

**Vortrag zum Forschungsfreisemester  
durchgeführt in Florianopolis SC (BR)**

Prof. Dr. Wolfgang Schramm

Informatik-Berichte

Hochschule Mannheim – Fakultät für Informatik

Computer Science Reports

Mannheim University of Applied Sciences – Computer Science Department

CSR 001.06

August 2006

URL: <http://www.informatik.hs-mannheim.de/reports>



# hochschule mannheim

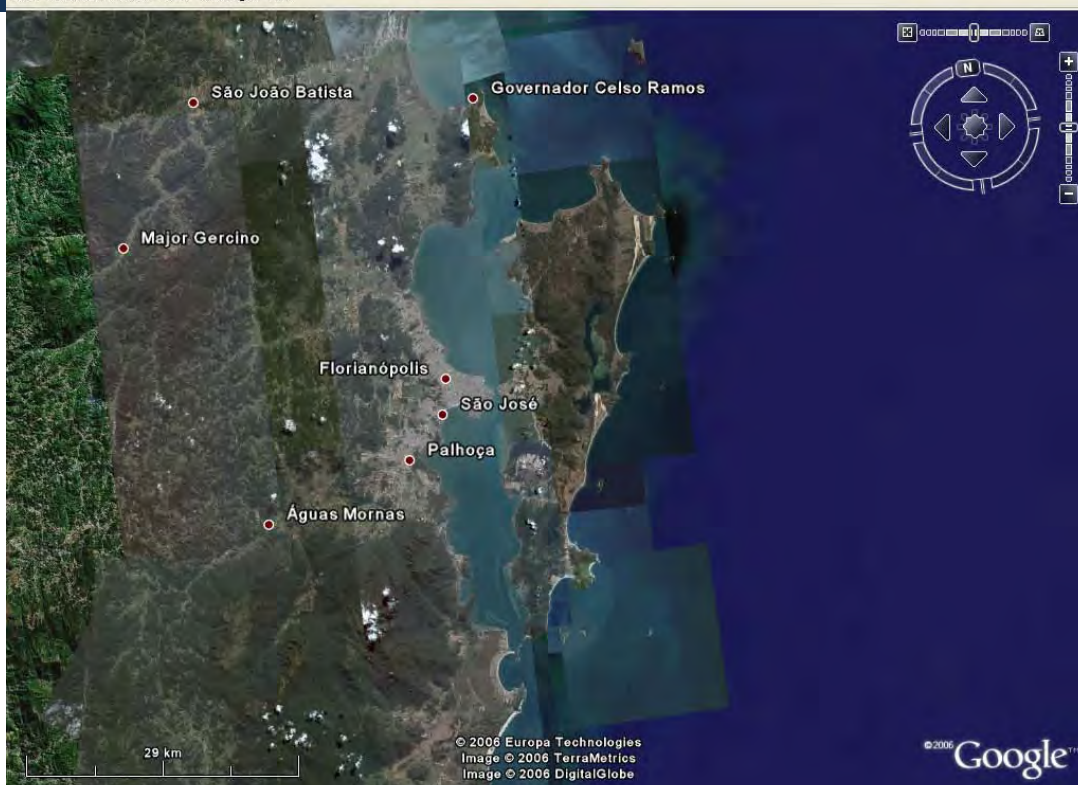
## Vortrag zum Forschungsfreisemester durchgeführt in Florianópolis SC (BR)

Prof. Dr. Wolfgang Schramm



## Wie ist die Lage?

Datei Bearbeiten Ansicht Tools Hinzufügen Hilfe





# Software Prozesse als Basis für Software Qualitätsmanagement



## Gliederung

- **Motivation für Qualitätsmanagement**
- **Qualitätsmanagement**
- **Prozessorientierte Ansätze – kurzer Überblick**
- **Software Prozess Projekt an der UFSC**
- **Fachbereich Informatik an der UFSC**
- **Leben in Florianopolis**



- **Venussonde fliegt 1979 am Ziel vorbei.**

Ursache: Verwechslung eines Punktes mit einem Komma in einem FORTRAN Programm.

- **Salomon Brothers, Inc. erleidet 1992 einen erheblichen Verlust.**

Ursache: Ein Angestellter erteilt einen Auftrag über 11 Mio. Aktien statt über Aktien im Wert von 11 Mio. \$. Ein zweiter Angestellter versäumt, den Auftrag zu prüfen; das Informationssystem macht zu spät auf den Fehler aufmerksam.

