

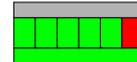
Vorbereitung

- ??
- Links auf GDPA, VM, RUP-Browser
- Browser bereitstellen
- VM-Übersichtsbild aus GDPA-Browser
- Infos aus VM-Growser ziehen

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



1
11

Vorlesung
Software Architektur-Modelle
Prozesse 1: Grundlagen, klassische Vorgehensmodelle

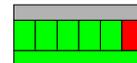
Dr. Harald Störrle

Ludwig-Maximilians-Universität München
Wintersemester 2001

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



2
11

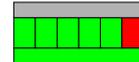
Organisatorisches

- Schwache Beteiligung beim letzten Mal
- Fragebögen soll ich auswerten und an den Dekan weiterleiten
- Schein / Anwesenheitsliste

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



3
11

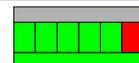
Gliederung für heute und nächstes Mal

- **Grundbegriffe**
 - Organisationsformen, Prozeßmodelle, am Bsp Änderungsverwaltung
- **Industrielle & Architektur-Prozesse**
 - VM97, RUP, ISO12207, ISO 15504 (SPICE); ATAM, SAAM, Siemens-PLP
- **Prozeßverbesserung**
 - CMM, SPICE (ISO15504)
- **Moderne Ansätze**
 - XP, Scrum, Catalysis, Crystal *, HS-Muster, ...

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



4
11

Terminologie

Software-Prozesse

Die Geschäftsprozesse zur Herstellung, Wartung, Beschaffung und zum Betrieb von Software.

Vorgehensmodell

Aufeinander abgestimmte Menge von Prozessen, Standards, Mustern, Methodiken, Werkzeugen, ...

Man sagt auch zum gesamten Gebiet „Sw.-Prozess“

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



5
11

Terminologie: Grundbegriffe, Definitionen

22.1.2002
Prozeß 1: Klassisch

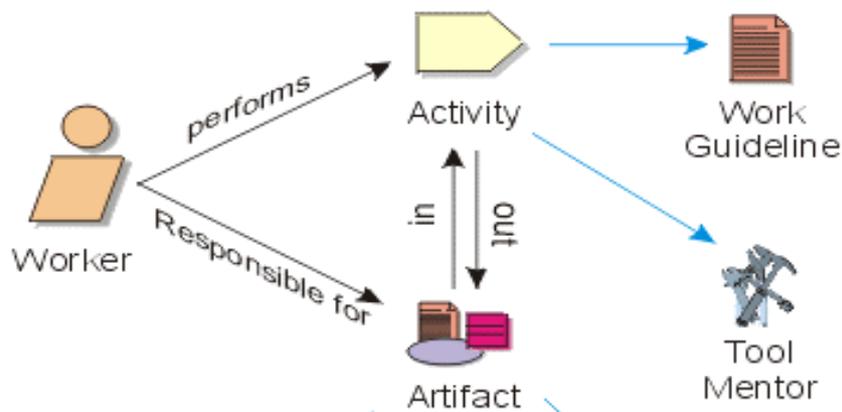
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



6
11

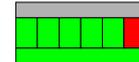
Grundbegriffe 1: Rolle, Aktivität, Artefakt, Standard, Muster



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



7
11

Grundbegriffe 2: Gliederung von Vorgehensmodellen

- Zusammengehörige Aktivitäten (usw.) werden zu Teilprozessen, und diese zu Prozessen zusammengefaßt.
- Terminologie weicht teilweise voneinander ab.

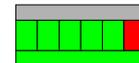


Beispiel: ?? Im VM '97

22.1.2002
Prozeß 1: Klassisch

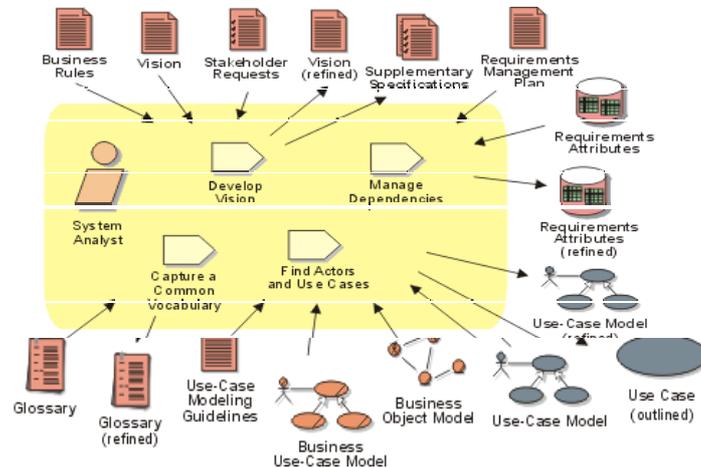
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



8
11

Grundbegriffe 2: Prozesse und Aktivitäten sind isomorph

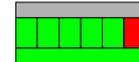


Beispiel: Anforderungsanalyse in RUP

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



9
11

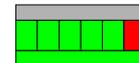
Grundbegriffe 3: Gliederung von Vorgehensmodellen

- Oft gibt es eine zusätzliche thematische Gliederung (Prozessklassen, Submodelle, ...)
- Im V-Modell gibt es die 4 „Submodelle“
- In ISO 15504 gibt es „Process Categories“
- Im CMM v1.1 gibt es „Key Process Areas“
- In etwa vergleichbar mit den Komponenten einer Facharchitektur

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



10
11

Verlgeich Sw.-Prozess / Geschäftsprozess

- Im Vergleich zu einem „normalen“ Anwendungssystem mit Geschäftsprozessen sind die Software-Prozesse in einem Vorgehensmodell
 - wesentlich zahlreicher und seltener
 - weit weniger detailliert
 - viel weniger automatisierbar
- Dadurch ist der Anteil kreativer Arbeit sehr viel höher, und damit die Anforderungen an die Mitarbeiter.

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



11
11

Verlgeich Sw.-Prozess / Geschäftsprozess

- Genau wie bei herkömmlichen Geschäftsprozessen werden auch Sw.-Prozesse hierarchisch gegliedert, nur daß die untersten Elemente Aktivitäten heißen, und nicht Nutzfal – vom Benutzer aus gesehen sind Nutzfälle aber genau Aktivitäten.
- Wenn ein Sw-Prozesse eine Art Geschäftsprozess ist, dann entspricht die Zerlegung in Submodelle einer Fachachitektur.

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



12
11

Motivation: Warum befassen wir uns mit Software-Prozessen?

