Universität Hannover
Institut für Systems Engineering
Fachgebiet System und Rechnerarchitektur

Strukturen und Prozesse großer Entwicklungsprojekte

Denis T. Wirries
Universität Hannover

Strukturen und Prozesse großer Entwicklungsprojekte
2 TV + 1 Ü, 4 CP
`Prüfung:` Mündliche Prüfung
`Prüfer:` Nobert Pich
`Webseite:` http://www.sra.uni-hannover.de/
`Lernziele:` Die Veranstaltung behandelt das Thema „Strukturen und Prozesse großer Entwicklungsprojekte" aus theoretischer und aus praktischer Sicht. Es ist das Ziel dieser Veranstaltung, Ihnen verständlich zu machen, wo überhaupt die Probleme bei der Planung und Umsetzung großer Entwicklungsprojekte liegen, und welche theoretischen Ansätze und Methoden Hilfe versprechen. Ich will außerdem versuchen Ihnen zu erläutern, was man in der Praxis tun muss, so dass der theoretisch vorausberechnete Erfolg dann auch tatsächlich eintritt.

`Stoffplan:`
Die Veranstaltung gliedert sich in 13 Lehr- und Lerneinheiten.

- Die Rahmeneinheiten 1 und 13 geben einen Überblick über die Themenstellung bzw. einen Rückblick auf den erzielten Erkenntnisgewinn.
- Die Einheiten 2 bis 5 widmen sich im Schwerpunkt dem Durchdringen und Verstehen der Gesamtproblematik und dem Verständnis der vielschichtig zu meisternden Herausforderungen.
- Die Einheiten 6 bis 12 widmen sich im Schwerpunkt dem Kennenlernen und beispielhaften Anwenden von geeigneten Methoden zur Bewältigung ausgewählter Aspekte des Themas.

`Vorkenntnisse:` Software-Technik I + Software-Qualität
`Literaturempfehlungen:`

- Tom Demarco, „Der Termin" als Taschenbuch preiswert. Projektmanagement leicht verpackt
- GoF96; Entwurfsmuster Elemente wiederverwendbarer objektorientierter Software; Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John; Addison-Wesley Deutschland GmbH, Bonn; 1. Auflage 1996, (nur das erste Kapitel ca. 25 Seiten, wichtig ist hier zu versthen was ein Muster ist und wie man es strukturiert aufschreibt)
- Booch96; Objektorientierte Analyse und Design; Booch, Grady; Addison-Wesley (Deutschland); 2. Aufl. 1996, (nur den ersten Teil die ca. ersten 120 Seiten, wichtig ist, sich in das objektorientierte Gedankengut und Grundverständnis einzulesen)
- Einen Film anschauen: Vorschlag „Italian Job" (das Remake), oder alt, aber dennoch gut „Der Clou" mit Paul Newman.
- Wird noch ergänzt um kleine Papers, die ich als PDF mitbringen werde.
- Becker95+: Grundsätze ordnungsgemäßer Modellierung; Becker, Jürgen, Rosemann, M. Schütte, R., In: Wirtschaftsinformatik 37 (5/1995), .S. 435 - 445: 199 guter Aufsatz, Titel gibt Inhalt gut wieder
- Jacobson92+, Object-Oriented Software Engineering A use case driven approach; Jacobson, Ivar; Christerson, M.; Jonsson, P.; Övergaard, G.; ACM Press, Addison Wesley Publishing Co., Reading MA; 1992, Wichtig für vertieftes Verständnis der Objektorientierung, hier wichtig wegen "Layer" Architekturen. Einige der zentralen Inhalte vermitteln wir in der Veranstaltung
- Jacobson97+; Software Reuse; Jacobson, Ivar; Griss, Martin; Jonsson Patrik; ACM Press Addison Wesley Publishing Co., Reading MA, 1997; Hier ist Wiederverwendung die zentrale Idee. Die Idee ist auch für uns wichtig, weniger aber die programmtechnische Umsetzung

- Luhmann99; Soziale Systeme, Grundriss einer allgemeinen Theorie, Suhrkamp Taschenbuch Wissenschaft 666, 7. Auflage Frankfurt am Main 1999, Luhmann erfindet die Systemtheorie neu, sehr schwer zu lesen, aber Quantensprung in der systemtheoretische Denkwelt.

- Sche98; ARIS vom Geschäftsprozeß zum Anwendungssystem; Scheer, August-Wilhelm; Springer Verlag; 3. Auflage, 1998, Geschäftsprozesse sind die zentralen Grundlagen zur Formulierung von Anforderungen. ARIS ist ein Standard in der Industrie zu Dokumentation von Geschäftsprozessen.

`Über dieses Dokument:`

Dieses Dokument ist für Nachbereitung der Vorlesung Strukturen und Prozesse großer Entwicklungsprojekte von Herrn Pich im Wintersemester 2005/2006 entstanden. Es dient als Prüfungsvorbereitung und Nachschlagewerk.

Dieses Dokument ist unter http://www.netpage-germany.de erhältlich. Anmerkungen und Korrekturen (auch Rechtschreibfehler) bitte an skript@netpage-germany.de.

Erstellt mit LaTeX.

Stand: **1. März 2006**

1 Software-Engineering

Definition 1 (Software-Engineering) *Software Engineering (SE)*

- *(1) the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software*
- *(2) the study of approaches as in (1) (IEEE Std 610.12 - 1990)*

Folgerung 2 (Folgerungen aus der Definition)

- *F.L.Bauer: „Anwendung von Ingenieursprinzipien für zuverlässige SW, die auf realen Rechnern läuft und mit wirtschaftl. Mitteln erstellt wird.“*
- *offenbar ist also Engineering: „ a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of X“*
- *Parnas: „SE erst ab 2 Personen und 2 Versionen“*

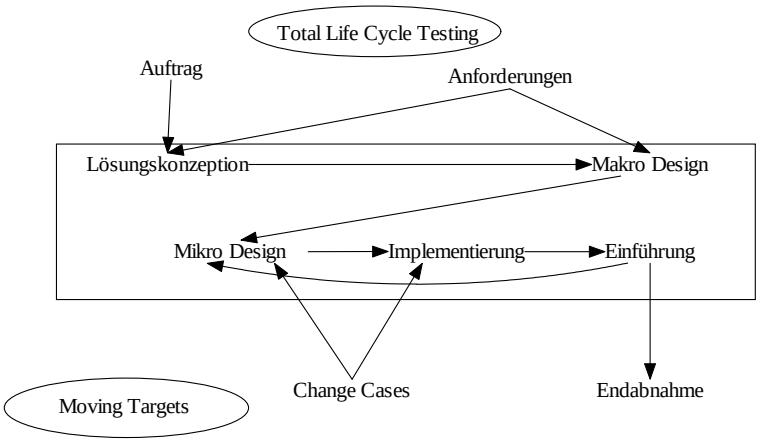
Definition 3 („Ingenieursprinzipien“) *„ a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of X“*
***Konsequenzen**:*

- *Kostendenken*
- *Qualitätsbewußtsein*
  - *nicht nur subjektiv definiert*
- *Anwendung von Normen und Regeln*
  - *keine Künstler*
- *Baugruppen und Wiederverwendung*
  - *statt „not invented here“*
- *Probleme durch Zerlegung lösen*
  - *„divide et impera“*

2 Projekt

Definition 4 (Projekt) *Ein **Projekt** ist eine **zeitlich begrenzte Unternehmung**, das durchgeführt wird, um ein einmaliges Produkt oder Dienstleistung zu erstellen. Projekte können **vielfältiger Komplexität, Dauer, Umfang und innerer Struktur** sein.*
*Ein Projekt ist ein **einmaliger Prozess**, der aus einem **Bündel von koordinierten und kontrollierten Aktivitäten** besteht mit **definierten Start- und Endterminen**. Es wird durchgeführt, um ein **Ziel zu erreichen**, das bestimmten **Anforderungen** gerecht wird. Zu diesen Anforderungen gehören auch **Restriktionen** wie Termine, Kosten, Ressourcen[1]. Projekte unterscheiden sich vom andauernden wiederholbaren Betrieb / Produktion durch ihre **Einzigartigkeit** und **zeitliche Begrenztheit**.*
Zusätzlich kann ein Projekt aber auch wiederholbar sein, bzw. Teile wiederverwenden. Außerdem steht es in Konkurrenz zu anderen Projekten um Resourcen.

