

Veranstaltung
10033013 (Diplom) und 101055,101056 (Bachelor)

Systementwicklung

Einführung und Überblick über den Systementwicklungsprozess

Uwe H. Suhl und Chris Bizer

SS 2008

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

1. Systementwicklung - Einleitung

- Systementwicklung beinhaltet die Gesamtheit der planenden, analysierenden, entwerfenden, ausführenden und prüfenden Tätigkeiten zur Schaffung eines neuen oder Änderung eines bestehenden Anwendungssystems
- Ziel der Systementwicklung ist also die Planung und Realisation von Anwendungen unter Abwägung von Nutzen und Kosten einer Lösung
- fundamental ist die Trennung in Spezifikation und Konstruktion
 - **Spezifikation:** Festlegung, was ein System leisten soll; maßgeblich: Anwender
 - **Konstruktion:** Festlegung, wie die Ziele erreicht werden (IT-Fachleute)
- softwaretechnische Aspekte, z.B. Programmierung sind zunächst sekundär; man unterscheidet das Fachkonzept und das daraus abgeleitete IT-Konzept
- Systementwicklung erfolgt im Rahmen eines Projektes, d.h. einer einmaligen zeitlich befristeten Aufgabe, die von einem Projektteam aus Anwendern und IT-Spezialisten durchgeführt wird

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Systementwicklung und Software-Entwicklung

- **Systementwicklung umfaßt Software-Entwicklung:**
 - Strukturierung aller Geschäftsprozesse
 - ganzheitliche Systembetrachtung (Informationsmanagement)
 - organisatorisches Umfeld
 - unternehmensweite Informationssysteme und Datenbasis
 - Systemanforderungen → Softwareanforderungen
- **Software-Entwicklung: Planung und Entwicklung von Softwaresystemen**
 - **Synonym: Software-Engineering**
 - Entwurf und Implementierung größerer Softwaresysteme
 - generelle Prinzipien, Methoden und Werkzeuge
 - Programmiertechniken im Kleinen

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Software - Begriffsbestimmung

- **Softwareformen**
 - **Computerprogramme: Quellen-, Objektprogramme, Lademoduln, Kommando- und Makroprozeduren, Middleware)**
 - **Benutzerdokumentation: Handbücher, On-Line-Hilfe (wird immer wichtiger)**
 - **Installationsdokumentation**
 - **Systemdokumentation: Systemarchitektur, Algorithmen, Datenstrukturen, Softwarebausteine mit Schnittstellen; wichtig für Systemänderungen durch andere Entwickler (Normalfall)**
- **Softwareklassifikation**
 - **Anwendersoftware: bindet den Computer in Anwendungen der IV ein**
 - **Systemsoftware: dient zur funktionellen Verbesserung der Hardware und erlaubt weitgehende Portabilität der Anwendersoftware**
 - **Basissoftware: z.B. RDBMS,, Kryptosoftware, Bildverarbeitung,**
 - **Middleware: verbindet unterschiedliche APIs, Softwareschichten, führt Transformationen aus, ist transparent für den Entwickler**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Softwareeigenschaften

- größere Softwaresysteme, die u.U. aus tausenden einzelner Funktionen bestehen, weisen eine große logische Komplexität auf
- Großunternehmen benutzten in der Vergangenheit tausende von Anwendungssystemen; der Einsatz von ERP-Systemen führte zu einer drastischen Reduktion von Softwaresystemen.
- Software ist auch nach langjähriger Ausreifung fehlerbehaftet (Fakt!)
- besondere Probleme werfen Design-Fehler auf, die erst nach langer Zeit wahrgenommen werden z.B. Jahr-2000 Problem
- Software muß permanent an neue Anforderungen angepaßt werden
- die Entwicklung größerer Systeme erfolgt immer in Teamarbeit: Anwender, Entwickler und Pfleger sind i.d.R. unterschiedliche Personen
- durch die hohe Dynamik im IT-Markt ist es leicht möglich auf falsche Systeme oder Entwicklungstrends zu setzen

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Qualitätsmerkmale von Software

- | | |
|--|--|
| <ul style="list-style-type: none">● Dokumente<ul style="list-style-type: none">■ Vollständigkeit■ Korrektheit■ Konsistenz■ Verständlichkeit■ Änderbarkeit | <ul style="list-style-type: none">● Software Produkt<ul style="list-style-type: none">■ Benutzerfreundlichkeit■ Korrektheit■ Zuverlässigkeit■ Effizienz■ Wartungsfreundlichkeit |
|--|--|