22.1.2002
Prozeß 1: Klassisch

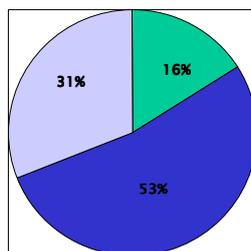
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



Software-Projekte scheitern sehr oft: Zahlen des Grauens

250.000 Projekte wurden untersucht (1994)



- erfolgreich
- massive Probleme
- abgebrochen

„erfolgreich“

Im Zeit- und Kostenrahmen, mit der erwarteten Funktionalität und Qualität

„Probleme“

Überschreitung von Zeit- und/oder Kostenrahmen, Nicht-Erfüllung von Funktionalität und/oder Qualität

- durchschnittliche Kostenüberschreitung um 800%
- durchschnittliche Zeitüberschreitung um 222%
- fast 50% aller Projekte haben mehr als 100% Zeitüberschreitung.

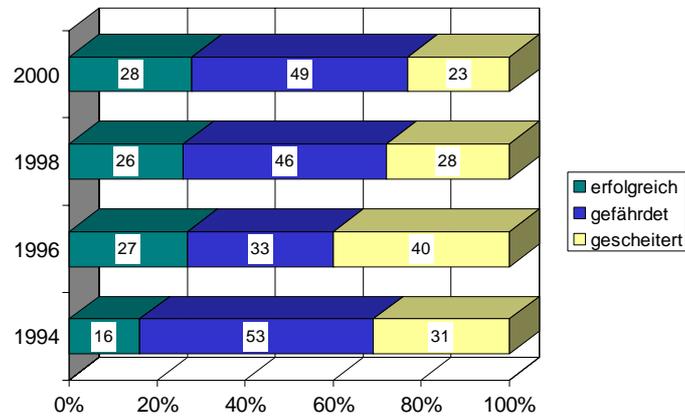
22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



Gründe für IT-Outsourcing in der Versicherungswirtschaft



Quelle: SYSTEM-Journal 04/2001

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



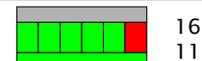
Software-Projekte scheitern sehr oft: Aktuelle Schreckensmeldungen – 1

- **BASIS3000**
 - System zur Abwicklung der Sozialhilfezahlungen des Landes Berlin: In Berlin werden pro Jahr ca. 3,4 Mrd. DM an Sozialhilfe an 27.000 Empfänger ausgezahlt, verwaltet von rd. 3000 Mitarbeitern an 46 Standorten
 - Ziel war eine verteilte 3-Schichten C/S-Lösung, Applet-Front-End, veranschlagt waren 12 UNIX-Server
 - Auftragsvolumen: 15 Mio. DM (Festpreis), Start 25.1.1999 (Konsortium von PSI & Oracle)
 - Konsortium hat im November 2000 abgebrochen, Konsortium muß die bereits erhaltenen 5 Mio. DM zurückzahlen.
- **Kostet PSI und Oracle 5 Mio. DM + Rufschäden**

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



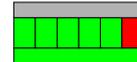
Software-Projekte scheitern sehr oft: Aktuelle Schreckensmeldungen – 2

- InPol–Neu
 - Das neue Polizeiliche Informationssystem.
 - Sollte Software aus den 70er Jahren ablösen (Stichwort Schleierfahndung)
 - Bis 10/2001 waren bereits ca. 100 Mio. DM investiert.
 - Bis 2005 sind weitere 195 Mio DM geplant.
 - Antwortzeiten zwischen 7 und 27 Sekunden (Ziel: 3)
 - KPMG–Bericht: „Das Projekt [...] befindet sich derzeit in einem erheblich sanierungsbedürftigen Zustand. Es ist nicht auszuschließen, daß das Projekt mit dem heutigen Entwicklungsansatz nicht erfolgreich abgeschlossen werden kann“
- Projekt scheitert, 300 Mio. DM Steuergelder in den Sand

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur–Modelle
Dr. Harald Störrle

LMU



17
11

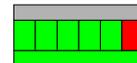
Software-Projekte scheitern sehr oft: Aktuelle Schreckensmeldungen – 3

- Fiscus
 - 1991 beschlossen die Finanzminister der Länder, in den rund 650 Finanzämtern bundesweit ein neues System zur Einkommenssteuerberechnung einzuführen.
 - Bislang: altes Großrechnersystem (Cobol/Assembler), 250 Programmierer sind mit der Wartung beschäftigt (= ca. 25 Mio DM Wartungskosten p.a.)
 - Das neue System sollte San–Francisco–basiert sein, Beginn der Einführung sollte 1997, sein Volleinsatz ab 2003
 - Geplant war ein Investitionsvolumen von 330 Mio DM. Der Bayerische Rechnungshof rechnet jetzt mit 1,4 Mrd. DM
- über 1 Mrd. DM Mehrkosten – Steuergelder

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur–Modelle
Dr. Harald Störrle

LMU



18
11

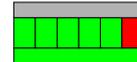
Software-Projekte scheitern sehr oft: Macht das was?

- Ja: das ist eine volkswirtschaftliche Katastrophe
- Das BM für Bildung und Forschung schätzt den „Markt für Software und verwandte Dienstleistungen“ in Deutschland“ 1999 auf 50 Mrd. DM, und geht von 10% Steigerung p.a. aus.
- Das macht 2002 etwa 34 Mio. Euro.
- Bei einer durchschnittlichen Kostenüberschreitung von 800% sind davon knapp 90% Fehlinvestitionen!
- Oder aber, fast alle Schätzungen sind gnadenlos zu niedrig...

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



19
11

Software-Projekte scheitern sehr oft: Woran liegt es?

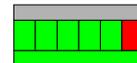
Softwareprojekte haben eine Reihe spezifischer Schwierigkeiten

- Sehr viele Systeme sind Unikate
- Es gibt sehr viele Lösungen, mit vielen Freiheitsgraden
- Software-Entwickler sind Individualisten
- Sehr rascher technologischer Fortschritt
- Abstrakte Domäne erschwert die Arbeit
- Systeme sind oft geprägt von der Anwendungsdomäne
- Die Anwendungsdomäne verändert sich rasch
- Die Informatik ist noch nicht ausreichend professionalisiert

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



20
11

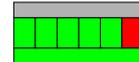
Software-Projekte scheitern sehr oft: Was kann man also tun?

- **Herstellungsprozess verbessern**
 - Abläufe
 - Organisation
 - Methoden, Standards, Werkzeuge
- **Informatik als Disziplin**
 - Massive Investition in Aus- und Fortbildung
 - Austausch zwischen Industrie und Hochschulen
 - Förderung berufsständischer Organisationen

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



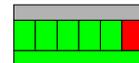
21
11

Beispiel: Der Software-Prozess „Änderungsverwaltung“

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



22
11

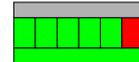
Beispiel: Änderungsverwaltung Szenario

- Ein System ist im Produktivbetrieb bei verschiedenen Kunden.
- Es kommen laufend neue Anforderungen (Erweiterungen, Korrekturen, Änderungen) von den Kunden aber auch von der eigenen Produkt-Gruppe.
- Die Anforderungen haben sehr unterschiedliche Qualität und Dringlichkeit, widersprechen sich teilweise oder sind unsinnig oder unrealistisch (Preis, Aufwand).

