

# Vorlesung "Softwaretechnik"

## **Konstruktive Qualitätssicherung (Qualitätsmgmt., Prozessmgmt.)**

Stephan Salinger  
(Foliensatz: Lutz Prechelt)

Freie Universität Berlin, Institut für Informatik  
<http://www.inf.fu-berlin.de/inst/ag-se/>

- Projekt- vs. Prozessmgmt.
- Arten von Prozessmgmt.-  
Leitlinien
- CMM-SW, CMMI
- TQM
- ISO 9000

# Wo sind wir?: Taxonomie "Die Welt der Softwaretechnik"

## Welt der Problemstellungen:

- Produkt (Komplexitätsprob.)
  - Anforderungen (Problemraum)
  - Entwurf (Lösungsraum)
- Prozess (psycho-soziale P.)
  - Kognitive Beschränkungen
  - **Mängel der Urteilskraft**
  - **Kommunikation, Koordination**
  - **Gruppendynamik**
  - Verborgene Ziele
  - **Fehler**

## Welt der Lösungsansätze:

- Technische Ansätze ("hart")
  - Abstraktion
  - Wiederverwendung
  - Automatisierung
- Methodische Ansätze ("weich")
  - Anforderungsermittlung
  - Entwurf
  - **Qualitätssicherung**
  - Projektmanagement

- Einsicht: Man macht beim Bau von SW zahlreiche Fehler
  - die häufig zu schwerwiegenden Mängeln führen
- Prinzipien:
  - **Konstruktive Qualitätssicherung**: Ergreife vorbeugende Maßnahmen, um zu *vermeiden*, dass etwas falsch gemacht wird (Qualitätsmanagement, Prozessmanagement)
  - **Analytische Qualitätssicherung**: Verwende prüfende Maßnahmen, die entstandene Mängel aufdecken
  - **Softwaretest**: dynamische Prüfung
  - **Durchsichten**: manuelle statische Prüfung

## Analytische QS:

- Dynamische Verfahren (Test)
  - Defekttest
  - Testautomatisierung
  - Benutzbarkeitstest
  - Lasttest
  - Akzeptanztest
- Statische Verfahren
  - Manuelle Verfahren
    - Durchsichten, Inspektionen
  - Automatische Verfahren
    - Modellprüfung
    - Quelltextanalyse

## Konstruktive QS:

- **Test- und Durchsichtsmgmt.**
- **Prozessmanagement**
- **Projektmanagement, Risikomanagement**

- Motto: "Vorbeugen ist besser als Heilen"
- Kann auf einzelnes Produkt hin orientiert sein:
  - **Projektmanagement**
  - Meist recht pragmatischer Ansatz
  - Siehe spätere Präsentationen
- oder auf die Verbesserung des Prozesses einer Organisation über alle Projekte hinweg:
  - **Prozessmanagement**
  - abstrakter; längerfristige Ausrichtung; Gefahr des "Verkünstelns"
- Dazwischen gibt es, quasi als Rohstoff für beides, **Prozessmodelle**
  - die allgemein das Zusammenwirken von Rollen und Tätigkeiten zu einem Softwareprozess beschreiben
  - und projektunabhängig dessen grobe Leitlinien festlegen



# Weiterer Aufbau der Vorlesung

- Prozessmanagement, Prozessmodelle u. Projektmanagement sind eng miteinander verknüpft
- Wir sprechen jetzt zunächst kurz über Prozessmanagement
  - genauer gesagt: Prozessverbesserung
- nächste Stunde dann über Prozessmodelle
- und anschließend ausführlicher über Projektmanagement
- Viele der Inhalte sind aber nicht fest nur einem dieser Bereiche zuzuordnen
  - Anmerkung: In manchen Lehrbüchern werden diese drei Themen verwirrenderweise recht weit voneinander getrennt

Aller konstruktiven Qualitätssicherung liegt folgende **Annahme** zu Grunde:

- Hohe Qualität erhält man in der Regel dann,
  - wenn man geeignete Arbeitsweisen verwendet
  - und diese sorgfältig durchführt
- Diese Forderung ist weder hinreichend
  - (denn man kann auch dann schlechte Qualität bekommen)
- noch notwendig
  - (denn man kann oft durch entsprechende Anstrengung auch mit ungünstigen Arbeitsweisen oder trotz viel Schlamperei noch gute Qualität schaffen – zumindest im Prinzip),
- beschreibt aber einen starken, typischen Zusammenhang



- Leitlinien:
  1. Gestalte den Konstruktionsprozess und sein Umfeld so, dass Qualitätsmängel seltener werden
  2. Beginne damit vor der eigentlichen Entwicklungsarbeit
    - Gestaltung von Organisation und Arbeitsumfeld
    - Auswahl von Prozessen, Technologie, Strategie
  3. Beseitige bei entdeckten Mängeln nicht nur den Mangel selbst, sondern auch seine Ursache(n) und ggf. deren Ursache(n)
    - Urgrundanalyse (root cause analysis)
  
