtreibers zu überprüfen.>
N. Testszenario:	<dito>
Lösung:	<Kurze Lösungsbeschreibung>
1. Strategiebaustein:	<Beschreibung eines Strategiebausteins. Mehrere Strategiebausteine können sowohl alternativ als auch kombiniert verwendet werden.>
N. Strategiebaustein:	<dito>
Verwandte Strategien:	<Liste anderer Architekturtreiber bzw. Strategiebausteinen, die in Bezug zu diesem Architekturtreiber stehen>

Anwendungsbeispiel: eVersicherung 2000

- Neues Informationssystem „eVersicherung 2000“ der „Versicherungs AG“
- Produkte der Versicherungs AG
  - Lebensversicherungen
  - Rentenversicherungen
  - Unfallversicherungen
- Funktionalität der eVersicherung 2000
  - Versicherungsangebote verwalten
  - Versicherungsverträge verwalten
  - Versicherungspolicen verwalten
- Anwender des Systems sind Endkunden, Makler und Mitarbeiter der Versicherungs AG

Beispiel einer Architekturanforderungsbeschreibung

A1: Verwaltung von Kundendaten	
Kategorie:	funktionale Anforderung
Beschreibung:	Das System ermöglicht Benutzern Kunden zu erzeugen, deren Daten (Name, Adresse, etc.) zu bearbeiten, zu suchen und zu löschen.
Priorität:	muss
Flexibilität und Änderbarkeit:	Externe Kundendatenbanken müssen in zukünftigen Versionen des Systems eingebunden werden können.
Auswirkungen:	Die nachträgliche Integration externern Kundendatenbanken kann große Auswirkungen auf die Architektur und somit auch auf das System haben sowie die damit verbundenen Entwicklungssaufwände haben.
A2: Verwaltung von Vertragsdaten	
Kategorie:	funktionale Anforderung
.....	

Beispiel einer Architekturanforderungsbeschreibung

A3: Gleichzeitiges Arbeiten mehrer Benutzer	
Kategorie:	Qualitätsanforderung
A4: Internet-basiertes System mit grafischer Oberfläche	
Kategorie:	Qualitätsanforderung
A5: Benutzer haben Zugriff auf aktuelle Daten	
Kategorie:	Qualitätsanforderung
A6: Time-to-market ist zwei Jahre	
Kategorie:	Rahmenbedingung System/Produkt
A7: Anforderungen sind priorisiert	
Kategorie:	Rahmenbedingung Prozess
A8: Release-Zyklus wichtiger als Funktionalität	
Kategorie:	Rahmenbedingung Management
.....	

Beispiel einer Architekturtreiberbeschreibung

Aktuelle, konsistente, mehrbenutzerfähige Datenbearbeitung	
Beschreibung:	Mehrere Benutzer müssen gleichzeitig mit dem System die Daten bearbeiten und auf aktuelle und konsistente Daten zugreifen können. Auf Datenänderungen anderer Benutzer muss ein möglichst zeitnaher Zugriff möglich sein.
Anforderungen:	A3, A4, A5, A6, A8, A1, A2
1. Testszenario:	Das System stellt dem Benutzer eine Kundenliste zur Verfügung. Der Benutzer wählt einen Kunden zur Bearbeitung aus und ändert den Namen des Kunden. Mit der Bestätigung der Namensänderung, ändert sich auch der Name in der dargestellten Kundenliste.
2. Testszenario:	Benutzer A bearbeitet die Daten eines Kunden. Ein anderer Benutzer B ermittelt gleichzeitig ein Liste mit Kunden deren Versicherungsvolumen ein bestimmtes Limit übertrifft. In dieser Kundenliste taucht auch der Name des Kunden auf, der gerade von Benutzer A bearbeitet wird.
Lösung:	????