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



23
11

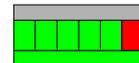
Beispiel: Änderungsverwaltung Szenario

- Aktuell werden Emails, Telefonate, Faxe, Disketten, und Fehlerdatenbanken völlig unkontrolliert, unprotokolliert und unkorreliert ausgetauscht.
- Änderungsanforderungen gehen verloren, bleiben liegen, können nicht abgerechnet werden, werden mehrfach abgerechnet, werden durchgeführt und wieder rückgängig gemacht...
- Es hängt zuviel an einzelnen Personen und an Glück. Die Anforderungen müssen also verwaltet werden, aber wie?

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



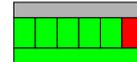
24
11

Beispiel: Änderungsverwaltung Grundüberlegungen

- Es muß ein Gremium geben, daß darüber entscheidet, welche Anforderungen in welcher Priorität umgesetzt werden (Change Management Board).
- Die eingehenden Meldungen müssen für die Entscheidungsfindung dieses Gremiums vorverdichtet werden.
- Die Verwaltung der Meldungen *muß* automatisch erfolgen, z.B. um...
 - ...der Masse überhaupt Herr zu werden
 - ...die Erfüllung von Service-Level-Agreements dokumentieren zu können
 - ... Änderungen mit dem Versionsverwaltungssystem koordinieren zu können
 - ...

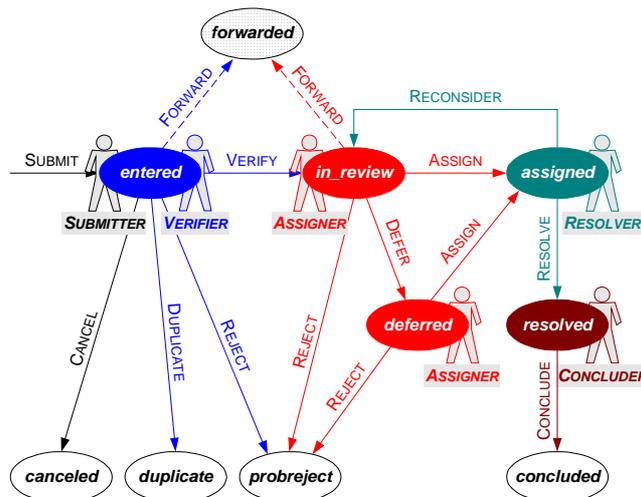
22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



25
11

Beispiel: Änderungsverwaltung Der Prozess



Notation

Rollen

Zustände

Kanten



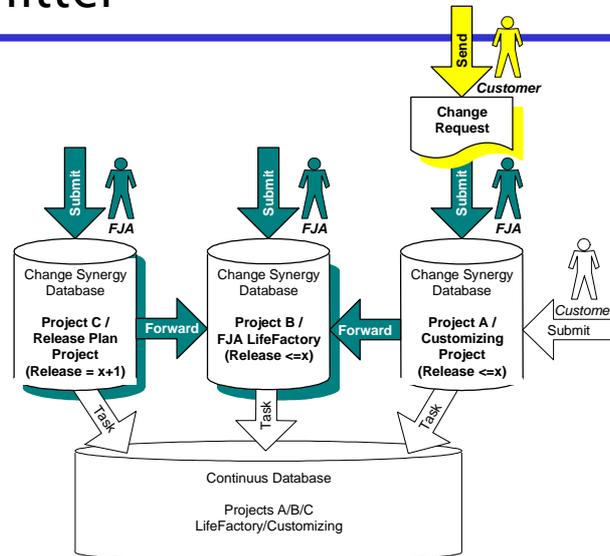
22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



26
11

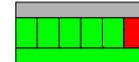
Beispiel: Änderungsverwaltung Die „Submitter“



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



27
11

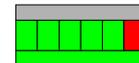
Beispiel: Änderungsverwaltung Kritische Fragen

- **Einspeisung**
 - Darf/Muß der Kunde Fehler direkt einspeisen (Menge/Qualität der Meldung)?
 - Per Zuruf (Telefon, Email), Ticket-System, übers Web im System?
 - Kosten, Latenzzeit, Kundenzufriedenheit
- Nach welchen Regeln soll die Vorverdichtung erfolgen?
- **Change-Management-Board (CMB)**
 - Wer gehört alles dazu?
 - Wie oft tagt das Board?
 - Nach welchen Kriterien wird entschieden? (Releaseplan...)
- **Wie werden Stellvertreter integriert?**

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



28
11

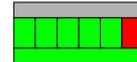
Beispiel: Änderungsverwaltung Die Implementierung des Prozesses

- Aktuelle Prozesse analysieren
 - Mengengerüst!
 - Ansprechpartner!
 - Probleme!
- Verbesserungen erarbeiten
- Projektleitung überzeugen
- Rollen belegen
- Mitarbeiter schulen
- DB aufsetzen, System installieren
- Kick-Off
- Nachverfolgung

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



29
11

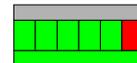
Beispiel: Änderungsverwaltung Probleme bei der Implementierung

- Leute haben Angst um ihre Position:
 - Bin ich danach noch unersetzbar? (Arbeitsplatz...)
 - Fällt dann auf, wie schlecht ich arbeite? (Gehalt...)
 - Muß ich Bereitschaftsdienst übernehmen? (Familie...)
- Widerspricht ggf. der Unternehmenskultur
 - Haben wir noch nie so gemacht! (Wir sind in Deutschland)
 - Ist nicht von uns! (Not invented here)
- Persönliche Animositäten
 - Alte Rechnungen werden beglichen (Für eine Handvoll Dollar)
 - Neue Feindschaften bilden sich ()

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



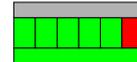
30
11

Beispiel: Die Methode „Inspektion“

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



31
11

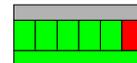
Beispiel: Methode „Inspektion“ Ziel und Anwendungsbereich

- Ziel:
 - Auffinden, Klassifizieren, Beheben von Fehlern im Prüfgegenstand
 - Validation
- Anwendbar auf:
 - im Prinzip auf alle Artefakte für die es ein Größenmaß gibt
 - (Programm: LoC, Entwurf: NoM, Textdokument: Seiten)
 - Kalibrierung/Bezugsgröße: Sprache, Stil- & Formatvorgaben
 - Inspektionsgruppe muß kompetent sein (doppelter Wortsinn)
 - mittlere und große Projekte, sehr frühe bis mittlere Phasen
- Einfache, aber wirkungsvolle Methode