- Vorgehensweise:
  - Vorstrukturierung des Arbeitsprozesses vorgeben
    - Produktvorgaben: Standards, Schablonen
    - Prozessvorgaben: Rollen- und Ablaufbeschreibungen
  - Laufenden Prozess überwachen
    - idealerweise quantitativ: Einhaltung der Vorgaben; Qualität

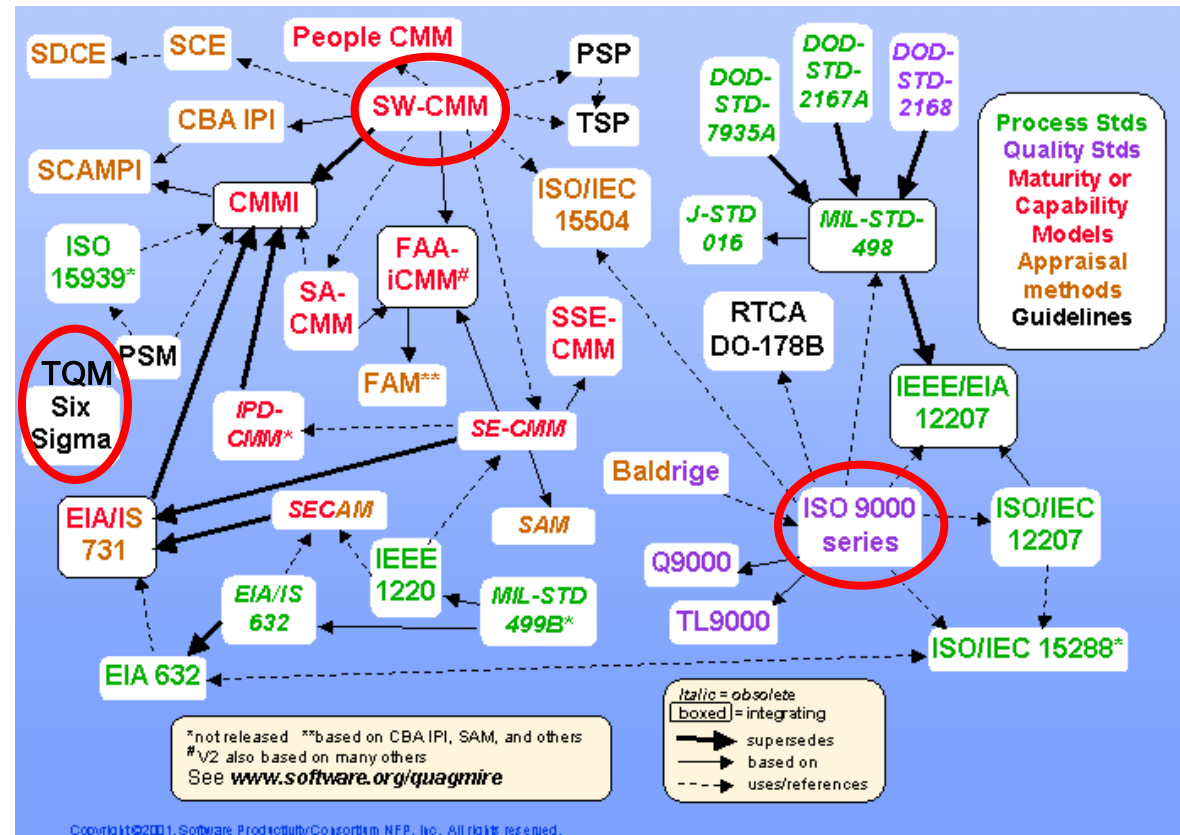




- Ziel von Prozessmanagement ist letztlich die lernende, sich selbst kontinuierlich verbessernde Organisation
- Aus der Vogelschau soll jedes Projekt wie folgt ablaufen (*Quality Improvement Paradigm, QIP*):
  1. Charakterisiere Projektumgebung (**Characterize**)
    - → Welcher Projekttyp? Welche passenden Modelle existieren?
  2. Setze quantifizierbare Ziele (**Set Goals**)
    - → Was soll erreicht werden?
  3. Wähle die geeignete Vorgehensweise aus (**Choose process**)
    - → Wie soll das Ziel erreicht werden?
  4. Führe Projekt aus (**Execute**)
    - → mit QS konstruktiv und analytisch
  5. Analysiere Projektergebnisse (**Analyze**)
    - → Was hat funktioniert? Was nicht?
  6. Ergänze Erfahrungsdatenbank (**Package**)
    - → Welche Erfahrungen/Modelle sollen wiederverwendet werden?