Beispiel einer Architekturtreiberbeschreibung

Lösung:	Kombination von Model/View/Controller Muster und Transaktionsmechanismus. Zuerst eine Kombination aus dem 1. und dem 6. Strategiebaustein. Dann von dem 1. zum 2. und später zum 3. Strategiebaustein erweitern, entsprechend dem Architekturtreiber „Kurze Entwicklungszyklen“.
1. Strategiebaustein:	Benutzer kann die Aktualisierung der Daten auf Knopfdruck aktivieren.
2. Strategiebaustein:	In bestimmten Zeitintervallen werden die angezeigten Daten aktualisiert.
3. Strategiebaustein:	Nach einer Datenänderung werden die Daten anderer Benutzer automatisch aktualisiert.
4. Strategiebaustein:	Eine Transaktion pro Benutzer
5. Strategiebaustein:	Eine Transaktion pro Dialog / Fenster
6. Strategiebaustein:	Eine Transaktion pro Anwendungsfall
Verwandte Strategien:	Architekturtreiber: Kurze Entwicklungszyklen; 1. Strategie: Einfaches, inkrementelles Hinzufügen von Funktionalität

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Modelle und Softwarearchitektur

- Modelle sind vereinfachte Abbildungen der Realität, die es erlauben Aussagen im Modell so darzustellen, dass sie von allen Beteiligten gleich interpretiert werden.
- Ein Architekturmodell besteht aus:
  - Menge von elementaren Komponenten
  - Export- und Import-Schnittstellen der Komponenten
  - Komposition der Bausteine zu einer geeigneten Software-architektur
- Ein Architekturmodell soll präzise und vollständig sein:
  - Interaktion und Kommunikation
  - Verbindungen und Strukturen
  - Zustandsraum und Datenstrukturen
- Trotzdem innere Struktur der Bausteine verbergen!

Ergebnistyp der Architekturmodellierung: Architekturbeschreibung bzw. Architekturspezifikation

- Eine Architekturbeschreibung bzw. Architektur-spezifikation beschreibt ein Architekturmodell (meist in Form eines Softwarearchitekturdokumentes)
- Ein Softwarearchitekturdokument sollte enthalten:
  - Entwurfsziele, Richtlinien und Muster der Architektur
  - Abstrakte, grafische Beschreibung mit verschiedenen Sichten
  - Zusätzlicher, erklärender Text
  - Entwurfsentscheidungen und Alternativlösungen
  - Funktionale und nicht funktionale Testfälle
  - Erweiterungsmöglichkeiten
- Ein Softwarearchitekturdokument sollte so einfach wie möglich sein, aber so umfangreich wie notwendig.
- “You know you've achieved perfection in design, not when you have nothing more to add, but when you have nothing more to take away.”

[nach Antoine de Saint-Exupery]

Techniken für den Architekturentwurf

- Ziel des Architekturentwurfs
  - Komponenten identifizieren
  - Schnittstellen der Komponenten beschreiben
  - Komposition der Komponenten beschreiben
- Techniken
  - Anwendung von Heuristiken zur Komponentenidentifikation
    - Hauptwörter in Anforderungen werden zu Entitäten und Verben werden zu Aktivitäten
    - Bündelung von Aktivitäten und Entitäten in Komponenten
  - Von Szenarien zu vollständigen Modellen
    - Von UML-Sequenzdiagrammen zu UML Klassendiagrammen
    - **C**lass **R**esponsibilities **C**ollaborations Workshop, CRC-Karten
  - Wiederverwendung von bewährten Architekturen (vgl. Kapitel „Ausprägungen und Wiederverwendung“)