- **Ariane-5 zerstört sich 1996 ca. 36 Sekunden nach dem Start selbst.**

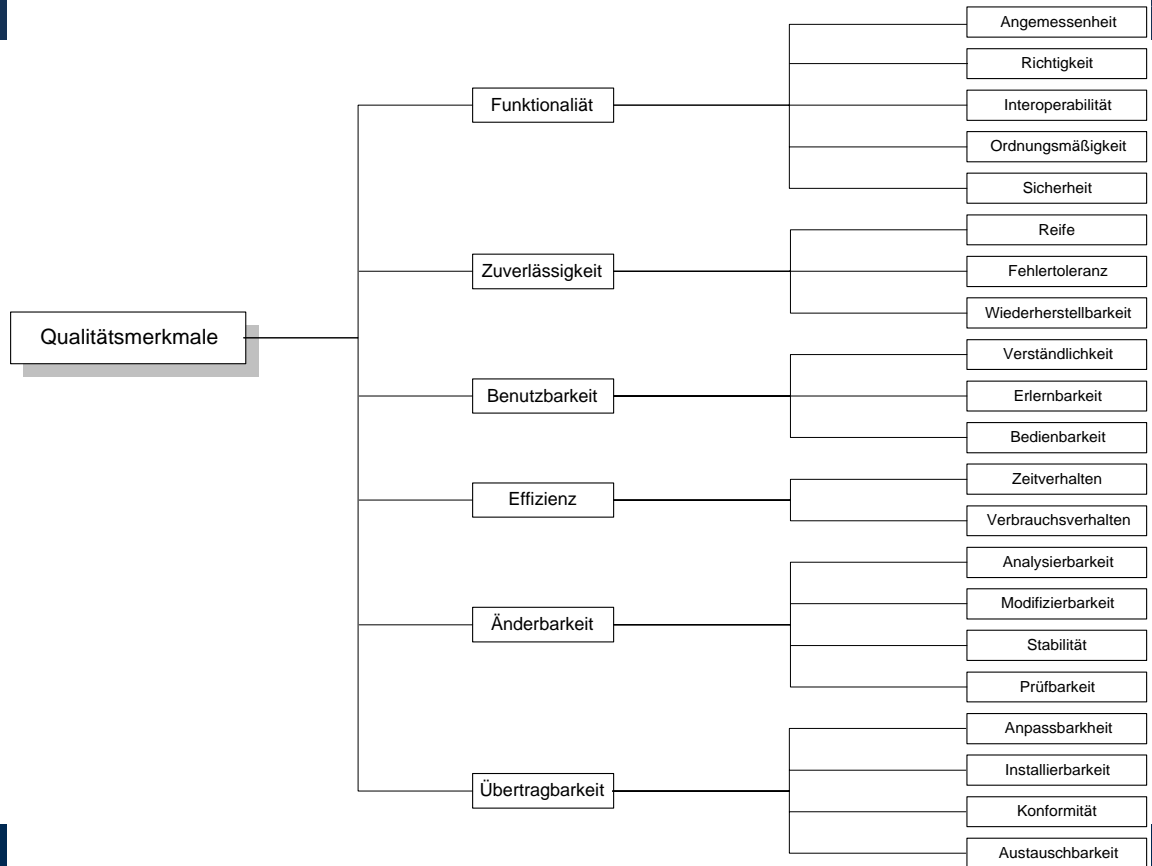
Ursache: Zusammentreffen verschiedener Faktoren: Es wurde nicht mehr benötigte Software des Vorgängermodells benutzt. Aus Effizienzgründen, wurde auf bestimmte Prüfungen zur Laufzeit verzichtet. Die höhere Geschwindigkeit der Ariane-5 führte dazu, dass es zu einem ungeprüften Bereichsüberlauf und damit zur Sprengung aus Sicherheitsgründen kam.



Projekt	Verspätung	Verlust
Deutsches Mautsystem „Toll Collect“	2 Jahre	~ 2,2 Milliarden €
American Airlines „Autovermietung“	7 Jahre	165 Mio. \$
Denver Flughafen „Gepäckverteilung“	2 Jahre	750 Mio. \$
London „Elektronische Börse“	12 Jahre	800 Mio. £
London „Krankenwagen Leitsystem“	5 Jahre	12 Mio. £ + 46 Menschenleben



# Qualitätsmodell für Software (nach DIN 66272 / ISO 9126)



Folie 6



## Qualitätsmodelle dienen ...

### ... als Sammlung von

- Entwicklungsleitlinien für den Auftragnehmer.
- Bewertungskriterien für den Auftraggeber.