- Allerdings braucht eine Organisation die meisten Erfahrungen nicht selbst zu machen,
  - sondern kann allgemein bekannte Erfahrungen wiederverwenden und für sich anpassen

- Deshalb gibt es zahlreiche Leitlinien dafür, worauf ein Prozessmgmt. generell hinarbeiten sollte



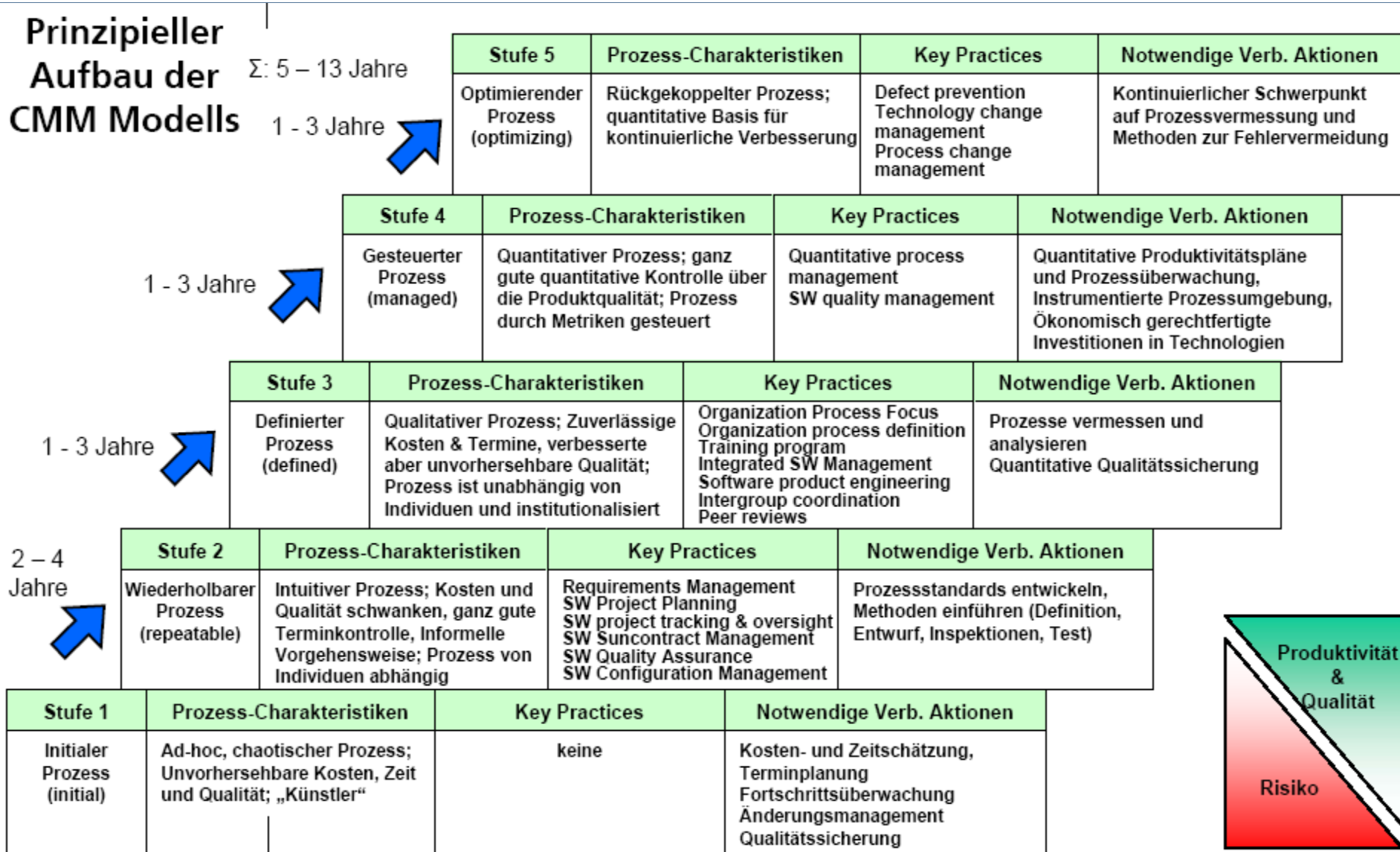
[www.software.org/quagmire](http://www.software.org/quagmire)