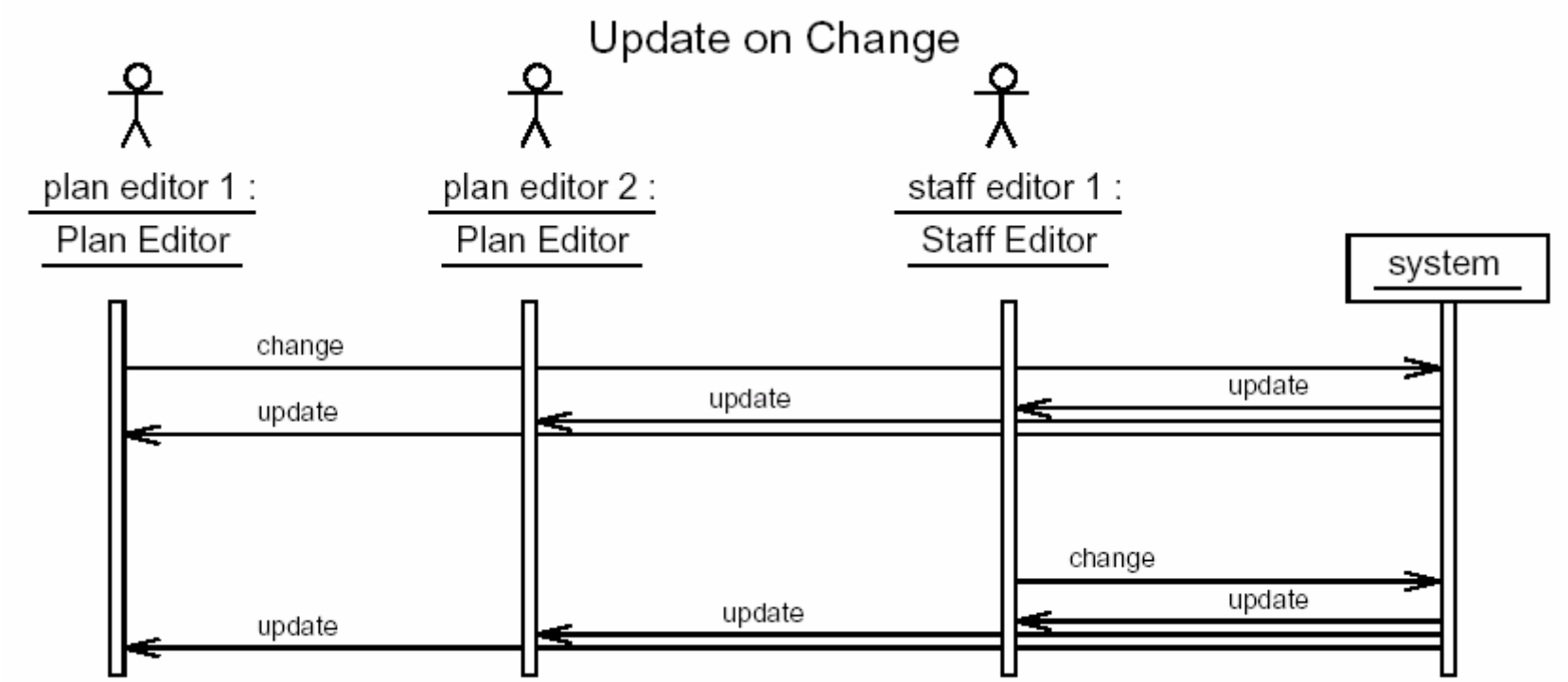
Erfolgsfaktoren des Architekturentwurfs

- In minimalen Schnittstellen und Komponenten denken
- Syntax und Verhalten von Schnittstellen und Komponenten beschreiben
- Komposition der Komponenten zu einer Architektur getrennt beschreiben

- Modularität
- Lose Kopplung
- Abhängigkeiten sichtbar machen
- Zuerst fachliche Sichten, dann technische
- Geeignete Kombination von bottom-up und top-down