### ... zur Beantwortung folgender Fragen

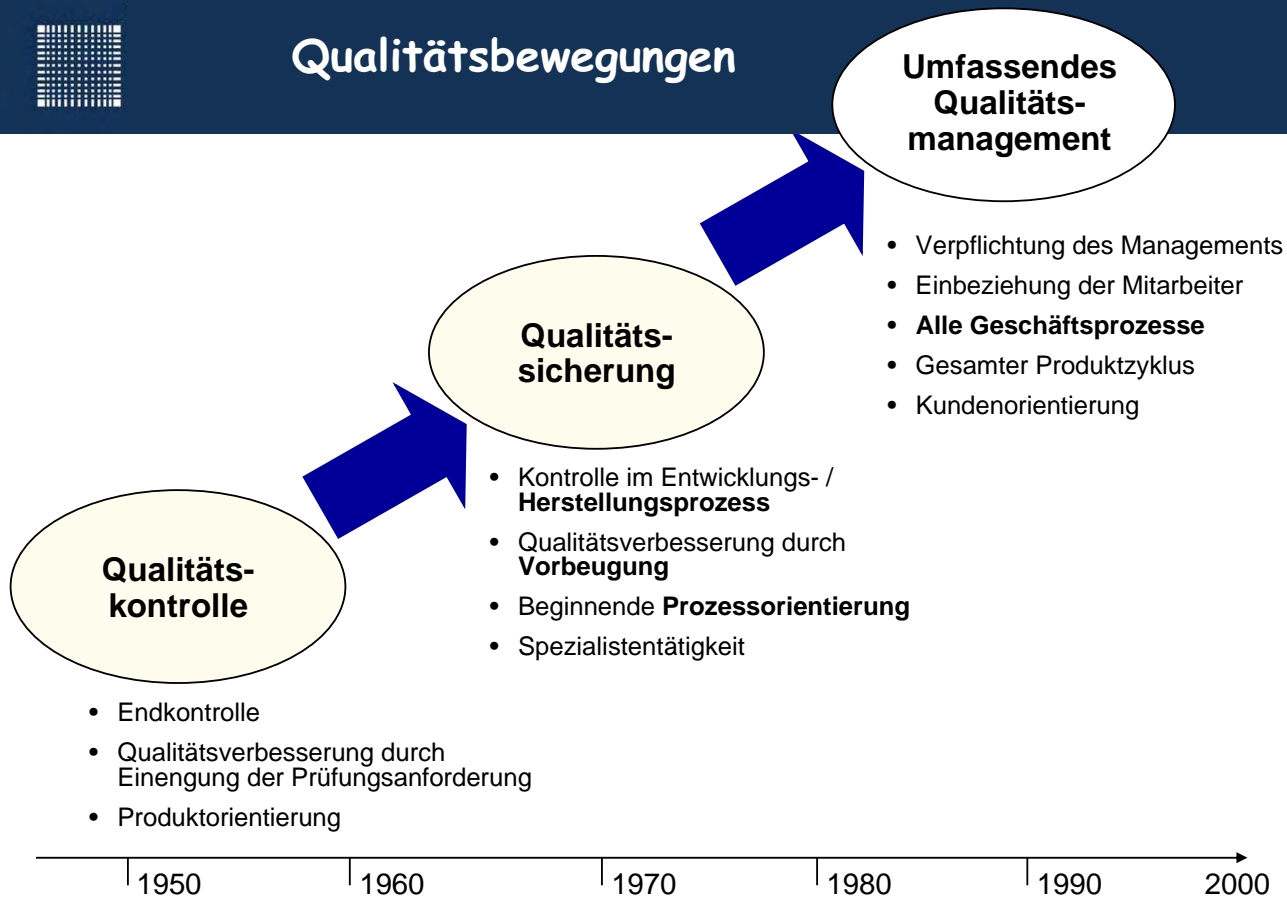
- Was versteht der Auftraggeber unter den einzelnen Qualitätsmerkmalen ?
- Wie gewichtet er die einzelnen Merkmale ?
- Wie will er die Ausprägung eines Qualitätsmerkmals überprüfen ?

⇒ **Operationalisierung des Qualitätsbegriffs**

Folie 7



# Qualitätsbewegungen



# Zwei Ansätze des Software-Qualitätsmanagements

## produktorientiertes Qualitätsmanagement

Software-Produkte und Zwischenergebnisse werden auf vorher festgelegte Qualitätsmerkmale überprüft.

## prozessorientiertes Qualitätsmanagement

Bezieht sich auf den Entwicklungsprozess der Software. Dazu gehören Methoden, Werkzeuge, Richtlinien und Standards.



## Managementkonzept

- das Software-Entwicklung als einen Prozess betrachtet,
- der durch bestimmte, auf Wiederholbarkeit und Vermeidung von Qualitätsmängeln zielende Maßnahmen
- zu planen, zu steuern und zu kontrollieren ist.

## Grundlegende These des PSQM besagt,

dass für die Steigerung des Erfolgs von Software-Unternehmen die Verbesserung des Software-Prozesses von höchster strategischer Bedeutung ist.



## Prozessorientierte Ansätze

- **Total Quality Management (TQM)** - *Demming*
- **ISO 900x-Normen** - *ISO*
- **Capability Maturity Model (CMM)** bis 2003 - *SEI an der CMU*
- **Capability Maturity Model Integration (CMMI)** seit 2002 - *SEI an der CMU*
- **Software Process Improvement and Capability Determination (SPICE)** / ISO 15504 - *ISO*
- **Bootstrap - Ansatz** - *Esprit-Projekt*
- **Personal Software Prozess (SM)** ist eine Methode für Software-Entwickler zur Selbstoptimierung. - *SEI an der CMU von W. Humphrey*



- Ein Softwareprozess ist eine Menge von Aktivitäten und damit zusammenhängenden Ergebnissen, die zur Herstellung eines Softwareproduktes führen.
- Feststellungen:
  - Die Vielfalt der Softwareprozesse ist groß.
  - Es gibt keinen idealen Prozess.
  - Verschiedene Organisationen haben völlig unterschiedliche Prozesse.
  - Folge: Art und Anzahl der Aktivitäten und der Ergebnisse ist sehr unterschiedlich



- Gemeinsamkeiten von Softwareprozessen.
- Spezifikation: Definition der Funktion, Randbedingungen (Anforderungen).
- Entwurf und Implementierung (Entwicklung): Erstellen der Software, gemäß den Anforderungen und Qualitätssicherung.
- Validierung: Sicherstellen, dass die Software tut, was der Kunde will.
- Evolution: Weiterentwicklung wegen sich ändernden Wünschen oder Randbedingungen.