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Individualsoftware

- **Wird auftragsbezogen entwickelt, um spezielle Anforderungen der Anwender zu realisieren**
- **wird durch einen Systementwicklungsprozess produziert**
- **in tieferen Schichten wird SSW für Basisfunktionen eingesetzt (DBMS)**
- **insbesondere bei größeren, komplexeren Softwareprojekten besteht ein hohes Risiko, daß die Software nicht fertiggestellt wird**
- **durch singulären Einsatz extrem teuer**
- **lange Ausreifungszeit, da es länger dauert bis alle Fälle „durchgespielt“ wurden**
- **Bindet für Wartung / Pflege einen hohen Anteil der Entwicklerkapazität (in manchen Unternehmen über 60%)**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Betriebliche Standardsoftware

- **branchenneutrale und branchenspezifische Software für genau spezifizierte Funktionen u. Problemstellungen**
- **Erstellung durch spezialisierte Softwareproduzenten**
- **meistens hochgradig parametrisierbar, daher vielfältig einsetzbar**
- **Aufgabe: Auswahl der SSW und Anpassung an das Unternehmen**
- **Teile einer Systementwicklung, mindestens bis zum Pflichtenheft, müssen auch hier durchgeführt werden**
- **wird durch Softwareproduzenten im Rahmen einer Systementwicklung sehr viel professioneller entwickelt und getestet**
- **In mysap sind Funktionalitäten enthalten, die mit Individualentwicklung praktisch nicht erreicht werden können**
- **Wartung / Pflege ist auch teuer, jedoch i.d.R. viel billiger als bei Individualentwicklung und bindet keine Mitarbeiter (Outsourcing)**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Entwicklungstrends

- **Veränderungen in der Softwareentwicklung / Herausforderungen**
 - zunehmende Bedeutung der Software
 - wachsende Komplexität
 - zunehmende Qualitätsanforderungen
 - Nachfragestau und Produktionsengpässe
 - Trend zu Standard-Software
 - Zunahme von „Altlasten“ (Legacy-Systems) mit Wartungsproblemen
 - Zunahme des „Outsourcing“
- **Trotz des Trends zur Standardsoftware gibt es nur wenige Unternehmen, die nur SSW einsetzen**
- **Fast immer wird bei einer Systementwicklung auf vorgefertigte Softwarebausteine zurückgegriffen, z.B.:**
 - Datenbanksysteme
 - Optimierungssoftware
 - Archivierungssoftware
 - Scannersoftware für Belege, Bar-Codes, RFID-Chips, ...

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Probleme von Systementwicklungen

- **die Entwicklung eines umfassenden Anwendungssystems ist komplex, teuer und mit vielen Risiken behaftet**
- **Hauptfaktor sind die sehr hohen Personalkosten in Deutschland, die auch dazu führen, dass *Softwarekomponenten* in „Niedriglohnländern“ entwickelt werden**
- **Systementwicklungen kosten häufig mehr und dauern länger als geplant**
- **Produktivitätsunterschiede (Fehlerrate) von Entwicklern sind groß**
- **ca. 1/3 aller komplexen Systeme werden vor Fertigstellung abgebrochen**
- **viele Systeme funktionieren nicht wie geplant o. werden nicht eingesetzt**
- **schwierig: Ändern bestehender Software (Altlasten) für neue Anforderungen**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Probleme von Systementwicklungen

- **Kosten für Wartung / Pflege von Softwaresystemen können während der Einsatzzeit das 10-fache der Entwicklungskosten betragen**
- **Denver Intern. Airport wurde 18 Monate verspätet eröffnet**
 - Grund: fehlerhafte Software zur Gepäcktransportsteuerung
 - jeder Tag Verzögerung verursachte ca. eine Mio. US \$ Kosten
 - Gepäcktransportsystem kostete 193 Mio. US \$
- **AA wollte Sabre funktionell um Hotel- u. Mietwagenbuchung erweitern**
 - Entwicklungskosten von Sabre: zwei Mia. US \$ (1970-75)
 - neue Systemkomponente mit Anpassungen: 165 Mio. US \$
 - fehlerhafte Software verursachte teure Schadenersatzansprüche
 - AA zog das System nach vergeblichen Korrekturen 1992 zurück
- **Bell Atlantic: Einnahmeverlust durch Softwarefehler 1993 ca. 1 Mia. US \$**
- **Die verspätete Inbetriebnahme des Maut-System in Deutschland verursachte Einnahmeverluste des Staates nahezu in Milliardenhöhe**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Liste von Risikoelementen bei Systementwicklungen

1. **Personalprobleme im Entwicklerteam: Projektleitung, organisatorische Defizite, mangelnde technische Qualifikation**
2. **Personalprobleme beim Auftraggeber: Legende sind gescheiterte Projekte für die öffentliche Verwaltung: komplexe Aufgaben und nicht ausreichende Präzision der Aufgabenstellung (⇒ 3.)**
3. **Häufige Änderungen der Anforderungen bzw. des Auftragnehmers**
4. **Unrealistische Zeit- und Kostenpläne**
5. **Entwicklung der falschen Funktionalität**
6. **Unpassende Benutzerschnittstelle**
7. **Qualitätsmängel bei extern vergebenen Komponenten**
8. **Qualitätsmängel bei extern vergebenen Aufgaben**
9. **Unzureichende Produktleistungen**
10. **Projektziele an der Grenze des „State-of-the-Art“**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Terminologie