2.1 Projekt als Lebenszyklus



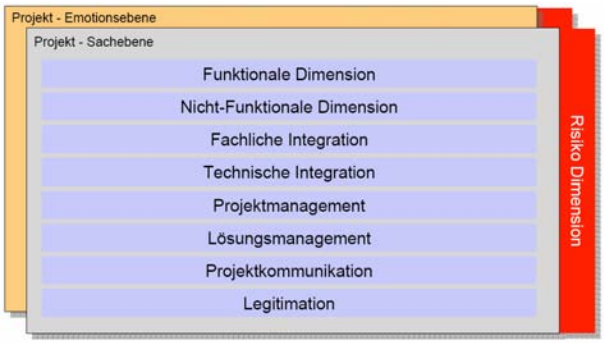
Begriffserläuterungen:

- **Lösungskonzeption**: das Business Problem ist verstanden und die Lösung ist im groben klar.
- **Makro Design**: das Business Problem ist vollständig verstanden, die fachliche und technische Lösung für alle Releases ist klar. Die Business Architektur steht. Ein fachlicher Stufenplan einschließlich Einführung und Migrationsstrategie ist erstellt. Die technischen Grundsatzentscheidungen sind getroffen.
- **Mikro Design**: Die technische Feinspezifikation für den Release ist abgeschlossen, so dass die Implementierung beginnen kann. Alle technischen Architekturentscheidungen sind getroffen. Die Entwicklungsumgebungen sind betriebsbereit.
- **Implementierung**: technische Umsetzung für den ersten Release

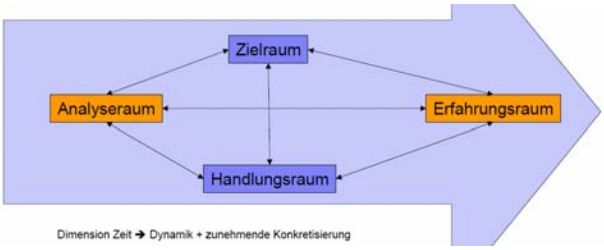
[1] ISO 10006

- **Einführung**: technische, organisatorische und emotionale Einführung des ersten Releases einschließlich Migration.
- **Release**: selbständig produktionsfähige Einheit eines Projektes

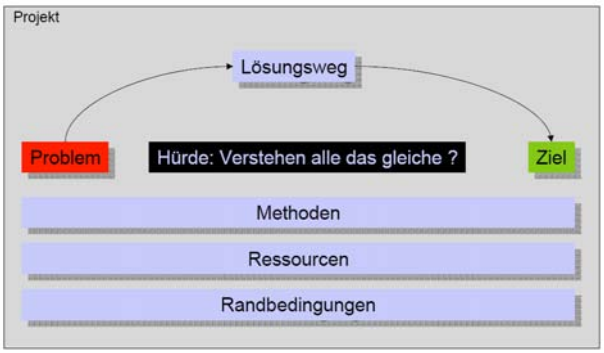
2.2 Projekt als Analyseraum



2.3 Projekt als Managment Aufgabe



2.4 Projekt als Handlungsraum



3 Anforderungen

Definition 5 (Anforderung) *Eine Anforderung kann zwei verschiedene Grundbedeutungen haben: etwas* ***fordern, verlangen oder voraussetzen*** *(siehe Forderung, Voraussetzung), z.B.* ***Eigenschaften, Fähigkeiten*** *usw. (siehe auch Mindestanforderung, notwendige und / oder hinreichende Bedingungen) oder etwas bestellen (im Sinne von die Anlieferung verlangen).*
In der (Software-)Technik ist eine Anforderung (häufig engl. requirement) eine Aussage über eine zu ***erfüllende Eigenschaft*** *oder zu* ***erbringende Leistung*** *eines Produktes, Systems oder Prozesses. Anforderungen werden üblicher Weise in einem Lastenheft zusammengefasst.*

Definition 6 (Requirements Engineering) *Requirements Engineering ist selbst schon wieder ein so kompliziertes Fach, von dem es mehrere Gliederungen und Strukturierungen gibt. Das hier gezeigte Modell (siehe Abb. 1 von Prof. Dr. Schneider, WS03/04) hat DaimlerChrysler auf einem Dagstuhl-Workshop gezeigt. Es unterscheidet zunächst zwischen der* ***Analyse*** *und dem* ***Management*** *von Anforderungen. Beide Begriffe werden dann noch feiner unterteilt.*
Andere Modelle verwenden ähnliche Begriffe, unterscheiden sich aber in der genauen Abgrenzung, auch von Req. Management und Analysis.

3.1 Nicht-funktionale Anforderungen

technische	**fachliche**
Performance	Änderbarkeit
Kapazitäten	Flexibilität
Skalierbarkeit	Mehrwährungsfähigkeit
Wartbarkeit	Mandantenfähigkeit
...	...

Nicht-funktionale Anforderungen können sich auch aus Sicherheits-, Qulitätsaspekten und / oder Randbedingungen ergeben. **Wichtig**: Zu jeder nicht-funktionalen Anforderungen gehört auch ein Metrik!

3.2 Wo befinden sich Anforderungen

- Definition der Feldgrenzen
  - Quelle 1: Projektdefinition

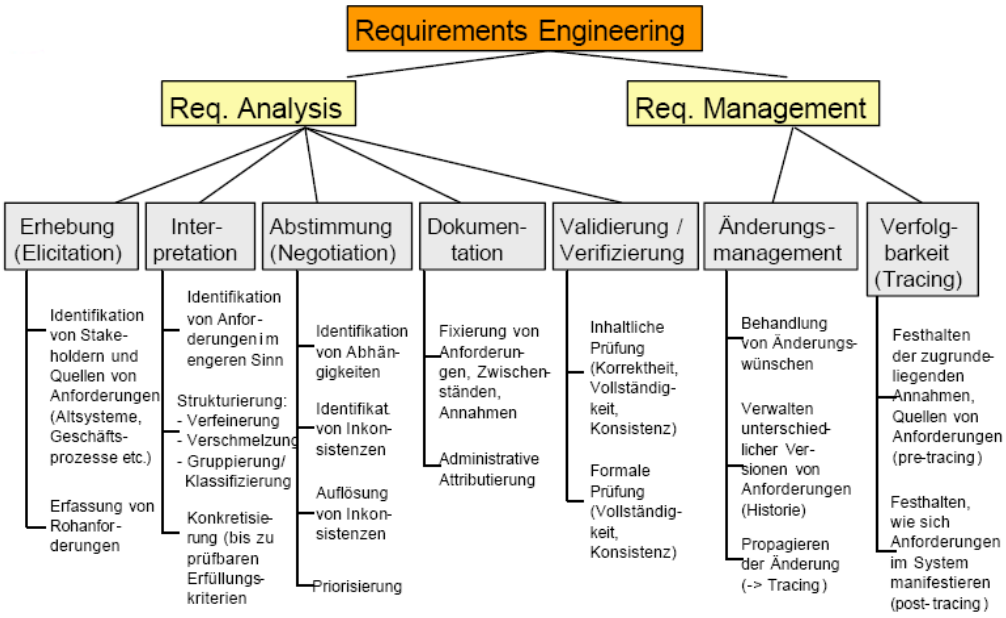


Abbildung 1: Aufteilung des Requirements Engineering

  - Quelle 2: Analyse der Benutzergruppen
  - Quelle 3: Wettbewerbsanalyse
  - Quelle 4: Strategieanalyse