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



32
11

Beispiel: Inspektion Durchführung

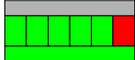
Schritt	Wer	Aktivitäten, Umfang, Dauer
Planung	M, A (, I)	Vorbedingungen prüfen Termine & Gegenstand festlegen
Überblick	M, A, I ₁ , ..., I _n	Autor stellt Gegenstände vor (ca. 1h) Gruppenabstimmung
Vorbereitung	M, I (, A)	Durchlesen (125NLoC/h·Person) Checklisten abarbeiten
Inspektion	M, A, I	L. trägt vor, I.´en unterbrechen, A. erläutert (90-125 NLoC/h, max.2h) Protokoll durch Moderator
Korrektur	A	gemäß Protokoll (nur diese Fehler!)
Abnahme	M, A	ggf. weitere Sitzung

} 3-5h

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



33
11

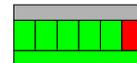
Beispiel: Inspektion Hinweise, Varianten

- Alle Rollen werden aus dem Projekt besetzt (wechselseitige Begutachtung verhindert Ausreißer).
- Kann bei sehr kleinen Projekten schwierig sein. Dafür gibt es („Walk Through“): Autor=Moderator=Leser, Inspektor stellt Fragen
- Re-Inspektion notwendig bei:
 - Korrekturen verändern Prüfgegenstand um mehr als 5% (z.B. NLoC/δNLoC)
 - #Fehler > 2•(φ frühere Inspektionen) ≈ 3-20 funktionale Fehler/1000 LoC
 - Killer-Fehler
- Beachte: Grundlagen für quantitative Erfassung werden gelegt!
 - Zeitreihen von Fehlerdichten
 - Raten/Aufwände von Reparaturen

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU

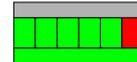


34
11

Klassische Software-Lebenszyklen

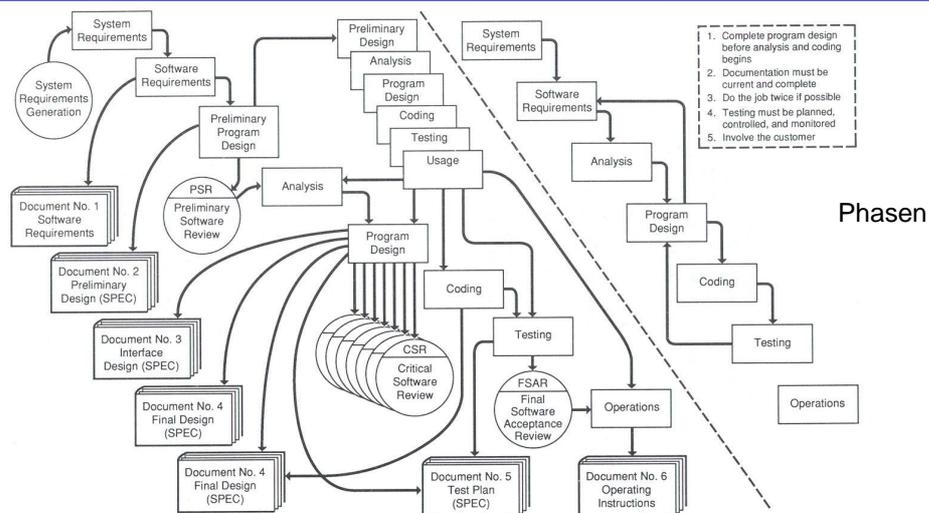
22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



35
11

Klassische Vorgehensmodelle 1 Das Wasserfall-Modell



Walker Royce
IBM, 19??

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



36
11

Klassische Vorgehensmodelle 1

Das Wasserfall-Modell

- **Vorteile**

- sehr einfach durchzuführen
- daher auch für sehr große Projekte anwendbar
- sehr effizient bei bekannten und konstanten Anforderungen
- Prozessverbesserung ist machbar

- **Aber**

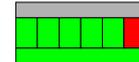
- Risiken gesammelt am Schluß („Big Bang“)

=> **geeignet für geplantes Vorgehen**

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

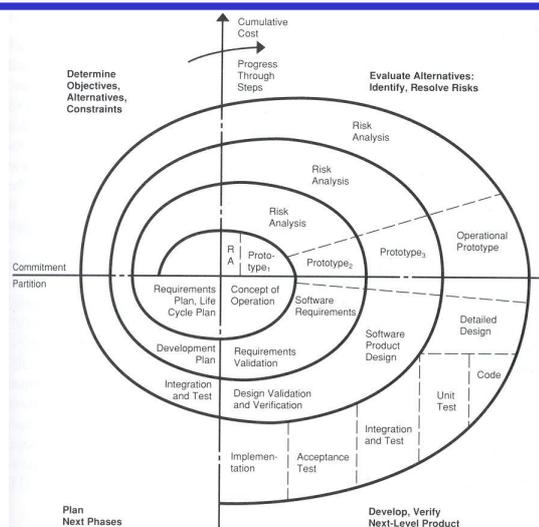
LMU



37
11

Klassische Vorgehensmodelle 2

Das Spiral-Modell



Steuerung durch Rückkopplungsschleife

360° ≈ Mini-Wasserfall

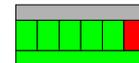
Hohes Risiko-Bewußtsein

Barry Boehm,
Uni Stanford, 1988

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



38
11

Abwägung Wasserfall vs. Iterativ

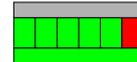
- Vorteile
 - flexibler in der Reaktion auf Umgebungsänderungen
 - verteilt Risiken durch inkrementelle Auslieferung
 - Prozessverbesserung sehr ähnlich machbar
- Aber
 - viel aufwändiger in der Durchführung
 - erfordert hohe Qualifikation der Mitarbeiter