- Systementwicklung beinhaltet u.a. die Bereitstellung und systematische Verwendung von Prinzipien, Methoden, Verfahren, Konzepten, Notationen und Werkzeugen für die arbeitsteilige Entwicklung von Software-Systemen unter Berücksichtigung von Kriterien wie Kosten, Zeit, Qualität
- Begriffsbildung:
 - **Methoden**: planmäßige, begründete Vorgehensweise (Regelwerk): Beispiel: Use-Cases
 - **Methodik**: in der Art des Vorgehens festgelegte Arbeitsweise; wir benutzen **Vorgehensmodell** als Synonym. Beispiel Unified Process, Top-Down, Bottom-Up
 - **Verfahren**: Anweisungen zum gezielten Einsatz von Methoden; Beispiel: Entscheidungstabellen, Normalisierung von Relationstypen
 - **Entwurfsprinzipien**: Grundsätze nach denen ein System entwickelt wird; Beispiele: Geheimnisprinzip, Minimalität von Schnittstellen, Objektorientierung
 - **Konzepte**: Modellierung von Sachverhalten unter bestimmten Gesichtspunkten Beispiele: ER-Modellierung, Ereignisgesteuerte Prozessketten (EPK)
 - **Notationen**: Darstellung von Konzepten durch eine festgelegte Menge grafischer oder textueller Symbole, zu denen eine Syntax und eine Semantik definiert ist
 - **Werkzeuge** dienen der automatisierten Unterstützung von Methoden, Verfahren, Konzepten und Notationen
 - **Techniken** aus unterschiedlichen Bereichen; Beispiel: Netzplantechnik (Projektplanung)
 - **Erkenntnisse** (Erfahrungen) aus unterschiedlichen Bereichen

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Terminologie

- **Pflege**: Anpassung eines Systems an geänderte Bedingungen oder Weiterentwicklung aufgrund geänderter oder neuer Anforderungen
- **Wartung**: Beseitigung von Fehlern und Defekten, die nach der Inbetriebnahme eines Systems auftreten
- **Software-Entwicklung**: Aus einem geplanten Software-Produkt ein fertiges Software-Produkt mit geforderten Eigenschaften entwickeln
- **Software-Management**: Planung, Organisation, Leitung und Kontrolle einer Software-Entwicklung
- **Software-Qualitätsmanagement**: Sicherstellung der geforderten Produkt- und Prozessqualität einer Software-Entwicklung durch geeignete konstruktive und analytische Maßnahmen
- **Artefakt**: Arbeitsergebnis, das durch Mitarbeiter erzeugt, geändert und benutzt wird; z.B. Pflichtenheft, OOA-Modell, C#-Programm
- **Artefakt-Muster**: Struktur, Inhalt und Layout eines Artefakts, Entwurfs-Muster

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Sichtweisen und Beschreibung von Anwendungssystemen

- **Funktionsmodell:** Systemaufgaben (Anforderungen) werden funktionsorientiert im *Lastenheft* spezifiziert
- **Benutzeroberfläche:** zeigt das System aus Anwendersicht (Systemeingaben / Systemausgaben)
- **Datenmodell:** konzeptionelle Beschreibung aller Informationsobjekte der Anwendung; basiert auf einem Modellierungsprozess
- **Organisationskonzept:** Beschreibung der Abläufe und Funktionen im Umfeld des Unternehmens, seiner Aufbauorganisation (Stellen) und der verwendeten Softwaresysteme: *Wer macht was womit?*

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Sichtweisen und Beschreibung von Anwendungssystemen

- **Systemarchitektur:** im Softwareentwurf wird das Funktionsmodell verfeinert und u.a. in Klassen- bzw. Modulspezifikationen des Anwendungssystems überführt; Klassen / Module werden in der Implementierung *programmiert*
- **Verteilungsarchitektur:** verteilte Anwendungssysteme erfordern eine Verteilung von Daten und Funktionen; relevante Aspekte sind die Art der Verteilung, technische Schnittstellen (Hardware), Protokolle, Software
- **Systemkonfiguration:** Systemplattform mit weiterer Basissoftware