### **sequentielle:**

- "Code and fix"-Modelle
- Phasen- und Wasserfall-Modelle (seit ca. 1970)
- Transformationsmodelle, formale Modelle (seit ca. 1975)

### **nicht-sequentielle:**

- V-Modelle
- Modelle zum Prototyping, zur inkrementellen Systementwicklung
- Spiral- und prozessorientierte Modelle [B. Boehm 1988]
- Evolutionäre, inkrementelle Entwicklungsmodelle [M. M. Lehman 1980]
- Wiederverwendungsorientierte Entwicklung (opportunistisch, geplant)

vgl. auch [Hesse/Merbeth/Frölich 1992], [Boehm88]



### **In der Praxis sehr häufig vorgefunden (IPAS Projekt):**

- **Offiziell:**
  - vorwiegend Wasserfall, zuweilen Prototyping oder inkrementelle Softwareentwicklung.
  - Formale Modelle werden nur in wenigen Projekten eingesetzt.
  - Wiederverwendbarkeit kommt in vielen Prozessen vor, allerdings weniger geplant als implizit.
- **Inoffiziell, subversiv:**
  - alle Formen, weit weniger Wasserfall, dafür mehr Prototyping, inkrementelle oder evolutionäre Softwareentwicklung, aber auch "Code and fix" .

[Hesse96]





## Capability Maturity Model (CMM) - Reifegrade (1)

### *Defined - 3*

- Unternehmensweit gültiger, dokumentierter Softwareprozess.
- Situationsspez., nachvollziehbare Anpassungen der Regelungen.
- Organisatorische Einheit (SEPG) pflegt den Prozess.
- Mitarbeiter und Management verstehen den Prozess.

### *Repeatable - 2*

- Grundlegende Projektmanagementaufgaben sind etabliert.
- Planung basiert auf Erfahrungen mit ähnlichen Projekten.
- Softwareprozess ist rudimentär vorhanden und stabil.
- Erfolge können unter ähnlichen Bedingungen wiederholt werden.

### *Initial - 1*

- Projekte ad hoc strukturiert, tendenziell chaotisch.
- Keine Vorgaben / Hilfsmittel zur Planung u. Steuerung der Projekte.
- Termine, Kosten, Funktionalität, Qualität kaum vorhersagbar.
- Erfolg abhängig von Motivation und Qualifikation der Mitarbeiter.



## Capability Maturity Model (CMM) - Reifegrade (2)

### *Optimized - 5*

- Unternehmen ist auf kontinuierliche Verbesserung eingestellt.
- Quantitative Prozessdaten → Erfahrungen / Voraussagen.
- Aktives Innovationsmanagement / Technologiemanagement:
  - gesteuerte Pilotanwendungen,
  - Quantitative Kosten-Nutzen-Analysen.