=> geeignet für evolutionäres Vorgehen

22.1.2002
Prozeß 1: Klassisch

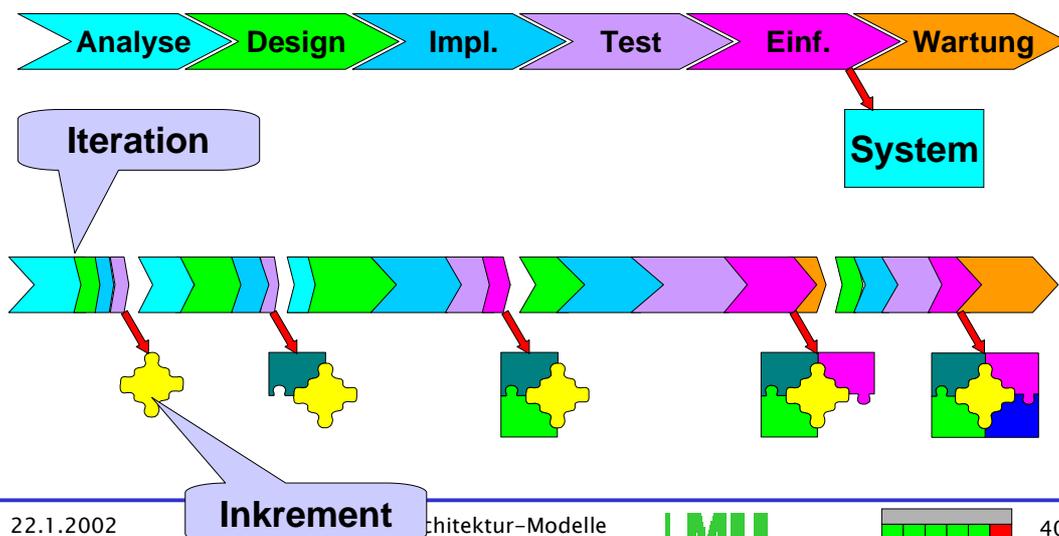
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



39
11

Abwägung Wasserfall/Big Bang vs. Iterativ/Inkrementell

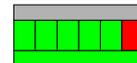


22.1.2002
Prozeß 1: Klassisch

Inkrement

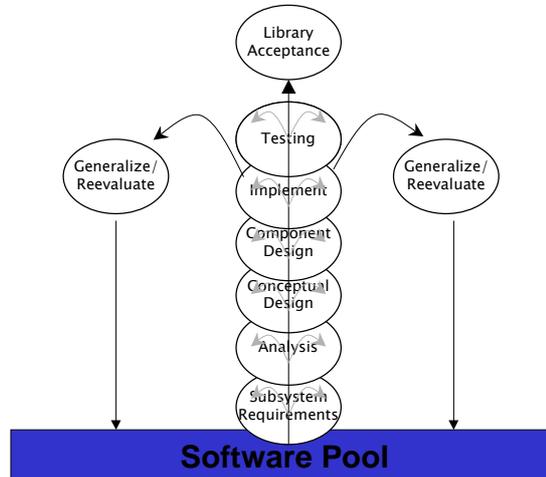
chitektur-Modelle
Dr. Harald Störrle

LMU



40
11

Klassische Vorgehensmodelle 3 Das Springbrunnen-Modell



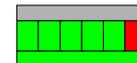
Phasen und Projekte
verschwimmen

Henderson-Sellers & Edwards
1990

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



41
11

Klassische Vorgehensmodelle 3 Das Springbrunnen-Modell

- Vorteile
 - Betrachtet nicht (nur ein) Projekt, sondern Wissen und „Überbleibsel“: dadurch realistischer
 - Aber
 - Wie soll das operationalisiert werden?
 - Funktioniert nicht für Hire-and-Fire-Kultur
- => geeignet für Wiederverwendung
(z.B. Produktlinien)

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



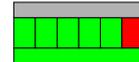
42
11

Beispiele: Industrielle Prozesse (klassisch)

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



43
11

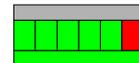
Industrielle Vorgehensmodelle 1: „Vorgehensmodell des Bundes“ (V-Modell)

- Aktuelle Version von 1997
- vor allem in Deutschland, weitverbreitet in Banken, Versicherungen, im E/RT-Bereich, Automotive, Aerospace...
- obligatorisch für Aufträge des Bundes
- geeignet für HW/SW-Integration

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

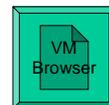
LMU



44
11

Industrielle Vorgehensmodelle 1: „Vorgehensmodell des Bundes“ (V-Modell)

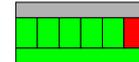
- 4 Submodelle
 - SE – Systemerstellung
 - PM – Projektmanagement
 - KM – Konfigurationsmanagement
 - QM – Qualitätsmanagement
 - plus eine Handvoll einzelner Bestandteile, z.B. Tailoring.
- ca. 50 Prozesse und Teilprozesse
- relativ abstrakt und (dadurch) robust
- gut anpassbar



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



45
11

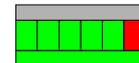
Industrielle Vorgehensmodelle 1: Aufgaben zum V-Modell

- Ist das V-Modell auf das Wasserfall-Modell festgelegt?
- Kann ich mit dem V-Modell einen
 - iterativen
 - inkrementellen
 - einen Wiederverwendungs-orientiertenProzess fahren?

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



46
11

Industrielle Vorgehensmodelle 2: „Rational Unified Process“ (RUP)

- Der RUP...
- ...hie frher Objectory (Jacobson)

- ...bezeichnet sich als
 - inkrementell & iterativ
 - Architektur-zentriert
 - Anwendungsfall-getrieben

- ...benutzt die UML, und ist fixiert auf OO-Software.

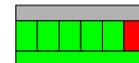
- ...braucht massive Werkzeuguntersttzung (Rational Tool Suite).

- ...ist sehr komplex, und damit ist schwierig umzusetzen

22.1.2002
Proze 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Strrle

LMU



47
11

Industrielle Vorgehensmodelle 2: „Rational Unified Process“ (RUP)

- | | |
|---|---|
| <ul style="list-style-type: none">• 9 Prozesse<ul style="list-style-type: none">• Business Modling• Requirements• Analysis & Design• Implementation• Test• Deployment• Config. & Change Mgmt.• Project Mgmt.• Environment | <ul style="list-style-type: none">• 4 Phasen<ul style="list-style-type: none">• Inception• Elaboration• Construction• Transition <p>rein zeitlich, kein Bezug zu Wasserfall-Phasen</p> |
|---|---|

22.1.2002
Proze 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Strrle

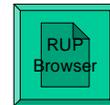
LMU



48
11

Industrielle Vorgehensmodelle 2: „Rational Unified Process“ (RUP)

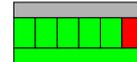
- RUP ist sehr detailliert (und dadurch komplex).
- RUP braucht massive Werkzeugunterstützung (Rational Tool Suite).
- RUP ist aufwändig, und damit teuer.
- RUP ist fixiert auf OO-Software.



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



49
11

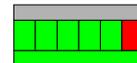
Industrielle Vorgehensmodelle 2: Aufgaben zum RUP

- Gibt es im RUP Punkte, an denen
 - Integration von Zulieferungen (z.B. COTS)
 - HW/SW-Integration (z.B. Automotive)stattfinden könnte?
- Identifizieren Sie diese Punkte!
- Schlagen Sie vor, wie die Integration von HW-Entwicklung ablaufen könnte.