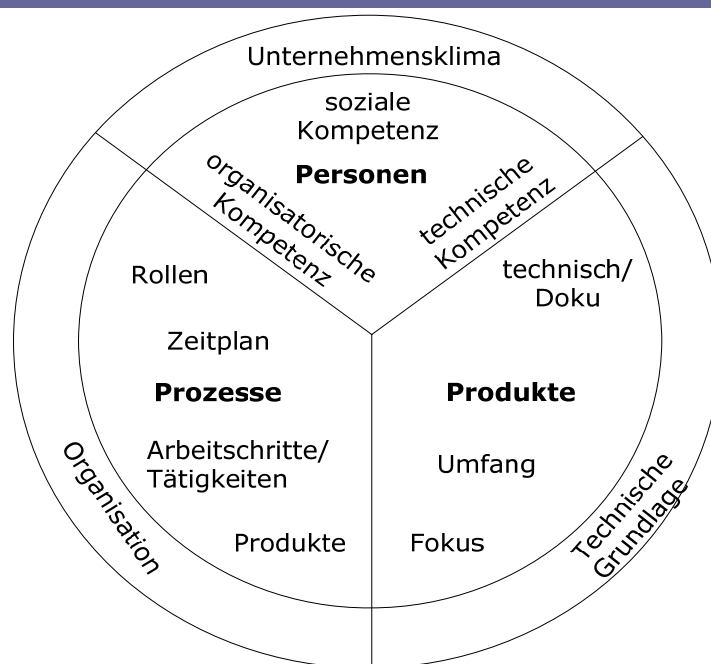
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Systementwicklung im Rahmen eines Projektes

- Einmaligkeit für das Unternehmen
- zeitlich begrenzte Aktivitäten (Mindestdauer), Zeitplan
- interdisziplinäres Team aus Anwendern und IT-Spezialisten
- Kosten, Nutzen, Konkurrenz um Betriebsmittel (Personal, Sachmittel)
- systematische Projektabwicklung:
 - anzuwendende Prinzipien, Methoden, Techniken, Richtlinien,....
 - Aktivitäten, zu erzielende Ergebnisse
 - unterstützende Werkzeuge, Hilfsmittel, Meilensteine, Qualitätssicherung
- Projektteam
 - ProjektleiterIn: Schlüsselrolle, verantwortlich für die fachliche Durchführung des Projektes unter Einhaltung von Kosten, Terminen und Fachvorgaben
 - Anwender: notwendig für das Fach- u. Organisationskonzept
 - IT-Spezialisten: Systemanalytiker, Programmierer
 - ProjektmanagerIn (Projektkoordinator): Mitglied der Geschäftsleitung, Bindeglied zum Auftraggeber, wichtig bei Zielkonflikten

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Faktoren eines Systementwicklungs-Projektes



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Vorgehensmodelle zur Systementwicklung

- **Definieren systematische Vorgehensweise zur Durchführung von Systementwicklungen.**
- **Zweck von Vorgehensmodellen**
 - **Effiziente Gestaltung von Systementwicklungen**
 - **Gemeinsame Sprache für Projekte**
 - **weniger Nacharbeit**
 - **Vereinfachung von Routineabläufen**
 - **Basis kontinuierlicher Verbesserung**
 - **Wettbewerbsvorteil Zertifizierung**
 - **Hilfe für wenig erfahrene Projektmanager**
 - **rascher Nutzen (z.B. Einschulung neuer Mitarbeiter)**
 - **Rücksicht auf typische Sonderfälle der Entwicklerrealität**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

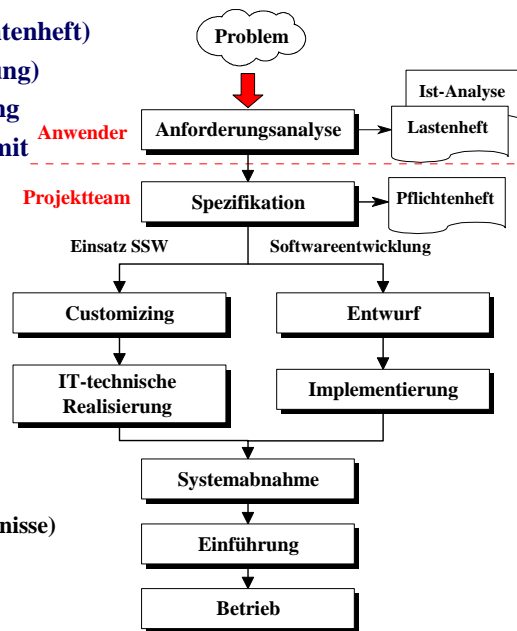
Vorgehensmodelle zur Systementwicklung

- **Klassische Vorgehensmodelle**
 - **Entstehung seit 60er Jahren**
 - **relativ statische Vorgehensweise**
 - **gut geeignet für große Projekte**
 - **Beispiele**
 - ✓ **Wasserfall-Modell**
 - ✓ **V-Modell**
 - ✓ **Spiralmodelle**
 - ✓ **Unified Process**
- **Agile Vorgehensmodelle**
 - **Entstehung seit Ende der 90er Jahre**
 - **flexibel, chaotisch**
 - **geben nur generelle Leitlinien und Prinzipien vor**
 - **geeignet für kleine Projekte**
 - **Beispiele**
 - ✓ **Extreme Programming**
 - ✓ **Crystal**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