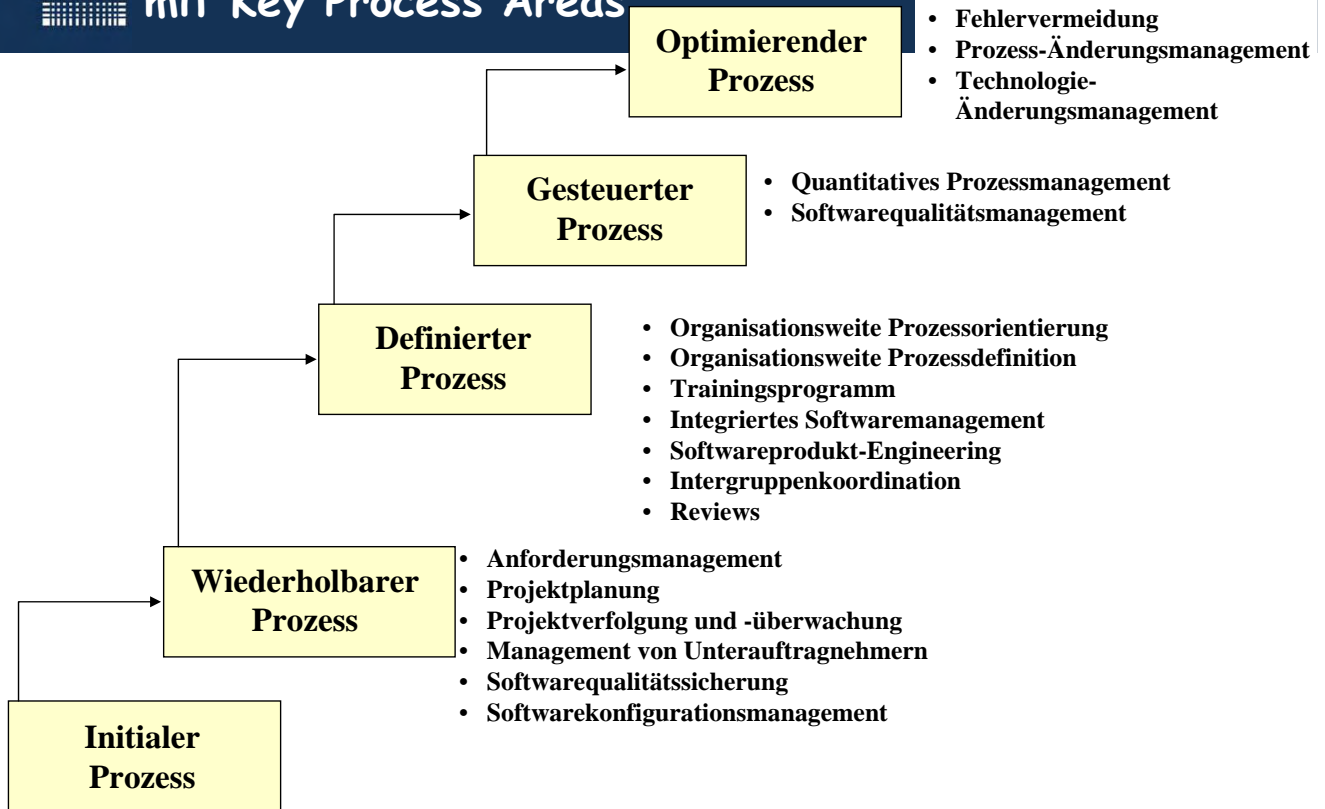
### *Managed - 4*

- Quantitative Ziele für Produkt und Teilprozesse.
- Messung wichtiger Zielgrößen → Prozesserfahrungsbasis.
- Einfluss der Maßnahmen auf Prozess-/Produktqualität wird verstanden.
- Ziele werden erreicht / Erfahrungsbasis erklärt Abweichungen.
- Frühzeitige Gegenmaßnahmen bei Abweichungen.