- Sammlen und Ordnen von Dokumenten
  - Use Cases
  - Prozessbeschreibungen
  - Arbeitsweisen
  - Beschreibungen
- Basiserhebung
  - Befragungen
  - Prozessanalyse
  - Laborversuche

Basis - Dokumentation

- Use Cases
- GOM bzw. BOM
- Geschäftsregeln
- Komponenten

Definition 7 (Use Case) *Ein **Use Case** beschreibt eine **Menge von Aktivitäten eines Systems** aus der **Sicht seiner Akteure**, die für die Akteure zu einem **wahrnehmbaren Ergebnis** führen.*
Use Cases stellen die wesentlichen Geschäftsprozesse dar, die automatisierbar sind. Anwendungsfälle werden stets von Akteuren ausgelöst. Ein Anwendungsfall ist eine komplette, unteilbare Beschreibung.

Definition 8 (Change Cases) *Ein **Change Case** ist eine **dokumentierte Änderung, Erweiterung oder auch Reduzierung einer Anforderung**, die einen relevanten Einfluss auf das Projekt hat. Die Akzeptanz eines Change Cases setzt einen Entscheidungsprozess voraus. Die Grundlage eines solchen Entscheidungsprozesses ist in der Regel eine Kosten-Nutzen-Analyse. Ein akzeptierter Change Case führt in der Regel zu Änderungen im Projektplan.*

Definition 9 (Akteur) *Ein **Akteur** ist eine **Rolle**, die ein Benutzer des Systems spielt, der an der im Anwendungsfall beschriebenen **Interation** beteiligt ist.*
Ein Akteur kann eine reale Person aber auch ein anderes System sein. Der Akteur beinflußt das System. Akteure liegen außerhalb des Systems.

Definition 10 (Geschäftsregel) *Eine **Geschäftsregel** ist eine **fachliche Regel**, die im Modell des Geschäftssystem und im Modell eines Anwendungssystems **abgebildet** wird.*
Fachliche Regeln sind Geschäftsstrategien, Anforderungen, fachliche Vorgaben / Restriktionen. Geschäftsregeln sind der wesentliche Bestandteil des Aufbau von Organisationen. Sie schreiben vor, wie die Geschäftsabwicklung zu erfolgen hat. Befinden sich z.B. in Workflows.

Definition 11 (Geschäftsprozess) *Der **Prozess** wird verstanden als **zeitliche und sachlogische Abfolge von Funktionen**, die zur Bearbeitung eines betriebswirtschaftlich relevanten Objektes notwendig sind. **Geschäftsprozesse** werden als **Prozesse** verstanden, die die **wesentlichen Aktionen der Unternehmung umfassen**. Dabei unterscheidet man noch die Kernprozesse und die Service- oder Support-Prozesse.*

Definition 12 (Geschäftsbegriff) *Ein **Geschäftsbegriff** definiert die **verbindliche Benennung eines Objektes** in einem bestimmten Fachgebiet.*
Die Gesamtheit der Geschäftsbegriffe wird als Terminologie bezeichnet. Ein Glossar (Wörterliste) enthält die definiten Fachwörter einer bestimmten Terminologie.

Definition 13 (Geschäftsobjekt und -attribut) *Ein **Geschäftsobjekt** stellt die **relevante Struktur** des zukünfigen Systems dar. Ein Geschäftsobjekt ist ein **abstraktes Objekt** innerhalb einer Anwendung und stellt eine **Einheit der realen Welt** dar. Ein **Geschäftsobjekt-Attribut** ist ein **fachliches oder technisches Merkmal** eines Geschäftsobjekts.*
Das Geschäftsobjektmodell (GOM) oder Business Object Modell (BOM) benutzt Geschäftsobjekte, um Prozesse zu beschreiben. Es enthält so Geschäftsobjekte mit ihren Attributen.

Definition 14 (Komponente) ***Komponenten*** *sind* ***eigenständige wiederverwendbare Bausteine*** *mit wohldefinierten* ***Schnittstellen*** *und* ***vollständig gekapselter****, nach außen unsichtbarer Implementierung.*

Definition 15 (Komponentenfunktion) *Eine* ***Schnittstellenfunktion*** *definiert eine* ***Aufgabe****, welche die zugehörige Komponente erledigen kann.*
Die Funktion enthält den Namen, eine Beschreibung mit Aufgabe, die Signatur, sowie die Vor- und Nachbedingungen.[2]

Definition 16 (Schnittstellenparameter) *Ein* ***Schnittstellenparameter*** *ist die* ***Definition von Eingabe- bzw. Ausgabe-Parametern*** *einer Schnittstellenfunktion.*

3.3 Testen von Anforderungen

- Fachliche Konsistenzaspekte
  - Fachinhalt
  - Fachlogik
  - Prozesskonsistenz
- Formale Konsistenzaspekte
  - Abstraktionsebenen (Level-Konsistenz)
  - Link Konsistenz
  - Semantische Konsistenz
  - Temporale Konsistenz
    - z.B. Gültigkeitsdauer der Aussagen

4 Prozesse

Definition 17 (Wesen des Prozesses)

1. *Folge von Aufgaben*
2. *Zeitdimension*
3. *Arbeitsteilung*
4. *Definiertes Ergebnis*
5. *(mit Einschränkungen) Wiederholbar*
6. *Gestaltbar*

Übergreifende Prozesse: „Just-In-Time", Supply Chain Management, Business Transformation Outsourcing.

[2] Formal gesehen ist dies ein eigener Arbeitsergebnistyp.

Definition 18 (Management eines Prozesse)

- *Zieldefinition*
- *Überwachung der Zielrichtung*
- *Arbeitsschnittdefinition*
- *Überwachung der Arbeitsschnittergebnisse*
- *Prozesskoordination*
- *(Ablauf-)Organisation*
- *Resourcenzuteilung*
- *Skalierbarkeit*
- *Nachweisbarkeit*
- *Skill Management*
- *Flexibilität*

siehe dazu auch Abbildung 2 (CMM)

Definition 19 (CMM) *Capability Maturity Modelle sind Rahmenkonzepte für die gesamte Projektstruktur.*

- *Meta-Modelle, die die elementaren Prozesse und Aktivitäten definieren.*
- *1-5 Stufen des Reifegrades*
- *Zertifizierung möglich*
- *1986 am Software Engineering Institute Pitsburgh, USA, entwickelt*

Auswirkungen (siehe Abbildung2)

- *realistischer Schätzungen (Level 2)*
- *verringerter Schwankungsbreite (Level 3)*
- *verbesserte Effizienz (Level 2 - 5)*

Definition 20 (Entwicklungsprozess im Projekt)

1. *Zielfindung*
2. *Zielvereinbarung*
3. *Planung*
4. *Aufbau physischer Infrastruktur*
5. *Programmieren*
6. *Testen*