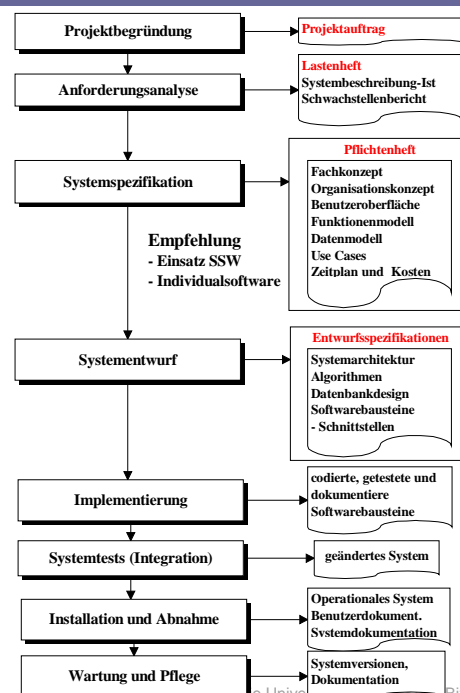
Grobes Phasenmodell zur Systementwicklung (Wasserfallmodell)

- Erstellung eines Lastenheftes (grobes Pflichtenheft)
- Erstellung eines Pflichtenheftes (Verfeinerung)
- Entscheidung: SSW oder Systementwicklung
- Entwurf liefert eine Software-Architektur mit Klassen- bzw. Modulspezifikationen
- Implementierung realisiert die Klassen bzw. Module in Form von Computerprogrammen (Softwarebausteine), die einzeln getestet werden
- Nach Integrationstests erfolgt die Systemabnahme durch den Auftraggeber
- Nach der Einführung erfolgt der Betrieb des Systems mit laufenden Änderungen
- Parallele Aktivitäten
 - Projektmanagement (Zeiten, Kosten, Ergebnisse)
 - Qualitätsmanagement
 - Aufwandsschätzungen



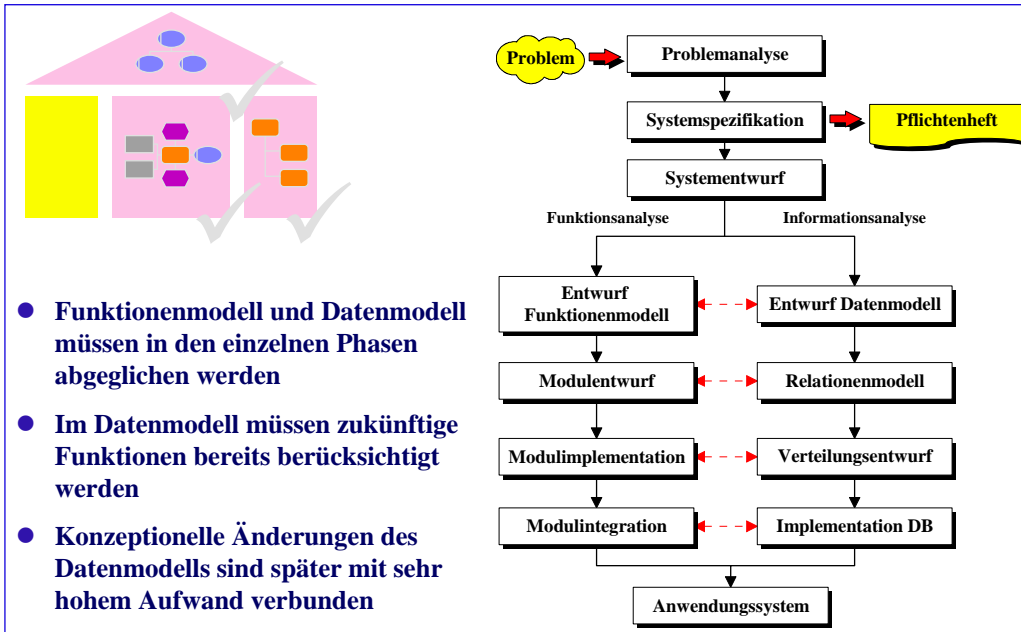
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Ergebnisse (Artefakte) der einzelnen Phasen