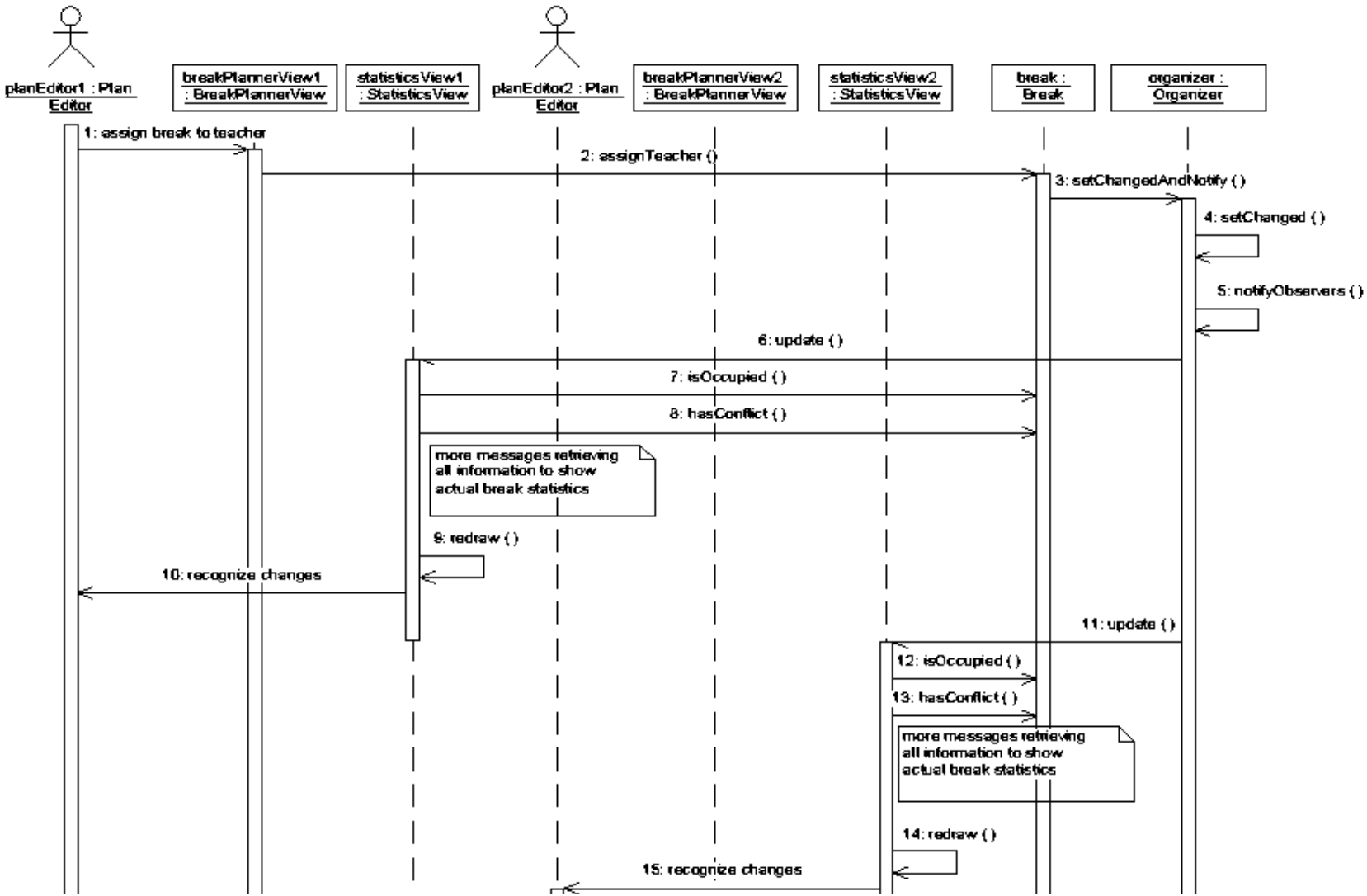
Beispiel: Von Szenarien zu vollständigen Modellen

- 3. Strategiebaustein von Folie 2.22:
  Nach einer Datenänderung werden die Daten anderer Benutzer automatisch aktualisiert.