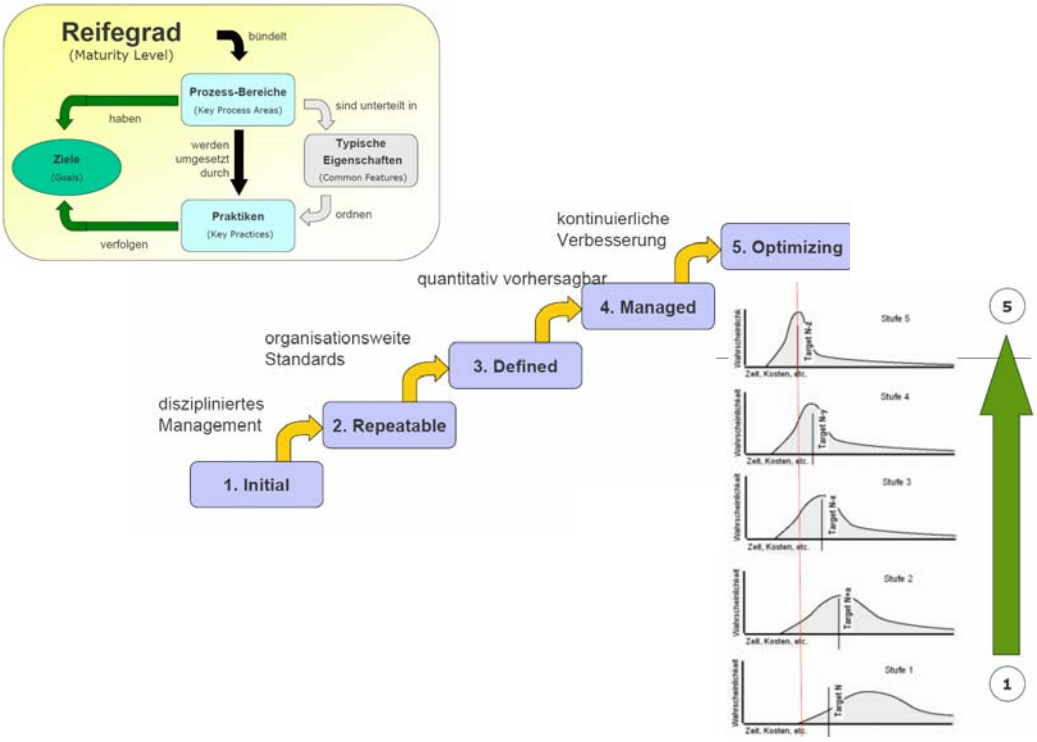
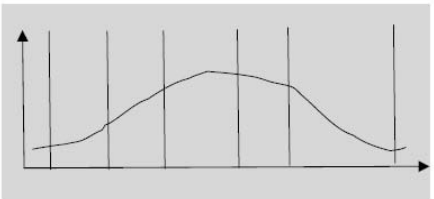


Abbildung 2: Abbildung CMM

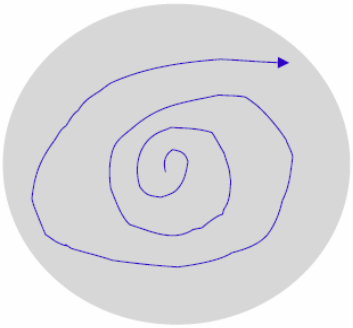
7. *Einführung*
8. *Betrieb*

- *Geschäftsprozess*
- *Lernen*
- *Kommunikation*
- *Konfliktmanagement*
- *Identifikation*
- *Teamarbeit*

Definition 21 (Ansätze für Entwicklungsprozesse) *Man unterscheidet generell zwischen „Phasen" orientierten Ansätzen*



und „Prototyoing (Iterative) orientierte Ansätze.



Definition 22 (Prozess des „Verbindlichmachens”)

- *Etablierung von Entscheidungsprozessen siehe Abbildung3*
- *Grundlegenende Vorgehensweise vereinbaren*
- *Typisierung der Arbeitsergebnisse als Aufgabe durchsetzen*
  - *Fachliche Arbeitsergebnistypen*
  - *Projetmanager Arbeitsergebnistypen*
- *Ergebnis der Typisierung verbindlich machen*
- *Termine und Resourcen zusagen*
- *Testverfahren und Testkriterien festlegen*
- *Überprüfen der Ergebnisse der Entscheidungsprozesse (begleitend)*
- *Testdurchführung*
- *Schließlich Abnahmetest durchführen und Verantwortlichkeit übergeben*

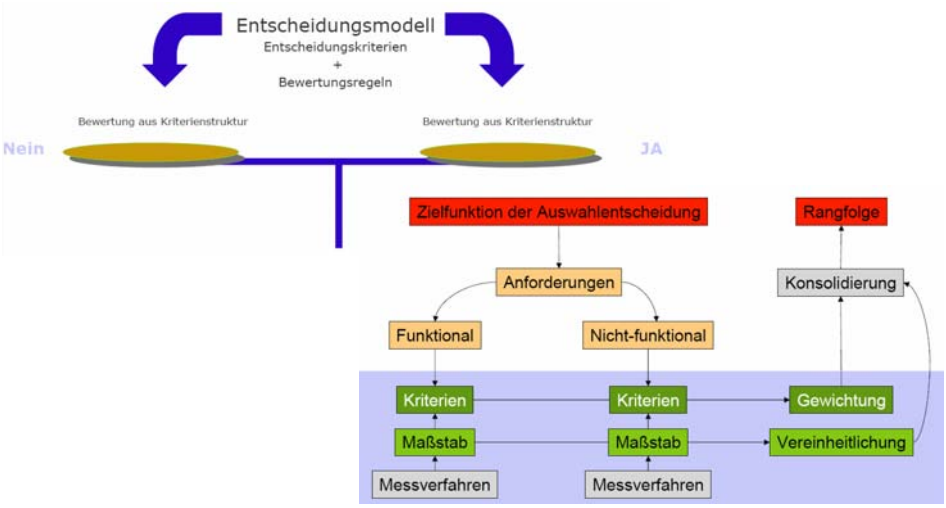


Abbildung 3: Entscheidungsprozess

4.1 Integration

- Erstellung von Schnittstellenkomponenten zu den anderen Systemen außerhalb von Projektes = Randsystem (siehe auch System Kontext Diagramm)
- Wichtige Arbeitsergebnisse
  - Use Cases für Integration
  - Nicht-funktionale Anforderungen
- Eine Schnittstellenkomponente besteht idealtypisch aus 4 Teilen:
  - Genutzte unveränderte externe Schnittstelle(n) des Randsystems
  - Verbindung zu anderen Systemen (Netzebene, Protokollebene, Datenebene,Transaktionsebene)
  - Technisch Aufnahme im eigenen System
  - Virtuelles Geschäftsobjekte die das Randsystem im eigenen System repräsentiert
- Keine Verantwortung, aber Mitwirkung bei:
  - Komponentenmodell
  - Gesamtarchitektur

5 Arbeitsergebnisse

Arbeitsergebnis:
definierte abnehmbare Projektaufgabe (Ausprägung, Instanz)

Arbeitsergebnistyp:
wiederverwendbare inhaltliche Beschreibung für Struktur und Form einer zu erbringenden und definierten Projektaufgabe (Vorlage)
zusätzlich noch ein Benutzerleitfaden, Typisierungsvereinbarungen und Integration mit anderen Arbeitsergebnistypen

Arbeitsergebnisnetz:
Zusammenhang (fachlicher und / oder zeitlicher) der zu erstellenden Arbeitsergebnisse, kein Projektplan!