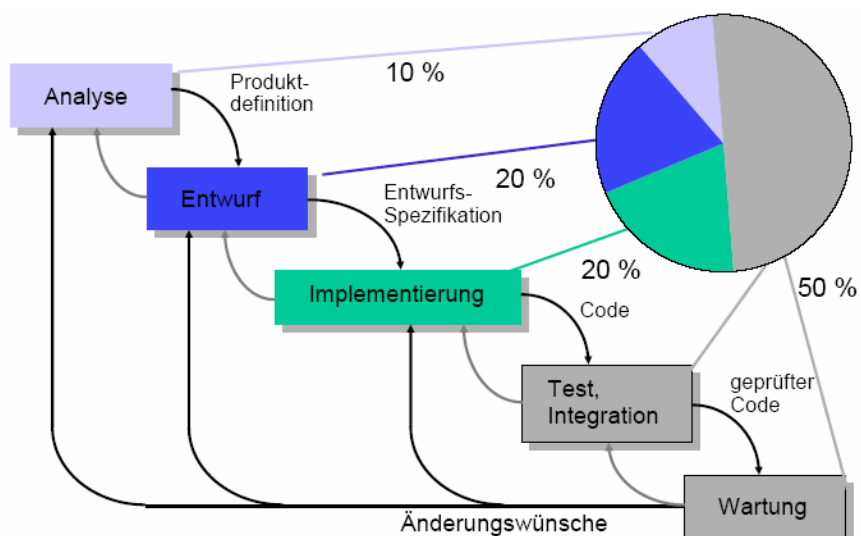
Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Datenbankentwurf in einer Systementwicklung



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Ungefähre Verteilung des Arbeitsaufwands



W. Royce (1970)

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

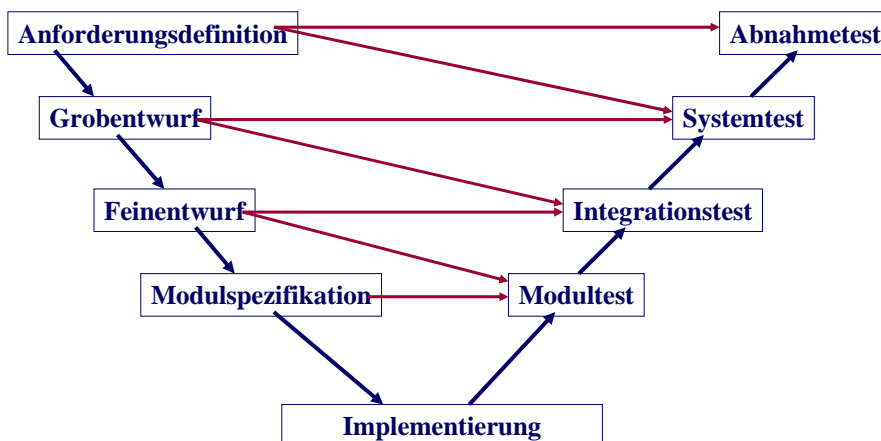
Merkmale von Wasserfallmodellen

- **Einteilung in klar abgegrenzte Phasen**
- **Ziele der Phaseneinteilung sind**
 - ✓ Strukturierung des Entwicklungsprozesses
 - ✓ phasenweise Ergebnisplanung u. Ergebniskontrolle
 - ✓ Komplexitätsreduktion durch Teilphasen
 - ✓ Einsatz spezifischer Methoden u. Werkzeuge
- **Schwächen von Phasenmodellen**
 - rein sequentiell, gehen von festen Anforderungen aus
 - Kritisch ist die Qualität des Lasten- bzw. Pflichtenheftes
 - Abgrenzung der Phasen ist manchmal schwierig, speziell bei technisch anspruchsvollen Projekten; Beispiel Gas-Einkaufsplanung
 - Fehler werden häufig nicht in der nächsten Phase erkannt
 - Sie erfordern in späteren Phasen erheblichen Aufwand und erfordern ein Re-Design bzw. Rücksprünge in vorherige Phasen
 - Für Modifikationen im Betrieb (Wartung) ist eine Rückkopplung nötig
 - Werden Fehler erst bei der Abnahme erkannt sind Änderungen kostspielig
 - Varianten: Wasserfall-Modell mit Rückkopplung, V-Modell