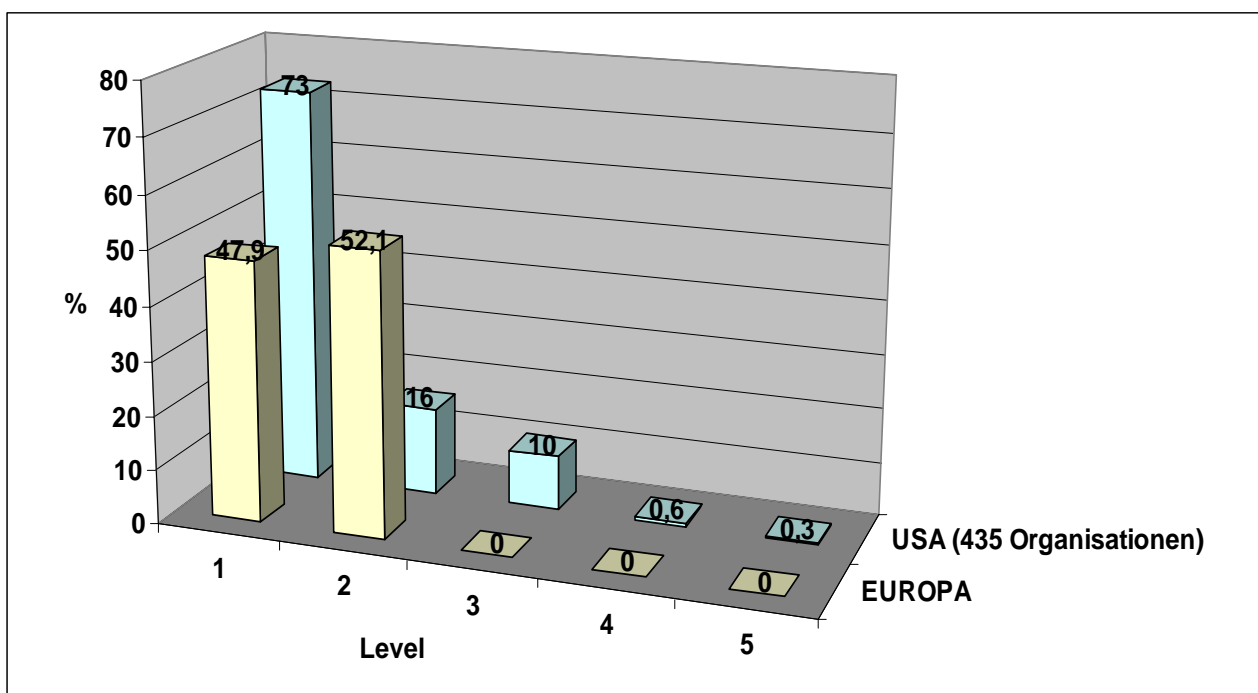
### *Defined - 3*



# Capability Maturity Model (CMM) - Reifegradmodell mit Key Process Areas



## CMM - Reifegradverteilung in der Praxis (1995)





## CMM - Bewertung (1): -

- Es gibt keine Garantie, dass Softwareprodukte erfolgreich entwickelt und alle Software Engineering Probleme adäquat gelöst werden.
- Das Modell ist technologielastig und berücksichtigt zu wenig personenbezogene Merkmale (z.B. Fähigkeit zur Kommunikation, Mitarbeiterkultur).
- Wichtige Schlüsselprozessbereiche, wie z.B. Risikomanagement und Wiederverwendung fehlen.
- Das Modell schreibt Tätigkeiten vor ohne explizite Beweisführung über deren Mehrwert.
- Um eine hohe Stufe zu erreichen, müssen alle Forderungen der niedrigeren Stufen erfüllt sein.
- Regeln für die Anwendbarkeit des Modell für kleinere Softwareorganisationen fehlen.
- Für die Stufen 4 und 5 gibt es nur wenige gesicherte Erkenntnisse.



## CMM - Bewertung (2): +

- Modell bietet Basis zur systematischen Verbesserung der Zielgrößen Qualität und Produktivität.
- Die veröffentlichten empirischen Untersuchungen zeigen, dass der Nutzen die Kosten rechtfertigt.
- Viele Firmen haben sich bisher auf Techniken und Werkzeuge konzentriert und den Entwicklungsprozess vernachlässigt ⇒ großes Verbesserungspotential.
- Modell erlaubt die Evaluation des aktuellen Prozesszustands einer Organisation ⇒ Vergleich mit anderen Organisationen.
- Bei der sorgfältigen Anwendung der CMM-Assessments werden Prozessschwachstellen identifiziert, die ideal für Verbesserungsprojekte sind.



## CMM - Ergebnisse von Prozessverbesserungsmaßnahmen (1)

	Minimum	Maximum	Ø
Jährliche Kosten für Prozessverbesserung (in US \$)	49.000	1.202.000	245.000
Jahre aktiv in Prozessverbesserung	1	9,5	3,5
Kosten der Prozessverbesserung pro SW-Entwickler (in US \$)	490	2.004	1.375
Produktivitätssteigerung pro Jahr	9 %	67%	35%