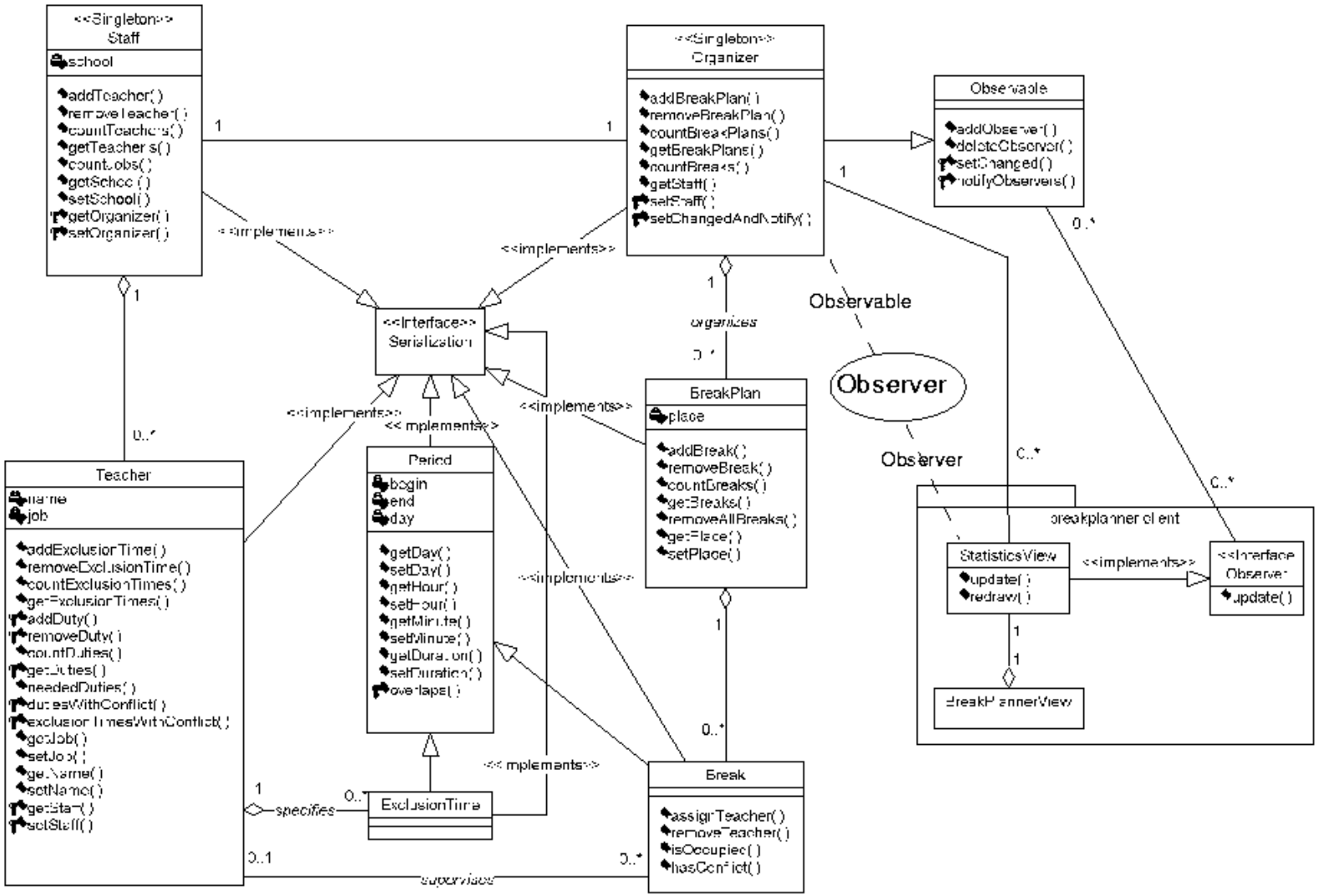


Beispiel aus [BRS97]

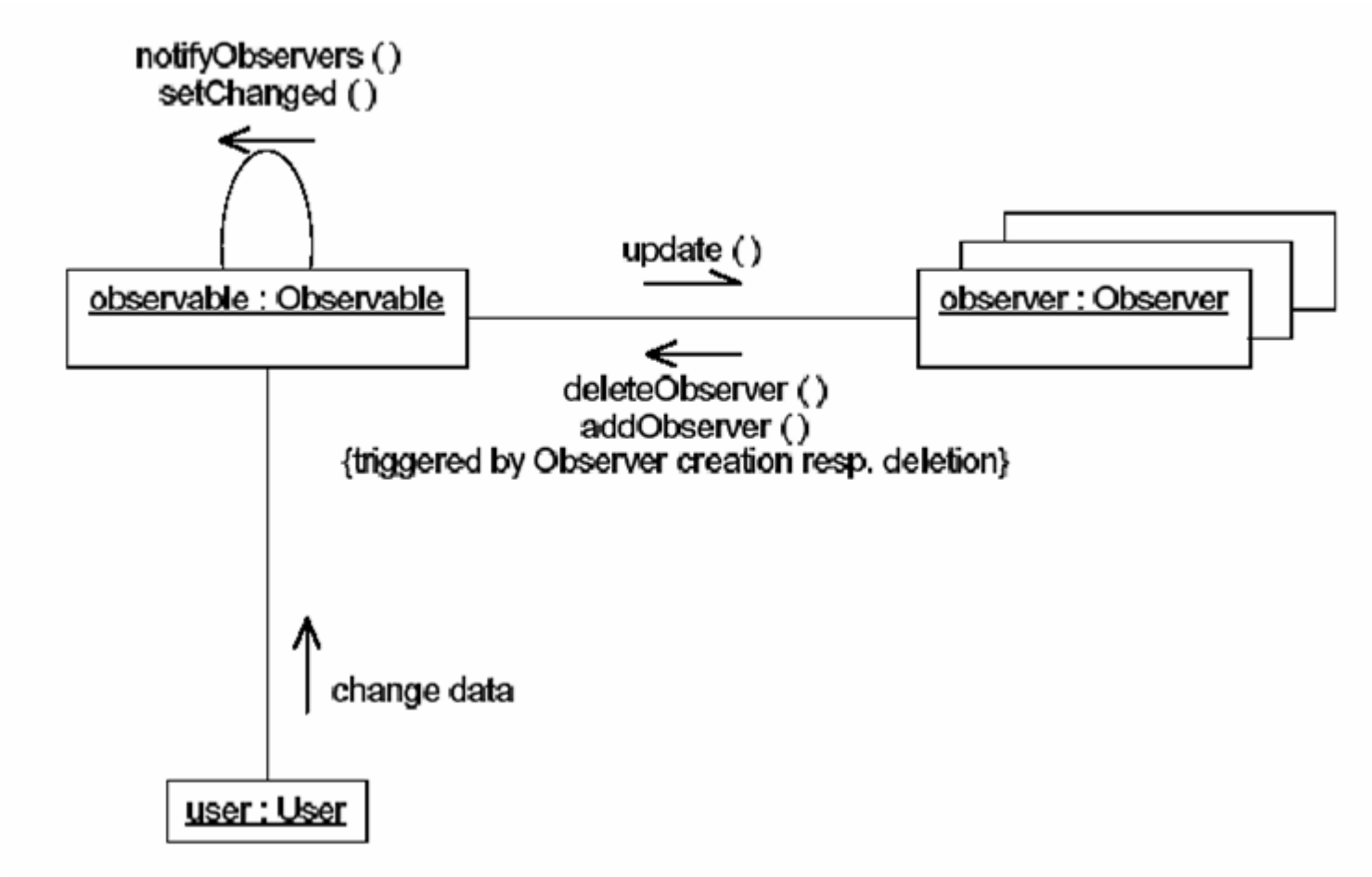
Beispiel: Von Szenarien zu vollständigen Modellen