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

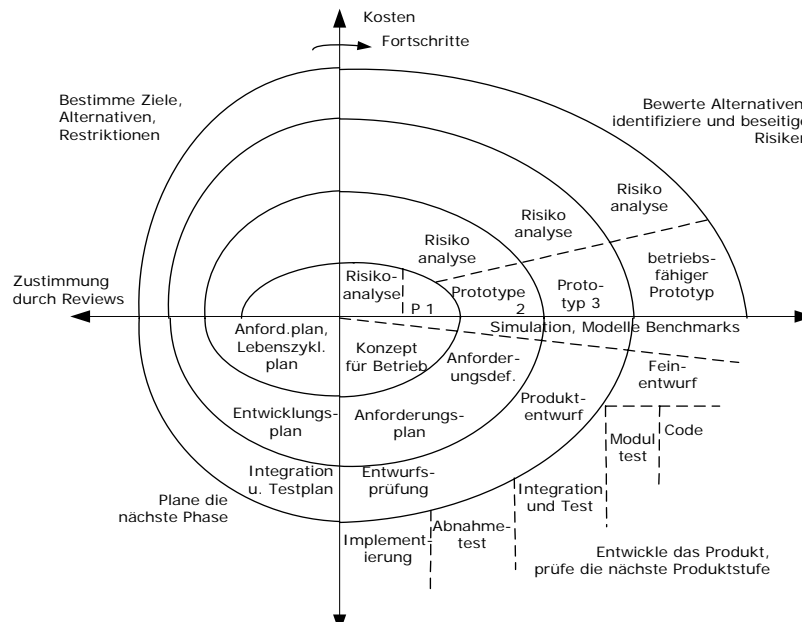
V-Modell

- **Variante des Wasserfallsmodells mit Betonung der Qualitätssicherung**
- **Zu jeder Phase existiert ein zugehöriger Test, der die Artefakte jeder Phase testet, bevor sequentiell fortgefahren wird**
- **Qualitätssicherung: Verifikation und Validation**



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Spiralmodell



Der Unified Process

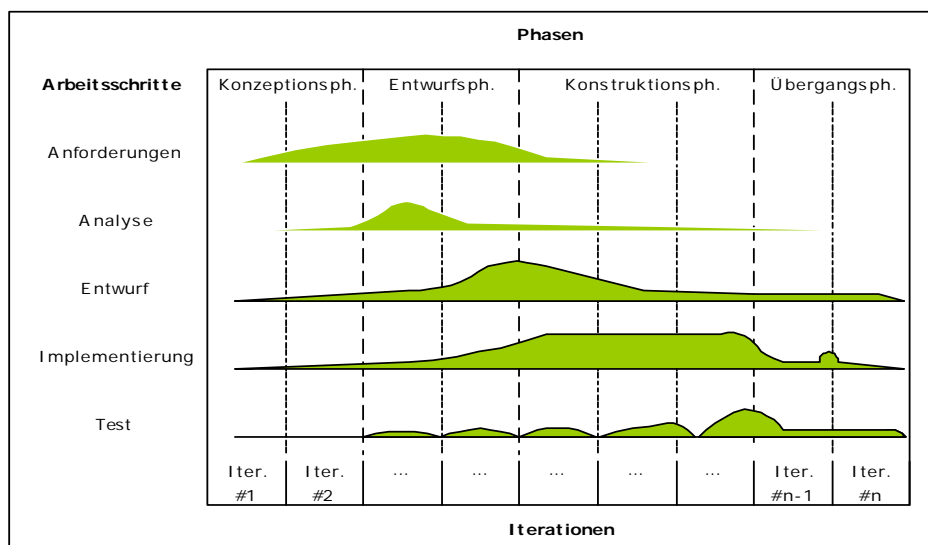
- **Der Unified Process ist ein neueres Vorgehensmodell das auf objektorientierter Softwareentwicklung ausgerichtet ist.**
- **Iterativ und inkrementell**
 - Iterative und inkrementelle Denkmuster statt sequentielle Denkmuster
- **Use-Case-Driven**
 - Use-Case = Interaktionen zwischen Akteure und System zur Bearbeitung einer bestimmten, abgegrenzten Aufgabe
 - Unterstützen die intuitive Anforderungsgenerierung
- **Visuelle Software-Modellierung**
 - Verwendung der Unified Modelling Language (UML) zur Visualisierung von Artefakten
- **Konfiguration des Prozesses**
 - Anpassung des Prozesses an die jeweiligen Anforderungen eines Projekts

Elemente des Unified Process

- **Rollen**
 - Eine Rolle definiert das Verhalten und die Verantwortlichkeiten eines Individuums oder einer Gruppe.
 - Beispiele: Projektmanager, Architekt, Programmierer, Tester
 - Dieses Verhalten wird in Aktivitäten beschrieben und jeder Rolle eine gewisse Menge von Aktivitäten zugeordnet.
- **Artefakte**
 - Produkte bzw. Informationseinheiten, die während des Projektes entstehen.
 - Sind sowohl Ein- als auch Ausgabe von Aktivitäten.
 - Artefakte entwickeln sich im Laufe des Prozesses weiter und sind Gegenstand der Versionskontrolle.
 - Beispiele: Pflichtenheft, Klassendiagramm, Programmcode
- **Aktivitäten**
 - Definieren die Arbeiten, die zu verrichten sind.
 - Aktivitäten sind beendet, wenn ein neuer Artefakt entstanden ist.
- **Vorgehen**
 - gibt die Reihenfolge an, in der bestimmte Aktivitäten aufeinander folgen

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

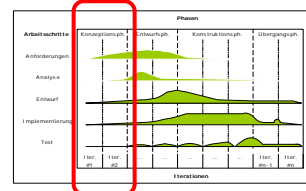
Phasen im Unified Process



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Konzeptionsphase

- Fokus des System festlegen
- Projektinfrastruktur aufbauen
- Projektstandards erstellen
- Anforderungen
 - finden
 - strukturieren
 - priorisieren
- Architekturgrundlagen bestimmen
- Architektur-Entwurf beginnen
- Kundennutzen argumentieren und präsentieren
- Kosten und Risiken vorhersagen