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



50
11

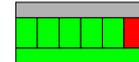
Industrielle Vorgehensmodelle 3: ISO 12207

- 4 Gruppen mit 18 Prozessen
- Prozesse haben Activities, diese haben Tasks
- Sehr detailliert gegliedert und beschrieben, sehr umfassend
- modern, und international standardisiert

22.1.2002
Prozeß 1: Klassisch

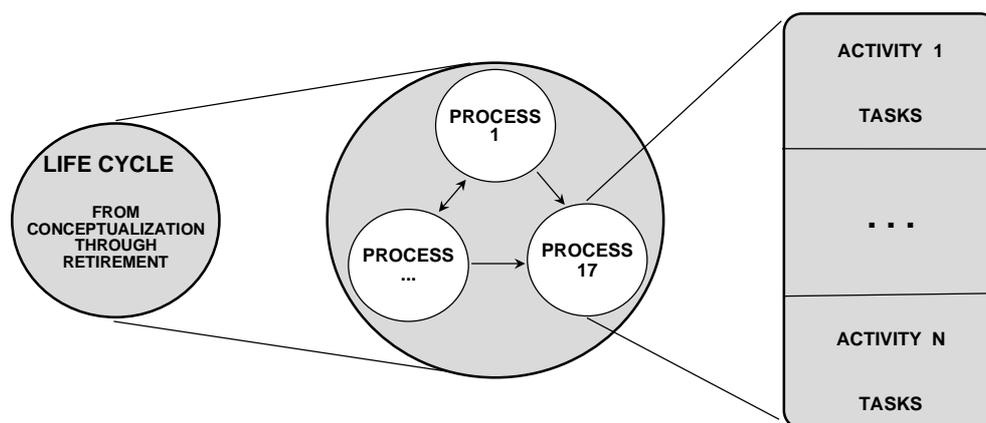
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



51
11

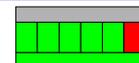
Industrielle Vorgehensmodelle 3: ISO 12207



22.1.2002
Prozeß 1: Klassisch

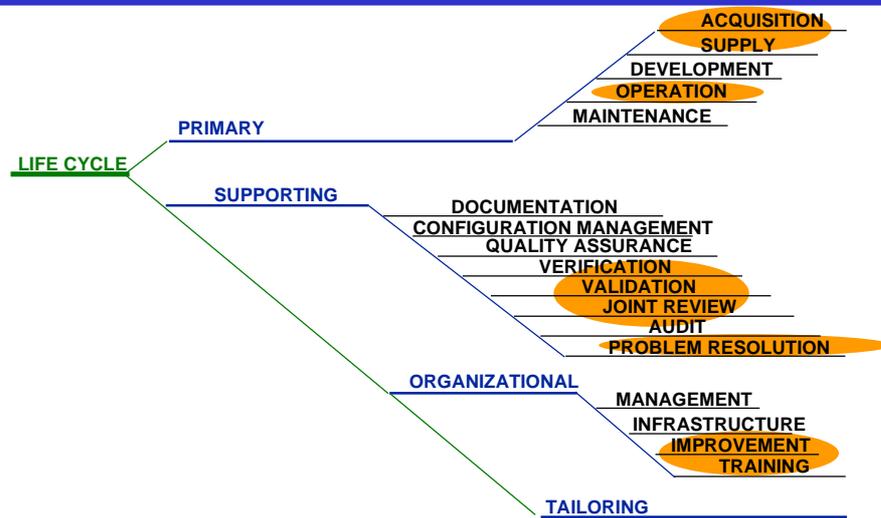
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



52
11

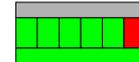
Industrielle Vorgehensmodelle 3: ISO 12207



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



53
11

Drittes Thema: Organisation

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



54
11

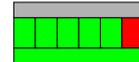
Management-Grundansatz

- Sun Tzu
- Iterieren
 - Lagefeststellung
 - Ursachenforschung
 - Lösungsfindung
 - Umsetzen

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



55
11

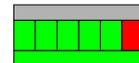
Brooks' Gesetz (1)

- Der Zeitbedarf für jede Tätigkeit in Teams besteht immer aus zwei Elementen:
 - 1 – Die eigentliche Arbeit
 - 2 – Der Aufwand für Verwaltung und Abstimmung
- Ohne (2) würde die notwendige Dauer d eines Projektes mit der Zahl n der Mitarbeiter abnehmen, also $d \approx 1/t$.

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



56
11

Brooks´ Gesetz (2)

- Aber jeder Mitarbeiter muß mit jedem anderen kommunizieren (Aufwand jeweils k). Also ist die tatsächliche Dauer

$$d \approx 1/t + k \binom{n}{2} \approx 1/t + k \frac{1}{2} n^2$$

- Dadurch kommt es zu Brooks´ Gesetz: „*Adding manpower to a late project makes it later*“
- Mit anderen Worten: Schlechte Planung am Anfang führt unweigerlich zur Überziehung von Terminen (und Budget).

- Wie kann man, trotz Brooks´ Gesetz, die Herstellung großer Software Systeme organisieren?
- Welche Organisationsformen gibt es überhaupt?

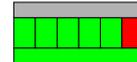
Organisationsformen

- Wie kann man trotzdem die Herstellung großer Software Systeme organisieren?
 - Team
 - Hierarchisch (Baumartig)
 - Fraktal (rekursives Netzwerk)

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



59
11

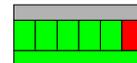
Organisationsform 1: Baum

- Vor- und Nachteile
 - beliebige Größe
 - Kommunikationsaufwand steigt, Qualität sinkt
 - sehr stabil gegenüber äußerem Druck und Teilausfällen
 - zuverlässig nur mit großem Kontroll- und Restriktionsaufwand
 - hoher Verwaltungs- und Veränderungsaufwand
- Gut bei
 - bei stabilen äußeren Bedingungen (Effizienz) und bei Druck
- Beispiele
 - Armeen
 - Automobilproduktion in USA und Europa vor 1990 (Tailorismus)
- ...und in der Software
 - VM´97

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



60
11

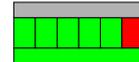
Organisationsform 2: Team

- **Vor- und Nachteile**
 - schnelle, genaue und vollständige Kommunikation
 - zuverlässig durch Gruppendynamik und -druck
 - Nur bis zu maximal 8 Leuten machbar
- **Gut bei**
 - gut bei rasch sich ändernden Randbedingungen
- **Klassische Beispiele**
- **...und in der Software**
 - Chefprogrammiererteam

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



61
11

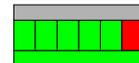
Organisationsform 3: Fraktal

- **Fraktal (rekursives Netz) kombiniert (ein Stück weit) die Vorteile beider Systeme**
- **Vor- und Nachteile**
 - Zusammenhalt durch geteilte Vision (**-> Architektur!**)
 - potentiell sehr schlagkräftige, aber selten dauerhaft
- **Gut bei**
- **Beispiele**
 - Automobilherstellung weltweit seit ca. 10 Jahren
 - Gore
 - wissenschaftliche Gemeinschaften
 - manche politische/religiöse Bewegungen

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



62
11

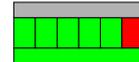
Organisationsformen der Softwareherstellung

- Wo kommt welche Organisationsform vor?
- Ist das jeweilige Vorkommen vergleichbar mit der Herstellung von Software? Wo ja, wo nein?