6 Testen

- Erarbeiten und Fortführen einer Gesamtteststrategie über die ganze Laufzeit des Projektes
- Umsetzung der Teststrategie in Testprozesse (echte Herausforderung!)
- Erarbeitung eines Testplanes für das Gesamtprojekt
- Laufende Planung und Kontrolle aller Testprozesse

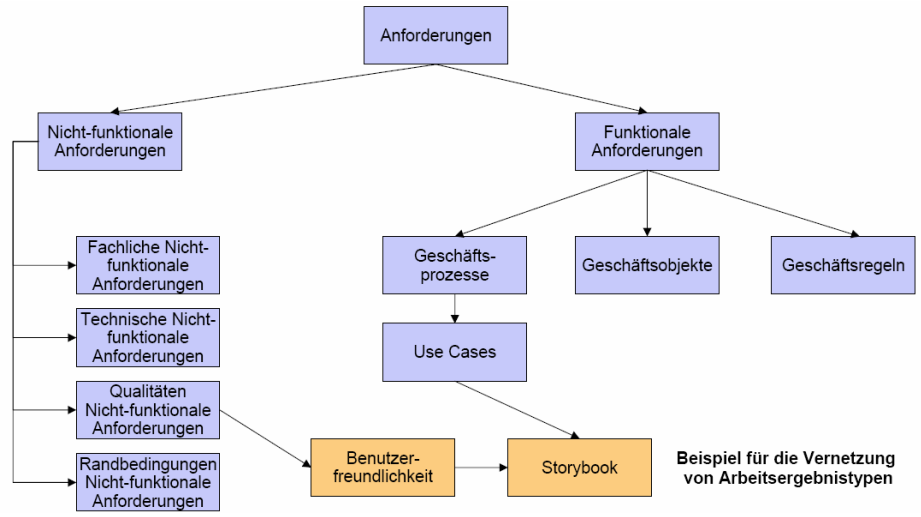


Abbildung 4: Typisierung von Arbeitsergebnissen auf oberster Ebene: Hierarchie von Arbeitsergebnissen

- Grundprinzipien
  - Philosophie des „Full Life Cylce Testing" oder „Total Life Cycle Testing"
  - 4 Augen Prinzip
  - Testen nur nach definierten Testkriterien und -verfahren
- Klare organisatorische und fachliche Verankerung im Projekt unter Wahrung der Unabhängigkeit

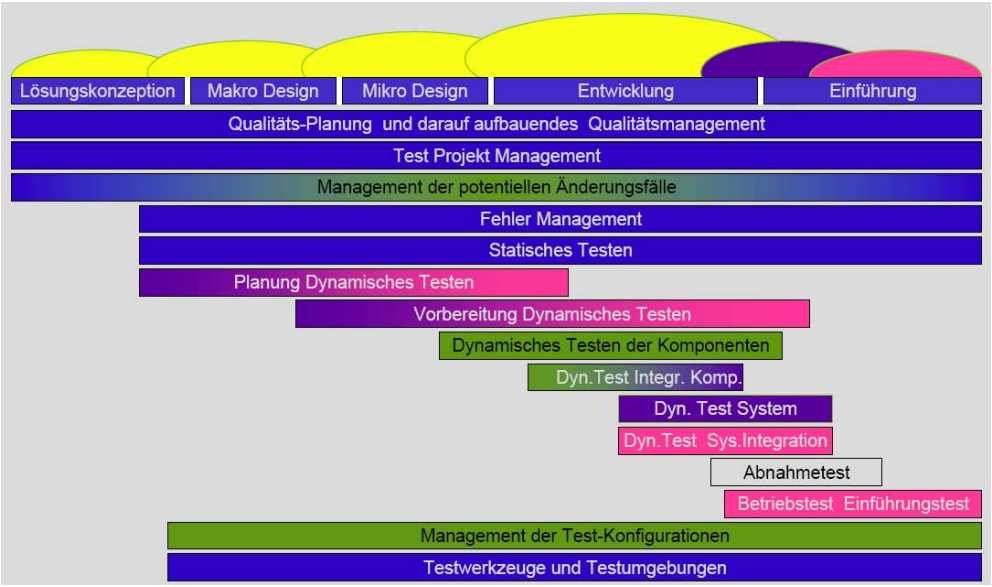


Abbildung 5: Prozessmodell „Testen"

6.1 Test Strategie (APP 141)

- Dieses Arbeitsergebnis legt die grundlegenden strategischen Regeln und Ziele für das gesamte Arbeitsgebiet Testen fest.
  - Geltungsbereich der Testregeln
  - Vereinbarung einer konsistenten Terminologie und Vorgehensweise
  - Hervorhebung kritischer Testschwerpunkte
- Das Arbeitsergebnis Test Strategie beinhaltet keine Details zu einzelnen Tests
- Folgende strategischen Regeln sind wichtig:
  - Testen ist eine Form der Qualitätssicherung (inhaltliche und organisatorische Implikationen)
  - Grundsatz des „Full Life Cycle Testing"
  - Verschiedene Stufen des Testens (vom Komponententest über Regressionstest bis zum Abnahmetest) erfordern verschiedene Formen des Testmanagements.
  - Unabhängigkeit des Testens - 4 Augen-Prinzip
  - Verbindlichkeit einer gemeinsamen Terminologie
  - Testen nur nach definierten Testzielen, Testkriterien und Testverfahren in einem einheitlichen Test-Management-Prozess
- Z.B. Klar definiertes Berichtswesen für Testergebnisse und Testfortschritt
- Z.B. Testfälle müssen vor Beginn der Programmierung erstellt sein
- Impliziert Aufbau eines Fehlermanagements
- Impliziert Aufbau eines Konfigurationsmanagements
  - Beachtung der adäquaten Modularität / Granularität
  - Beachtung der Testeffizienz

6.2 Gesamtplan Dynamisches Testen (APP 133)

Der Gesamttestplan legt in den Grundzügen fest, was, warum, wie, wann und durch wen getestet werden soll. Er bildet den Rahmen um alle projektbegleitenden Testarbeiten, enthält selbst aber keine Details. Er beschreibt unter anderem:

- **Test Ziele**: beschreibt die messbaren Ziele des Testens basierend auf den Projektzielen.
- **Test Fokus**: beschreibt die Zielrichtungen des Testens, die für das Projekt relevant sind und basiert auf den Qualitätszielen des Qualitätsplans

- **Test Gewichtung**: detaillierte Einschätzung der Folgen von Fehlentwicklungen (Nutzbarkeit und Kosten) bezogen auf die Geschäftsfunktionen und die technischen Anforderungen des Projekts.
- **Testarten**: bezieht sich auf Geschäftsfunktionen (funktionales Testen z.B. Komponententest, Integrationstest, Regresssionstest, oder Integrationstest) oder auf das Testen Nicht-Funktionaler Anforderungen wie (Benutzerfreundlichkeit, Performacne, Verfügbarkeit, Sicherheit).
- **Verantwortlichkeiten**: beschreibt die Schnittstellen zu anderen Organisationseinheiten, mit denen die Testgruppe zusammengearbeitet (z.B. wie und wann erfolgen die Code-Übergaben von der Entwicklung, wer übernimmt den Erstellungsprozess für die zu testenden Releases etc.).
- **Ein- / Ausgangskriterien**: werden für jede Testphase festgelegt.
- **Test Daten Strategie**: beschreibt die notwendige Qualität der Testdaten, wie sie erzeugt werden und wie auf sie zugegriffen werden kann.
- **Testumgebung**: beinhaltet Anforderungen an die Testumgebung /Hardware, Software, Verfügbarkeit, Durchsatz, Kapazität, Sicherheit etc.)
- **Tools**: Spezifikationen, Auswahl, Evaluation und Einsatzplanung für Testwerkzeuge.
- **Test Meilensteine**: beschreibt die wichtigsten Testziele für den gewünschten Zeitpunkt
- **Test Management Prozess**: beschreibt wie die Testaktivitäten geleitet und kontrolliert werden.
- **Test Reporting Prozess**: beschreibt in definierter Form die Testergebnisse und den Testfortschritt.
- **Fehler Management Prozess**: Beschreibt, wie gefundene Fehler analysiert, kategorisiert und behoben werden.