Beispiel: Von Szenarien zu vollständigen Modellen



Beispiel: Von Szenarien zu vollständigen Modellen



Beispielgliederung eines Softwarearchitektur-dokumentes

- Einleitung und Überblick
  - Motivation und Rahmenbedingen
  - Ziele
  - Anforderungen
  - Architekturtreiber
- Grobarchitektur
  - Überblick über alle Komponenten und deren Beziehungen
  - Beschreibung der Interaktion zwischen den Komponenten
- Beschreibung ausgewählter Sichten und Aspekte
  - Verteilung
  - Fehlermanagement
  - Transaktionsmanagement
  - ...
- Erweiterungs- und Änderungsmöglichkeiten

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Ergebnistypen der Architekturvalidierung

- Reviews
- Modellgetriebene Validierung und Verifikation
  - exemplarische „Ausführung“ der Modelle
  - Modelchecking
  - Beweisen bestimmter Eigenschaften (Theorembeweiser)
- Prototyping
  - experimentelle Prototypen
  - explorative Prototypen
  - evolutionäre Prototypen
- Tests
  - funktionale Tests
  - Lasttests
  - Performance-Tests
- Messungen mit Metriken (siehe [Gil02])

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Der Architekt als Programmierer

- Evaluierung von Entwurfsalternativen mit Hilfe von experimentellen Prototypen
- Einführung und Verbreitung der Softwarearchitektur im Entwicklerteam durch evolutionären Prototypen
- Evaluierung und Einführung von neuen innovativen Technologien und Konzepten
- Mitarbeit im Entwicklerteam als Programmierer u. Tester
- Trainer und Berater des Entwicklungsteams
- Koordination und Moderation der Schnittstellen-festlegung während Feindesign und Implementierung
- Initiierung und Etablierung des Dialogs zwischen den Entwicklern im Team

Der Architekt als Analyst

- Maßgebliche inhaltliche Gestaltung der Systemvision: Festlegung der übergreifenden Ziele, Anforderungen und Rahmenbedingungen des Systems und Architektur
- Abstimmung der Systemanforderungen durch explorative Prototypen
- Wirkungsanalyse, Abstimmung und Review von Anforderungen an das System
- Integration von technischen und konzeptionellen Innovationen in die gemeinsame Systemvision
- Verbreitung und Akzeptanz der Systemvision unter allen Beteiligten: Endbenutzer, Kunde, Entwicklungsteam, etc.