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



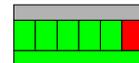
63
11

**Nächstes Thema:
Prozessverbesserung**

22.1.2002
Prozeß 1: Klassisch

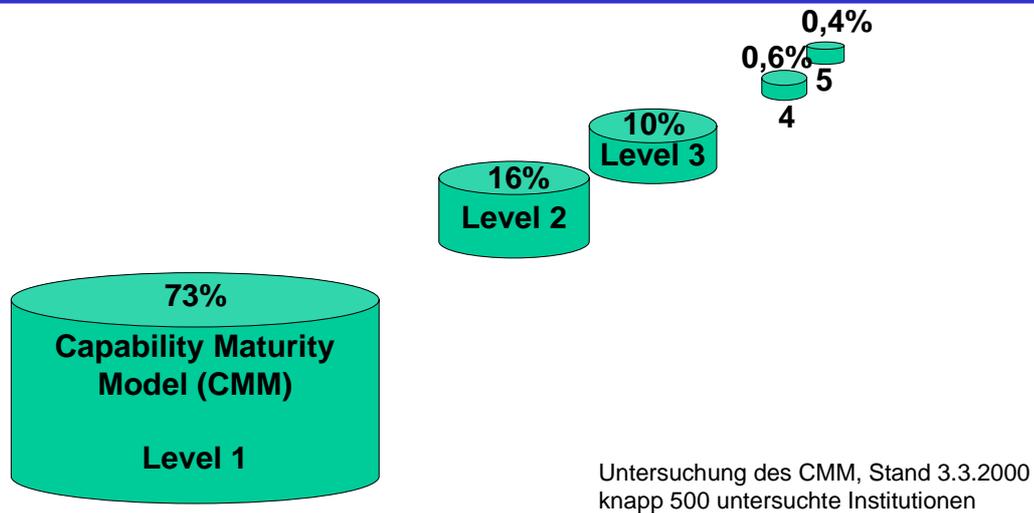
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



64
11

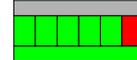
Software-Projekte scheitern sehr oft: Was bringt Prozessverbesserung?



22.1.2002
Prozeß 1: Klassisch

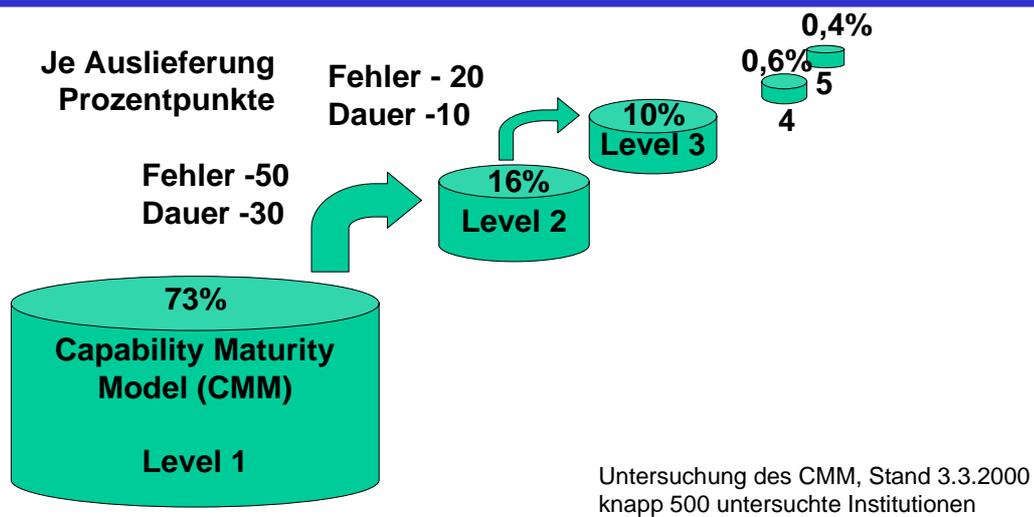
VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



65
11

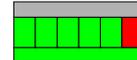
Software-Projekte scheitern sehr oft: Was bringt Prozessverbesserung?



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



66
11

Software-Projekte scheitern sehr oft: Was bringt Prozessverbesserung?

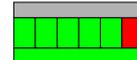
CMM Level	Monate	Personen-Monate	gefundene Fehler	ausgelieferte Fehler	Gesamtkosten (1000 US\$)
1	29,8	593,5	1348	61	5.440
2	18,5	143	328	12	1.311
3	15,2	79,5	182	7	728
4	12,5	42,8	96	5	392
5	9	16	37	1	146

Daten für ein 200 KLoC-Projekt (Schätzung der SemaTech)

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

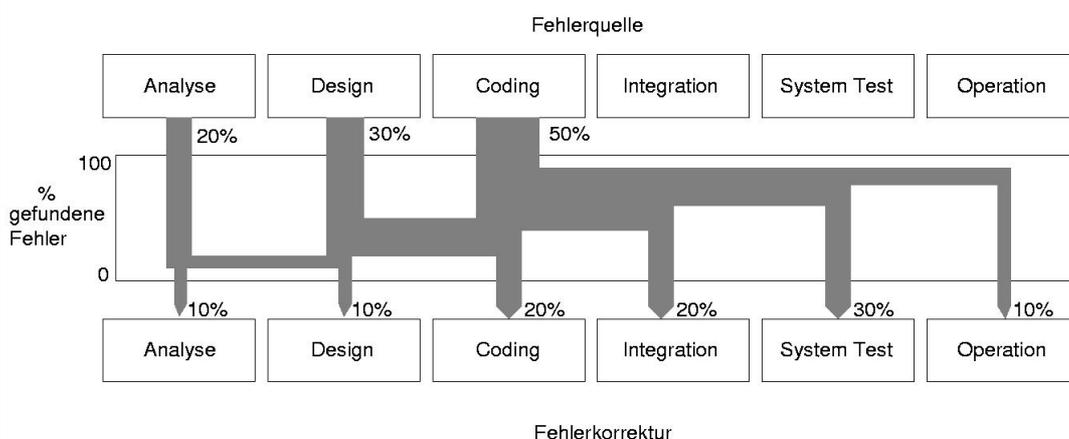
LMU



67
11

Wie können gleichzeitig Kosten und Dauer sinken, bei verbesserter Qualität?