6.3 Statischer Testplan (APP 135)

- Der Statische Testplan beschreibt, wie Arbeitsergebnisse, die als Dokumente vorliegen, getestet werden. Dokumenten-Arbeitsergebnisse werden statisch genannt, da sie nicht - wie etwa Anwendungen - unter wechselnden Bedingungen geprüft werden. Die erarbeiteten Ergebnisse werden beim statischen Test mit erwarteten Ergebnissen verglichen. Als Vergleichsreferenz dienen, z.B. Standards, publizierte Ergebnisse anderer Projekte, oder anderweitige eigene Erfahrungen.
- Der statische Test bezieht sich auf alle Teile des geplanten Projektes, wie Dokumente zu funktionalen und nicht-funktionalen Anforderungen, die Analyse und Designmodelle, sowie Dokumente zu Projektumfang, -zielsetzung, -aufgaben, -zeitpläne, Arbeitsmittel, Modelle usw.. Er beinhaltet auch alle Projektprozesse (Änderung-, Problem- und Fehlermanagement usw.) in allen Projektphasen.

- Der Statische Testplan trägt entscheidend zur Fehlerfrüherkennung und -behebung bei und vermeidet die Vererbung von Fehlern in spätere Phasen.
- Typische Verfahren des Statischen Testplans sind: Checklisten, Interviews, vergleichende Analyse, Planspiele, Prototyping.

6.4 Abnahme Testplan (APP 131)

- Dieses Arbeitsergebnis beschreibt die Durchführung von Abnahmetests. Es enthält die zwischen Entwicklungsteam und beauftragender Fachabteilung vereinbarte Beschreibung der Abnahmekriterien, der ausgewählten Testverfahren, sowie die Beschreibung der konkreten Testdurchführung und der Bewertungsregeln für die Testergebnisse.
- Die Testkriterien und Testverfahren des Abnahmetestplans müssen eindeutig beschrieben sein, so daß der Abnahmetest auch zu eindeutigen Ergebnissen führen kann, so daß Auftragnehmer und Auftraggeber klar wissen, wann die Anforderungen erfüllt oder nicht erfüllt sind.
- Deshalb muss der Abnahmetestplan zu Beginn der Entwicklungsarbeit einvernehmlich verabschiedet sein.

6.5 Testspezifikationen (APP 140)

Dieses Arbeitsergebnis hat sicherzustellen, daß die einzelnen Tests einheitlich, hinreichend genau und konsistent spezifiziert sind. Sie sind Leitfaden zur Durchführung der konkreten Testaufgaben. Dazu gehören:

- Genaue Beschreibung der Testarchitektur, -elemente, -herangehensweisen, -daten und der erwarteten Resultate.
- Beschreibung aller Testfälle für sämtliche relevanten Test-Szenarien. Dabei ist grundlegend zu unterscheiden.
- Tests mit gültigen Ein-/Ausgabebedingungen – soll zeigen, dass das System bei richtigem Umgang das tut, was es tun soll.
- Tests mit ungültigen Ein-/Ausgabebedingungen – zeigt, dass das System robust auf Falsche Inputs reagiert, das kann bis zum „Hacking“ gehen.
- Es muss bei der Festlegung der Testspezifikationen ein klarer Bezug zwischen den Testspezifikationen und den erwarteten fachlichen und technischen Ergebnissen hergestellt werden.

6.6 Testumgebung (APP 136)

Dieses Arbeitsergebnis beschreibt die grundsätzlichen Anforderungen an eine sichere und verlässliche Umgebung für die relevanten Testarten , sowie die dafür notwendigen Prozesse, Ressourcen und Werkzeuge. Eine effiziente Testumgebung reduziert Aufwand und Kosten. Dazu gehören u.a. Aussagen zu folgenden Punkten:

- Ausstattung an Hard- und Software für die Testumgebung
- Integrationsanforderungen: Schnittstellen zu Datenbanken, anderen Anwendungen oder externen Systemen ...
- Netzwerkanforderungen für die Testumgebung
- Anzahl und Qualifikation des Personals für Aufbau, Betrieb und Wartung der Testumgebung
- Notwendige Tools
- Notwendige Methoden und Verfahren
- Sonstige Hilfsmittel
- Beschaffung, Zugriff, Speicherung der Testfalldaten und der Testergebnisdaten
- Sicherheit
- Dokumentation der Abweichungen zu Produktions- und Abnahmeumgebung
- Sonstige Abhängigkeiten

Hinweis: Für unterschiedliche Teilprojekte können unterschiedliche Anforderungen an Testumgebungen bestehen, wie auch für unterschiedliche Arten von Tests (z.B. Integrationstest, Systemtest, Abnahmetest ...). Das führt in der Regel zu mehreren physikalisch unterschiedlichen Testumgebungen.

6.7 Testergebnisse (APP 139)

Dieses Arbeitsergebnis dokumentiert die durchgeführten Tests, in einer einheitlichen und konsistenten Form, inkl. einer ordnungsgemäßen Archivierung.

- Dazu zählen auch:
  - Klare Identifizierung erfolgreicher und fehlgeschlagener Tests
  - Einheitliche Form und Struktur der Ergebnisdarstellung
  - Einheitliche Form und Struktur zur Darstellung der Ergebnisanalyse
  - Einheitliche Form und Struktur zur Darstellung der gefundenen Fehler
  - Sicherstellung, daß die Vollständigkeit der Testaufgabe erreicht worden ist
- Dokumentation von Veränderungen im Testablauf, Testdaten etc.
- Dokumentation des Fortschritts der Testeffizienz

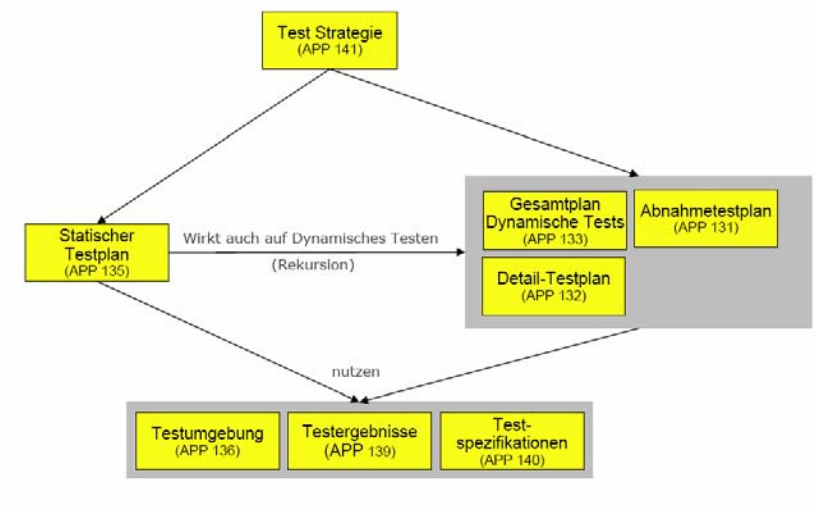


Abbildung 6: Testen: Abhängigkeiten der Arbeitsergebnisse

7 Sicherheit

Sicherheit läßt sich in 7 Teilaspekte gliedern:

1. Vertraulichkeit
2. Integrität
3. Verfügbarkeit
4. Idenfikation
5. Authentifizierung
6. Nachweisbarkeit
7. Fachliche Berechtigung