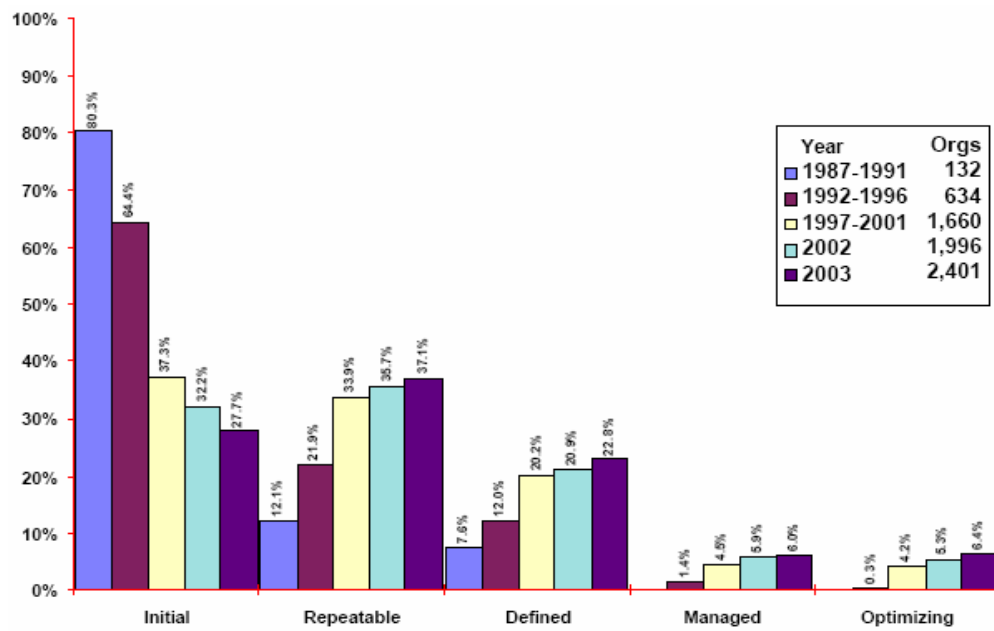


## CMM - Ergebnisse von Prozessverbesserungsmaßnahmen (2)

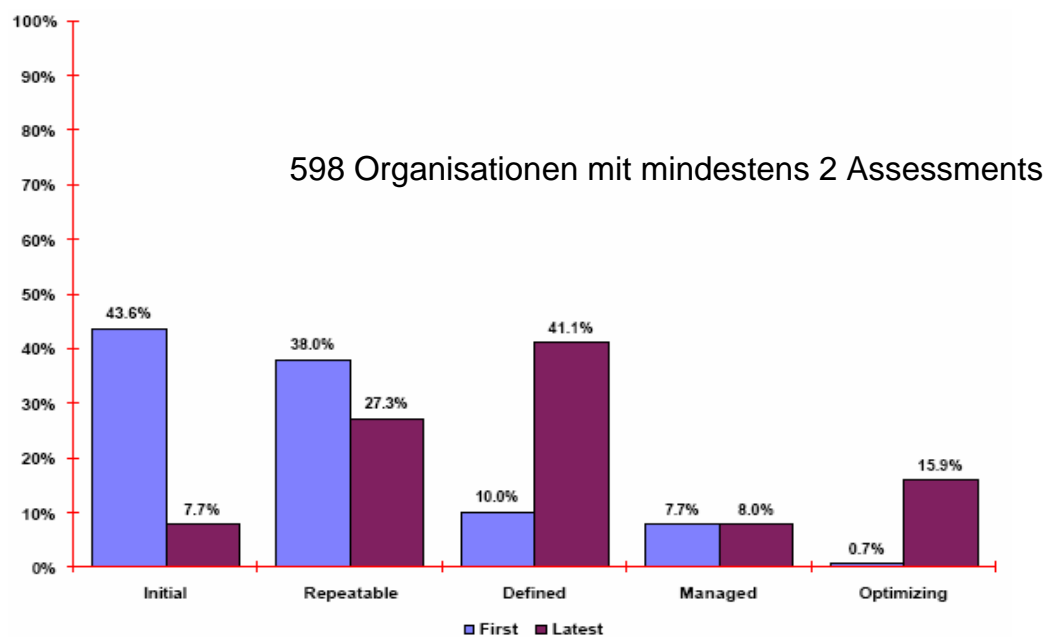
	Minimum	Maximum	Ø
Verbesserung der frühzeitigen Fehlerentdeckung (Fehler vorm Testen entdeckt)	6 %	25 %	22%
Jährliche Verkürzung der Entwicklungszeit	15%	23%	19%
Jährliche Reduktion der Fehler nach Auslieferung	10%	94%	39%
Return on Invest (in Jahren)	4	8,8	5



# Trend des Reifegrades



# Reifegrad bei erstem und letztem Assessment





## Schwer- vs leichtgewichtige Methodologien

Schwergewichtige (heavyweight) Methodologien:

- sind bürokratisch
- sind prozessorientiert
- produzieren viele Dokumente
- planen weit in die Zukunft
- planen detailliert
- sind wenig flexible auf Änderungen

Leichtgewichtige (lightweight) Methodologien:

- sind unbürokratisch
- sind personenorientiert
- produzieren kaum Dokumente
- planen nur in die unmittelbare Zukunft
- betreiben keine detaillierte Planung
- reagieren schnell auf Änderungen



## Projekt zu Software-Prozessen an der UFSC

- Etablierung von Software-Prozessen in kleinen Unternehmen.
- Untersuchung
  - Grundsätzliche Schritte
  - Dauer
  - Kosten
- Fragen
  - Werden die Prozess-Guides benutzt?
  - Wie wirken sie sich aus?
  - Wie werden sie verbessert?
- 2 Fallstudien in kleinen Unternehmen
  - VOID CAZ Sistemas Ltda., Florianópolis/SC (5 feste MA)
  - 4ILOG Tecnologia Ltda., Florianópolis/SC (18 feste MA)



- Kaum Studien wie Software-Prozesse bei KMUs etabliert sind.
- Keine Guidelines für Software-Prozess für KMUs.
- Software-Prozesse sind informell.
- Organisationsstrukturen sind primär darauf ausgerichtet das (einzige) Produkt herauszubringen, um zu überleben.
- Support und Maintenance des Produkts.
- Schlanke Organisation.
- Direkte Kommunikationswege.
- Wenige Mitarbeiter mit unterschiedlichen Rollen.
- Kunden- bzw. Markt getrieben.
- Kaum Ressourcen für Software-Prozesse bzw. Prozessverbesserung.



## ASPE-MS (Approach for Software Process Establishment in Micro and Small Companies)

- Einführung des Software-Prozesses → Prozess Engineer (externer Berater).
- Ein Mitarbeiter wird (in Teilzeit) mit eingebunden → Wissenstransfer.





## Definitionsphase: Prozessfluss-Diagramm

Pro Prozess:

- Wesentliche Aktivitäten.
- Abhängigkeiten / Reihenfolge der Aktivitäten.
- Festlegung von Entscheidungen.
- Hierarchische Verfeinerung der Aktivitäten (falls notwendig).
- Identifikation von Rollen, Kompetenzen, Verantwortung (von Aktivitäten).
- Arbeitsergebnisse der Aktivitäten.
- Festlegung von Templates für die Arbeitsergebnisse.
- Festlegung der quantitativen (messbaren) Ziele des Prozesses.
- Ziele des Prozesses selbst.
- Spätere Verfeinerung:
  - Vor- und Nachbedingungen der Aktivitäten.
  - Einzusetzende Techniken und Werkzeuge.
  - Warnung vor möglichen Problemen.