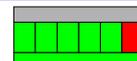
=> Vermeidung sinnloser Arbeit



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



68
11

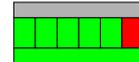
Software-Projekte scheitern sehr oft: Was ist das Fazit?

- Man sollte sich um den Software-Prozess kümmern– es lohnt sich.
- Das wichtigste ist, sich überhaupt zu kümmern.
Wie das geschieht, ist eher zweitrangig.
- Dies sind empirisch belegte Fakten, keine Meinungen oder Moden.
- Aber wie?

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



69
11

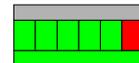
Überleitung zum nächsten Mal

- Das CMM (und SPICE/ISO15504) sind sehr stark auf Wasserfallmodell ausgerichtet.
- Aber heutzutage kann dieses Vorgehen in vielen Fällen gar nicht (mehr) angewendet werden!
- Daraus folgen die Leitfragen für das nächste Mal:
 - Wie sind denn die Anforderungen heute genau?
 - Wie definiere ich Prozesse, damit sie diesen Anforderungen genügen?
 - Wie kann ich solche Prozesse steuern?
 - Wie kann man solche Prozesse kalibrieren und ihre Implementierung verbessern?

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



70
11

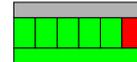
Wichtige Architektur-Prozesse

- Änderung hatten wir
- Anforderungsverwaltung!!
- Architekturmigration
- Releaseplanung

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



71
11

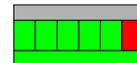
Literatur Empfehlenswert

- Bücher von Watts Humphrey:
 - Managing the Software Process
 - The Capability Maturity Model
 - The Personal Software Process

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



72
11

Literatur

Weniger empfehlenswert

- Bücher von Scott Ambler

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



73
11

Literatur

Spezielle Prozesse

- Wiemers et al.: *Das Vorgehensmodell des Bundes Oldenbourg*, 1997
- Jacobson, Booch, Rumbaugh: *The Unified Software Development Process* Addison-Wesley, 2000
- D`Souza & Wills: *Catalysis* Addison-Wesley, 1999
- Beck: *Extreme Programming Explained* Addison-Wesley, 2001

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



74
11

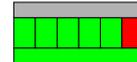
Literatur Code-Inspektion

- Wheeler, Brykczynski, Meeson: *Software Inspection* IEEE Press, 1996
- Darin ausführliche Literatur

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



75
11

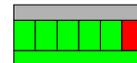
Literatur Organisation und Management

- Womack, Roos, Jones: *Die zweite Revolution in der Automobilindustrie* Campus, 1991
- Krause: *Die Kunst des Kriegs für Führungskräfte* Ueberreuther, 1995
- Malik: *Führen, Leisten, Leben DVA, 2001*
- DeMarco, Lister: *Wien wartet auf Dich!* Hanser 1991

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



76
11

Vorlesung
Software Architektur-Modelle
Prozesse 2: moderne Vorgehensmodelle, Prozeßverbesserung

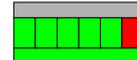
Dr. Harald Störrle

Ludwig-Maximilians-Universität München
Wintersemester 2001

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



77
11

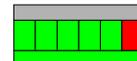
Es gibt viele vernachlässigte Prozesse
„Prozesswaisen“

- Akquise
- Auslieferung/Abnahme
 - Vertragsgestaltung, Seitenabsprachen, Schiedsgerichte, Klagewege, Mediation
- Betrieb
- Konfigurations-Verwaltung
- Änderungsverwaltung
- Qualitätssicherung

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



78
11

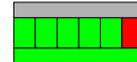
Motivation für modernere Prozesse

- Outsourcing / contracting out von Teilen ist ein allg. Trend in der Industrie, und besonders auch in der IT.
- Daher werden Zuliefer-, Ausliefer, und Beschaffungsprozesse immer wichtiger, aber auch die flächendeckende Prozessexpertise insgesamt.
- Integration statt Herstellung (aber Teile und die Integration immer noch in einer Hand): gab auch bislang Integration, aber weniger, und die Jobs wurden intern erledigt, nicht über verschiedene Firmen hinweg – Firmenweite Standardisierung reichte aus

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



79
11

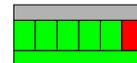
Prozessverbesserung

- SPI
- verschiedene Arten von CMM
 - Sw
 - People
 - Sw Acquisition
 - Integrated Product Development
 - Systems Engineering
 - Test

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle

LMU



80
11

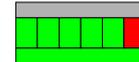
Industrielle Vorgehensmodelle 3: ISO 15504

- 6 Prozesskategorien
 - CUS : Customer–Supplier Relations
 - ENG : Engineering
 - PRO : Project
 - SUP : Support
 - ORG : Organization

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur–Modelle
Dr. Harald Störrle

LMU



81
11

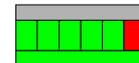
Industrielle Vorgehensmodelle 3: CMM v1.1

- Key Process Areas (KPAs)

22.1.2002
Prozeß 1: Klassisch

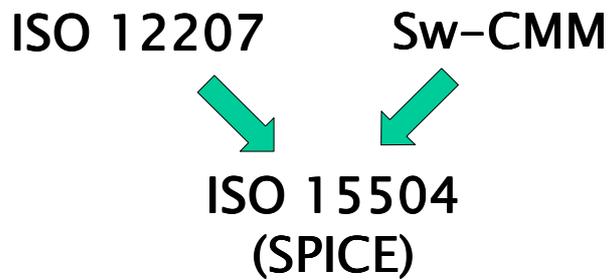
VL Software Architektur–Modelle
Dr. Harald Störrle

LMU



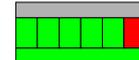
82
11

Industrielle Vorgehensmodelle 4: ISO-15504



22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



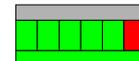
83
11

Industrielle Vorgehensmodelle: Gegenüberstellung

ISO 12207 Gruppe	ISO 15504 Process Category	VM '97 Submodelle	RUP 2001 Main Process
Primary	ENC	SP	Business Modeling
Supporting	PRG	PM	Requirements
Organizational	MAN	RM	Analysis & Design
Tailoring	CUS	CM	Implementation
	SUP		Test
	ORG		Deployment
			Config. & Change Mgmt.
			Project Mgmt.
			Environment
	Reuse, HR, Measurement	Tailoring SPI	

22.1.2002
Prozeß 1: Klassisch

VL Software Architektur-Modelle
Dr. Harald Störrle



84
11