8 Muster

Definition 23 (Muster) *„Jedes Muster beschreibt ein in unserer Umwelt beständig wiederkehrendes Problem und erläutert den Kern der Lösung für dieses Problem, so dass Sie diese Lösung beliebig oft anwenden können, ohne sie jemals ein zweites Mal gleich auszuführen." [GoF96, S. 2f]*

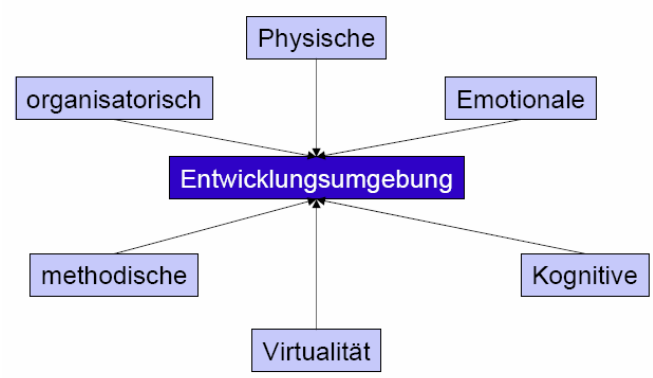
Mit dieser Definition geben Gamma et al. eine Antwort auf die Frage „Was ist ein Entwurfsmuster" mit der Übersetzung der Definition von Christopher Alexander [Alex77,S X]. Sie ist die Grundlage für die Beschreibung fünf wesensbestimmenden Elemente eines Entwurfsmusters [GoF96, S 3]:

- Mustername: beschreibt in einem Stichwort das Entwurfsproblem seine Lösung und Auswirkungen.

- Problemabschnitt: beschreibt das Problem und seinen Kontext, für das das Muster anwendbar ist
- Lösungsabschnitt: beschreibt ähnlich einer Schablone die Elemente des Entwurfes sowie die Beziehungen, Zuständigkeiten und Interaktionen
- Konsequenzabschnitt: beschreibt die Konsequenzen der Musteranwendung und die daraus resultierenden Vor- und Nachteile.
- kommunikativen Bedeutung der Muster.
- Die definierten Muster fügen sich zu einem Entwurfsvokabular (Gamma et al.) oder gar zu einer Sprache (Alexander „A Pattern Language“).
  - Er fordert die Architekten auf, sich aus den Mustern eine eigene Sprache für die eigenen Projekte zu bauen.
  - Er erläutert seinen Appell am Beispiel einer Veranda [Alex77 S. XXXV], indem er „seine” Veranda mit zehn seiner 253 Muster beschreibt:
    * private Terrasse auf der Straßenseite, sonniger Platz, Raum im Freien, sechs Fuß Balkon, Wege aus Ziegeln, variable Dachhöhe, Säulen an den Ecken, Bank vor der Tür, erhöhte Blumen, unterschiedliche Stühle.
  - Er zeigt damit anschaulich, wie mit zehn wieder verwendeten „Pinselstrichen“ ein Bild der wesentlichen Elemente „seiner“ Veranda entsteht.

9 Umgebungen

Entwicklungsumgebungen sind nicht nur Programmierumgebungen. Wir unterscheiden 6 Aspekte von Entwicklungsumgebungen.



9.1 Physische Entwicklungsumgebungen

- Gebäude
- Räume
- Maschinen
- Netzwerke

9.2 Virtuelle Entwicklungsumgebungen

- Betriebssystemumgebungen
- Netzumgebungen (WWW, IP Domain, WebSphere, LPAR)
- Anwendungssysteme
- Untersuchungsfelder in einem Labor
- Testumgebungen (Hinweis auf „Explosion“ der notwendigen Anzahl vonTestumgebungen)

9.3 Organisatorische Entwicklungsumgebungen

- Menschen
  - Individuen
  - Gruppen
- Rollen
- Aufgaben
- Regeln
- Prozesse

9.4 Methodische Entwicklungsumgebungen

- Vorgehensweisen
  - Analyse Methoden
  - Design Methoden
  - Entwicklungsmethoden
  - Testmethoden
- Granularität
  - Vorgehensmodelle (*abstrakt, grob*)
  - Leitfäden (*konkret, fein*)
  - Technique Papers (*bis in jedes Detail*)

9.5 Kognitive Entwicklungsumgebungen

- Die „Denk“-Umgebung
  - Wissensvorrat
  - Denkfähigkeit, die man mitbringt / Denkfähigkeit, die gefordert wird
  - Abstraktionsfähigkeit, die man mitbringt / Abstraktionsfähigkeit, die gefordert wird
  - Angebot / Bedarf von Wissensankern

9.6 Emotionale Entwicklungsumgebungen

- Die „Gefühls“-Umgebung
- Wahrnehmung meiner Position, formal, inhaltlich, sozial
- Wahrnehmung meiner Aufgabe, formal, inhaltlich, verantwortungsbezogen
- Selbstbild / Fremdbild
- Eigene Verhaltensmuster / die Verhaltensmuster des Team
- Eigene emotionale Gestimmtheit / wahrgenommene emotionale Gestimmtheit anderer

9.7 Reale Entwicklungsumgebungen

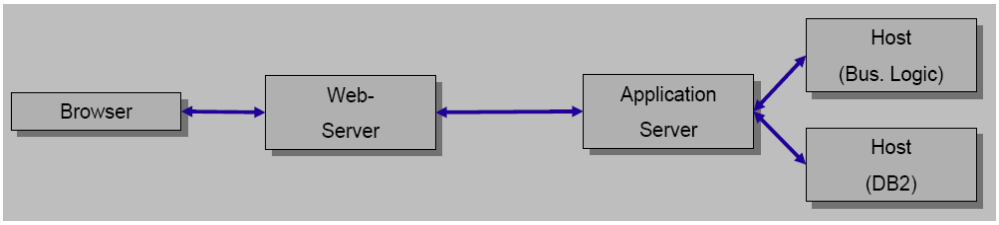
- Reale Entwicklungsumgebungen sind ein komplexer Mix aus allen Aspekten.
- ute Entwicklungsumgebungen zu schaffen ist eine schwierige Aufgabe und setzt voraus:
  - Differenzierte Wahrnehmung
  - Maßnahmenbündel mit direkter und mit indirekter Wirkung
  - Wissen um Grenzen des eigenen Handelns
- Grundlage ist gute Kommunikation
  - in Form und
  - in Inhalt

10 Architekturen

- Schichtung (Layer) $\longrightarrow$ Abstraktionslevel $\longrightarrow$ Wiederverwendung
- Stufung (Tiers) $\longrightarrow$ (loosely) Coupling (logische Trennung)

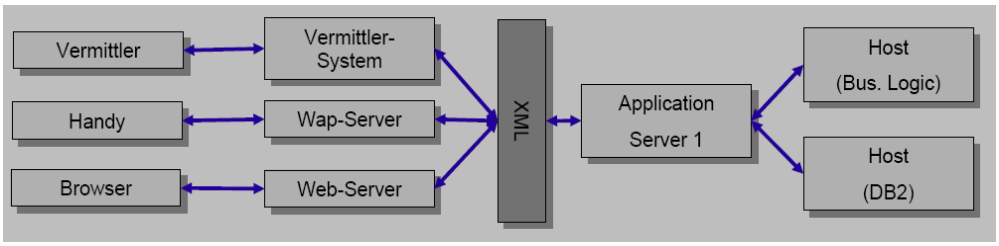
10.1 Ableitung einer Zielarchitektur Ausgangspunkt: Reine Web-Architektur