## Beschreibung einer Aktivität - Auszug

ID	CRM1.3
Activity	Technical Analysis
Purpose	Analyse the change request wrt. viability, impact and required resources/effort/duration.
Entry criteria	Request for the analysis created and send to technical support
Exit criteria	- Viability analysis completed - Effort and resources estimated ...
Input	- Template for technical analysis report (F:\...\CRM1.3_TTecAna.pdf)
Output	Report describing the result of the analysis
Methods, techniques & tools	OTRS system
Responsible	System analyst
Roles involved	System analyst, Programmer
Measures for management	Per request: number of person-hours spent on analysis Per request: time spent on analysis ...
Description & guidelines	<ul style="list-style-type: none"><li>• <b>Step 1 – Viability analysis:</b> analyse if the request can be realised technically.</li><li>• <b>Step 2 – Estimation of effort and resources:</b> estimate the effort and resources required for the realisation of the request. Effort should be estimated in person-hours and resources can include personnel, software or equipment.</li><li>• <b>Step 3 – Impact analysis:</b> analyse, if the modification will have any other impact on the system ...</li></ul>
Adaptation guidelines	If necessary, the system analyst can get in contact with the customer in order to clarify open issues.
...	



# Process Flow Diagramm - im richtigen Leben



# Dauer der Prozess-Einrichtung (ILOG)

	2003	2004												2005					
	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5 <sup>1</sup>	6
Planning																			
Phase 1																			
Phase 2																			
Phase 3																			
Phase 4																			
Monitoring & Control																			
Post-mortem																			
	Cycle 1.1												Cycle 1.2						



## Aufwandsverteilung (in Ph bei ILOG)

Phase	Process engineer	Company representatives, including process engineer assistant, process performers and director	Total
Planning	10,0	5,5	15,5
Phase 1 (revising assessment results)	35,5	30,5	66,0
Phase 2	3,0	2,0	5,0
Phase 3	20,5	56,5	77,0
Phase 4	7,0	7,5	14,5
Monitoring and Control	1,5	6,0	7,5
Post-mortem	1,0	1,0	2,0
<b>Total</b>	<b>78,5</b>	<b>109,0</b>	<b>187,5</b>



## Fachbereich Informatik an der UFSC - Bachelorstudiengänge

- 2 undergraduate Studiengänge mit je 100 Anfängern/Jahr
  - Informatik
  - Informationssysteme (Wirtschaftsinformatik)
- Dauer bis zum Bachelorabschluss: 5 Jahre
- Dauer Abschlussarbeit: 1 Jahr
- Auswahl der Studierenden (nach 11 Schuljahren): Aufnahmeprüfung pro Hochschule (Vestibular)
- Studieninhalte: vergleichbar mit Bachelor HS Mannheim



- Organisationsform: Trimester
- Aufnahme: Mindestnote bei landesweiter (Extra-) Prüfung
- Aufnahmekapazität: hängt von der Veröffentlichungsquote der Profs ab.
  - Profs, die in der Klasse 1 für Veröffentlichungen eingestuft sind, dürfen bis zu 8 Studenten im Master betreuen. In der nächst niedrigeren Stufe noch 4 und auf der 3. (und letzten Stufe) nur 2.
- Die Gesamtaufnahmekapazität im Master richtet sich also nach der Forschungsqualität (-quantität) der Profs.
- Z.Zt. sind ca. 100 Studenten im Master.
- Die Studenten sind eng an den Prof. gebunden; mit ihm werden die VL, die zu besuchen sind abgesprochen. Jeder Prof. hat also „seinen“ Plan.
- Aussage: „Unsere Studenten im Master machen oft noch mal so ähnliche VL wie im Bachelor.“ → Qualitätsproblem beim Master.



- Pro Prof. im Masterstudiengang aktuell: 0 – 8 Studenten
- Dauer Master: 2 Jahre (offiziell, inoffiziell 3 Jahre).
- Vergleichbar mit Assistententätigkeit a.d. Uni in Deutschland
- Während dieser Zeit: 24 SWS Vorlesungen + Abschlussarbeit.
- Vorlesungen werden nicht geprüft.
- Arbeit läuft während der ganzen Zeit parallel.
- Anforderungen an die Abschlussarbeit: vergleichbar mit Masterarbeit HS Mannheim aber Veröffentlichungspflicht.



- # Profs: ca. 50
- Lehrveranstaltungen in beiden Bereichen (Bachelor und Master)
- Profs werden leistungsorientiert bezahlt und eingesetzt.
  - Einordnung erfolgt jährlich.
  - Ist eine Art Verwaltungsakt
  - In Brasilia wird zentral festgelegt, wie viele Punkte es für eine Veröffentlichung gibt (es zählen Einstufung der Konferenz etc.). Damit wird die Forschungsleistung bewertet.
  - Weitere Kriterien, wie Auslandskontakte, zählen auch.
- Einstufung der Profs./ Forschungsleistung bestimmt auch das Lehrdeputat.
  - Profs. ohne Forschung machen ca. 18 SWS VL,
  - Profs. mit entsprechender Forschung 8 SWS (inkl. Übungen) – und nehmen das m.E. nicht sehr genau (manchmal halten Masterstudenten eine VL).



**Ansichten**



**Aussichten**