Der Architekt als technischer Projektleiter

- Entscheidungskompetenz für alle wesentlichen Entwurfs-entscheidungen und Veränderungen: Entscheidungen so spät wie möglich und so früh wie nötig treffen!
- Auswahl und Beurteilung der technischen Fähigkeiten von Bewerbern und Mitarbeitern
- Auswahl und Festlegung von Schulungen, Werkzeugen und Technologie
- Schulung und Einführung der Softwarearchitektur im Entwicklerteam
- Motivation, Begeisterung und Integration des Entwicklerteams bzgl. des Architekturentwurfs
- Organisation des Entwicklerteams entsprechend dem Architekturentwurf und dem Projektplan

Der Architekt als technischer Projektleiter

- Zuordnen der konkreten Implementierungsaufgaben zu den Entwicklern
- Delegieren von „lokalen“ Entwurfsentscheidungen an Spezialisten und an das Entwicklungsteam
- Verfolgung und Überprüfung der implementierungs-spezifischen Projektmeilensteine
- Verfolgung und Überprüfung der Qualität des Feindesigns der Implementierung
- Überprüfung der Integrität und korrekten Realisierung der Softwarearchitektur
- Organisation von Wartung und Weiterentwicklung der zentralen Schnittstellen des Systems

Der Architekt als Hüter der Softwarearchitektur

- Überprüfung und Sicherung der Einhaltung der Architekturvorgaben im konkreten Entwicklungsprojekt
- Softwarearchitektur und die Rolle des Software-architekten als Bestandteil der Unternehmenskultur etablieren
- Identifikation, Koordination und Erarbeitung von projektübergreifenden Architekturthemen aus der eigenen Projektlandschaft, Industrie und Forschung
- Definition und Integration des Architekturentwurfs in die firmenübergreifende Projektmanagement- und Qualitätssicherungsstandards

Inhalt

- Umfeld und Ziele des Architekturentwurfs
- Vorgehen beim Entwurf einer Softwarearchitektur
- Ausgangs- und Ergebnisprodukttypen des Entwurfs
  - Anforderungen
  - Architekturanalyse
  - Architekturmodell
  - Architekturvalidierung
- Querschnittliche Aufgaben des Architekten
  - Programmierer
  - Analyst
  - Technischer Projektleiter
  - Hüter der Softwarearchitektur
- Zusammenfassung und Ausblick

Zusammenfassung

- Ziel des Architekturentwurfs ist es eine vollständige und prägnante Architekturbeschreibung zu erstellen, um die geforderten funktionalen und nicht funktionalen Anforderungen zu gewährleisten
- Der Softwarearchitekt ist DIE Brücke zwischen allen Anforderungsträgern und der Softwaretechnik
- Der Entwurfsprozess steht im Zentrum des gesamten Entwicklungsprozesses; Er hat Schnittstellen zu Anforderungsanalyse, Systemarchitektur und Implementierung
- Der Entwurfsprozess ist ein Zyklus aus Analyse, Modellierung und Validierung der Softwarearchitektur
- Die querschnittlichen Aufgaben des Architekten gehen von der Programmierung bis zur Firmenstrategiegestaltung

Literaturhinweise

- [BRS97] Klaus Bergner, Andreas Rausch, Marc Sihling: Using UML for Modeling a Distributed Java Application, Technischer Bericht der Universität München, TUM-I9735, http://wwwbib.informatik.tu-muenchen.de/infberichte/1997/TUM-I9735.ps.gz,1997
- [AF98] Paul Allen, Stuart Forst: *Component-Based Development for Enterprise Systems: Applying The SELECT Perspective*, Cambridge University Press, 1998
- [BCK+98] Len Bass, Paul Clements, Rick Kazman, Ken Bass: *Software Architecture in Practice (Sei Series in Software Engineering)*, Addison-Wesley Publishing, 1998
- [DW98] Desmond Francis D'Souza, Alan Cameron Wills. *Objects, Components, and Frameworks With UML: The Catalysis Approach*, Addison Wesley Publishing Company, 1998
- [HNS99] Christine Hofmeister, Robert Nord, Dilip Soni: *Applied Software Architecture*, Addison Wesley – Object Technology Series, 1999
- [HS99] Peter Herzum, Oliver Sims. Business Component Factory: *A Comprehensive Overview of Component-Based Development for the Enterprise*, John Wiley & Sons, 1999
- [JRL+00] Mehdi Jazayeri, Alexander Ran, Frank Van Der Linden, Philip Van Der Linden: *Software Architecture for Product Families: Principles and Practice*, Addison-Wesley Publishing, 2000
- [Gil02] Tom Gilb: Competitive Engineering, to be appear, www.gilb.com, 2002