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

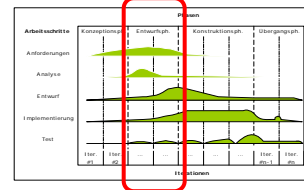
Ergebnisse der Konzeptionsphase

Artefakt	Status	Artefakt	Status
Geschäftsmodell	fast fertig	Architekturbeschreibung	fertig
Anwendungsfalldiagramm	fortgeschritten	Risiken- und Maßnahmenliste	aktuell
Anwendungsfallbeschreibung	begonnen		

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Entwurfsphase

- **Detaillierung aller Anforderungen**
- **Analysemodell**
- **Analyse-Prototypen**
- **Review der Anforderungen, des Analysemodells und des Prototypen durch den Auftraggeber**
- **Aufwandsschätzung für Realisierung**
- **Projektplanung (Wer macht wann was?)**
- **Planung und Entwurf der Anwenderdokumentation**
- **Testplanung**



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Ergebnisse der Entwurfsphase

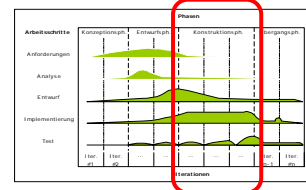
Artefakt	Status	Artefakt	Status
Geschäftsmodell	fertig	Benutzerdokumentation	begonnen
Anwendungsfalldiagramm	fertig	Entwurf der Klassen	begonnen
Anwendungsfallbeschreibung	fast fertig	Analyseprototyp	fertig
Analysemodelldiagramm	fertig	Beschreibung der Komponenten des Analysemodelldiagr.	fast fertig
Architekturbeschreibung	fertig	Testplan	fertig
Implementierung der Architektur	begonnen	Testfälle	fortgeschritten
Projektplan für Realisierung	fertig	Risiken- und Maßnahmenliste	aktuell

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Konstruktionsphase

- Implementierung der Systemteile
- Test von Systemteilen
- Qualitätssicherungs-Maßnahmen
- Integration der Systemteile
- Test des integrierten Systems
- Anwenderdokumentation, Schulungsvorbereitung
- Vorbereitung Zielplattform, Datenmigration
- Vorbereitung Wartungsinfrastruktur

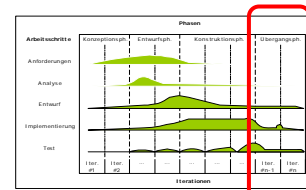
- Ergebnis der Konstruktionsphase:
Auslieferungsbereites System inklusive Dokumentation



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Übergangsphase

- Installationspakete werden fertiggestellt
- Zielplattform wird aufgesetzt
- Datenübername
- Auslieferung und Integration
- Abnahme
- Schulung von Anwendern und Systemadministratoren
- Inbetriebnahme Wartungsinfrastruktur
- Kommunikations-Kanäle für Support
- Design und Entwurfsdokumente werden archiviert
- Analyse des Projektverlaufs, Lessons-Learned



Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Extreme Programming (XP)

- **XP ist ein Vertreter der so genannten agilen Prozesse.**
- **XP wurde 1999 von Kent Beck entwickelt.**
- **XP gibt generelle Prinzipien und nur ein Mindestmaß an Regeln vor, die laufend je nach Projektbedingungen adaptiert werden.**
- **Starker Fokus auf Design und Implementierung.**
- **Eher für kleinere Projekte geeignet.**
- **Vorgehensmodell vieler Open-Source Projekte**
- **Kernansprüche:**
 - **Kunde ist Teil des Projektteams**
 - **Konsequente Orientierung des Projektes an Kundenanforderungen**
 - **Kurze Release-Zeiten**
 - **Iteratives Vorgehen**
 - **Integriertes Testen**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

Extreme Programming (XP)

- **XP kann in vier Bereiche eingeteilt werden:**
- **Planen**
 - **Ausarbeitung der Anforderungen durch Team und Kunde in Form von Stories**
 - **Einteilen des Projektes in Iterationen**
- **Kodieren**
 - **Code wird nach vereinbarten Standards entwickelt**
 - **Der Code wird immer in Paar-Programmierung erzeugt**
- **Modellieren**
 - **Einfache Strukturen mit einheitlicher Systemmetapher**
 - **Refactoring soviel und sooft wie möglich**
- **Testen**
 - **Jedes Codefragment wird getestet**
 - **Testfälle werden vor dem Programmcode geschrieben**

Freie Universität Berlin – Suhl, Bizer: Systementwicklung – SS /08 (Version vom 25.03.08)

XP Prozess