- Treibende nichtfunktionale Anforderungen:
  - mehrere und unterschiedliche Benutzergruppen, die auch Nicht-Mitarbeiter beinhalten können
  - uneingeschränkte örtliche Verfügbarkeit.
  - keine Softwareverteilung
  - keine Offline-Fähigkeit gefordert



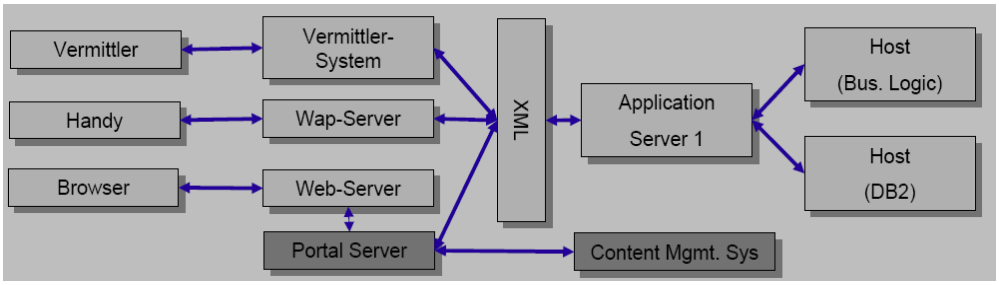
10.2 Ableitung einer Zielarchitektur: zusätzlich Multikanal-Architektur

- Treibende nichtfunktionale Anforderungen:
  - die verschiedenen Benutzergruppen benutzen unterschiedliche Endgeräte: Desktop, Browser, Mobiltelefon, Anwendungssysteme,
  - unterschiedliche Bedienung der Systeme Kommunikation und Kooperation zwischen mehreren Systemen



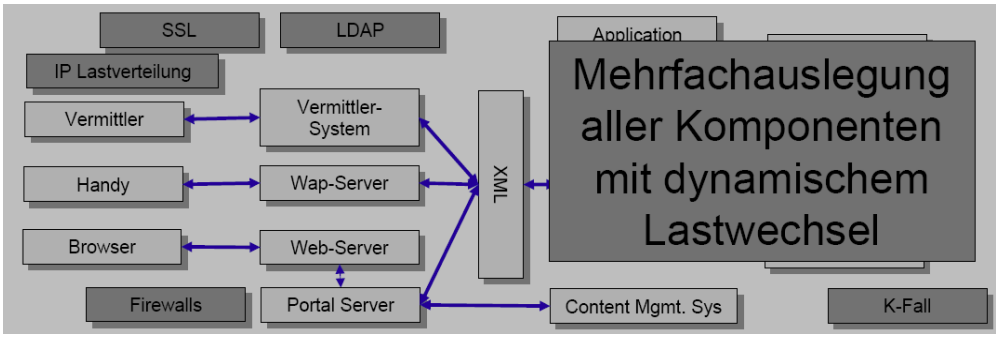
10.3 Ableitung einer Zielarchitektur: zusätzlich Portal-Architektur

- Treibende nichtfunktionale Anforderungen:
  - Oberflächen-Integration
  - einheitliches Aussehen, einheitliche Bedienungsmuster, Nutzung gemeinsamer Funktionen wie Hilfe, Chat, Videokonferenz
  - Konfigurierbarkeit der eigenen Arbeitsumgebung



10.4 Ableitung der Zielarchitektur: zusätzlich Sicherheit, Skalierbarkeit und Verfügbarkeit

- Treibende nichtfunktionale Anforderungen:
  - Sicherheit
  - Skalierbarkeit
  - Verfügbarkeit



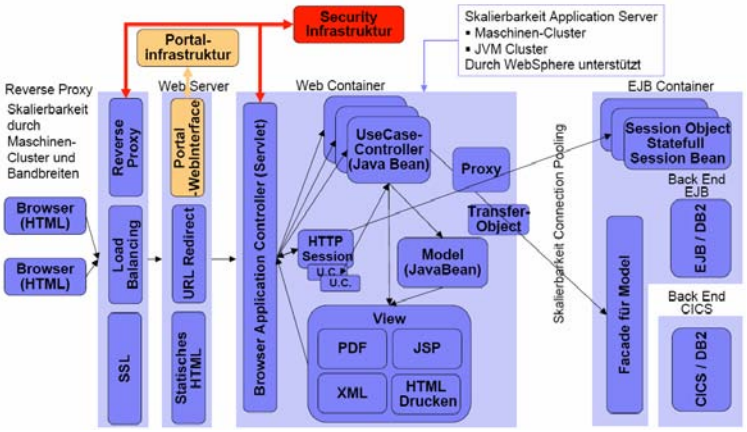


Abbildung 7: Standard Web-Architektur

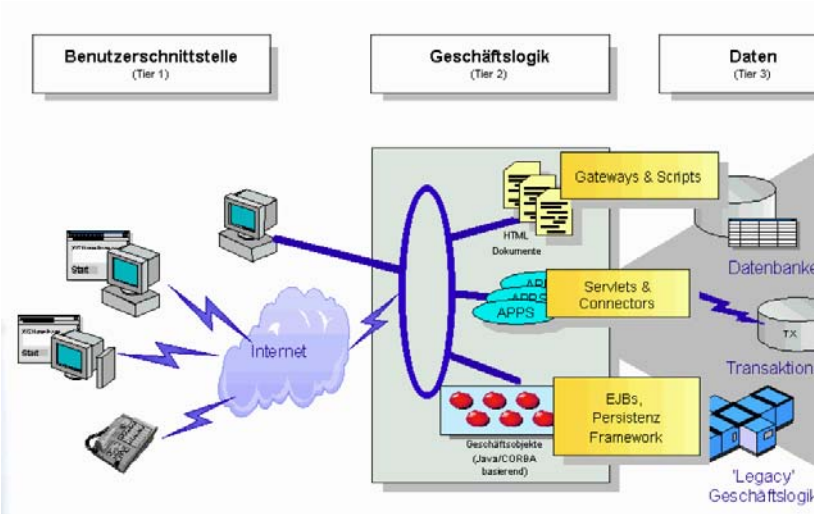


Abbildung 8: Web-(R)Evolution zur 3-tier Architektur

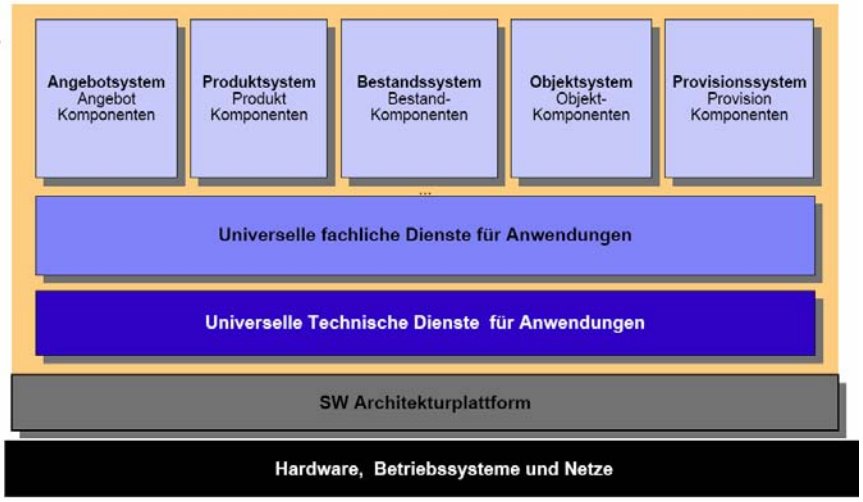


Abbildung 9: Komponentenorientierte geschichtete (Layer) Architektur