- **Prozessstandards**
  - Geben vor, wie gewisse Prozesse strukturiert und durchgeführt werden sollen
- **Qualitätsstandards**
  - Geben vor, wie gewisse Ergebnisse von Prozessen beschaffen sein sollen
- **Prozessreifemodelle**
  - Beschreiben Bereiche, die für einen erfolgreichen Prozess optimiert werden müssen
    - schreibt aber nicht vor, wie die resultierenden Prozesse aussehen
- Die Übergänge sind sehr fließend
- Wir sprechen konkret kurz über:
  - CMM-SW: Ein Prozessreifemodell
  - ISO 9000: Ein Prozess-Qualitätsstandard
  - TQM: Eine allgemeine Qualitätsmanagement-Methode





- CMM-SW: Capability Maturity Model for Software
  - Entwickelt in den 1980er Jahren am SW Engineering Institute (SEI) in Pittsburgh (Carnegie Mellon Univ.)
    - inzwischen **CMMI**: Capability Maturity Model Integration
  - Entwicklungsansatz: Welche Prozesseigenschaften minimieren das Fehlschlagsrisiko und maximieren die erwartete Qualität?
- Unterscheidet **5 Reifestufen**:
  - 1 Initial (initialer Prozess):
    - ad-hoc: SW-Entwicklung erfolgt "irgendwie"; Erfolg ist nicht erklärbar
  - 2 Repeatable (wiederholbarer Prozess):
    - Planung, Meilensteine, QS; Erfolg hängt aber von Personen ab
  - 3 Defined (definiertes Prozess):
    - Konstruktive QS; Institutionalisierte Prozess; Qualität meist gut
  - 4 Managed (gesteuerter Prozess):
    - Durch Messen gesteuert; vorhersagbar hohe Qualität
  - 5 Optimizing (selbstverbessernder Prozess):
    - Kontinuierliches Feedback zur Verbesserung des Prozesses



# CMM-SW Stufen, Schlüsselbereiche





Wie viel und welchen Einblick hat ein Projektmanager in den Projektstatus?

- Initial (initialer Prozess): 
  - fast keinen: Prozess ist weitgehend undurchsichtig; Erfolge und Mißerfolge sind gleichermaßen überraschend
- Repeatable (wiederholbarer Prozess): 
  - Gültiges Anforderungsdokument; Einblick punktuell über Meilensteine, dazwischen nur Bangen
- Defined (definierter Prozess): 
  - Zusätzlich: Qualitätsinformation aus Durchsichten; wohldefinierter (also verständlicher) Prozess
- Managed (gesteuerter Prozess): 
  - Kontinuierliche Einsicht durch Messungen
- Optimizing (selbstverbessernder Prozess):
  - Einblick sowohl in Projektschwächen als auch Prozessschwächen

- Angenommen, ein/e SW-Entwickler/in, der/die nur die Stufe A gewöhnt ist, arbeitet leihweise in einem Projekt der Stufe B

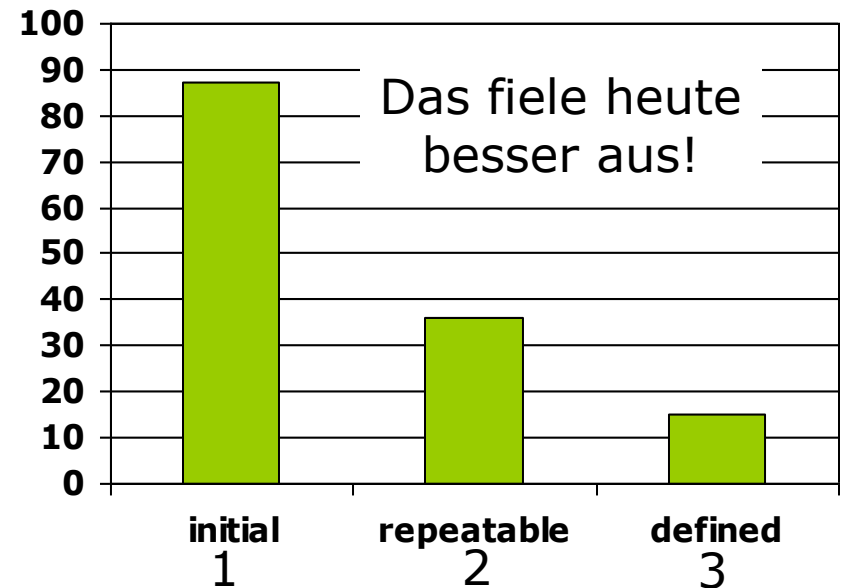
So könnte die Wahrnehmung aussehen:

- Stufe-1-Entwickler/in zu Besuch in Stufe 3: 
  - "ein Wunder"
  - unerhört diszipliniert (Durchsichten immer, auch unter Druck!), Leute so zuversichtlich und zufrieden, es gibt für alles Regeln, ganz anderes Qualitätsniveau
- Stufe-3-Entwickler/in zu Besuch in Stufe 1: 
  - "ein Chaos und Sumpf"
  - ausgenommen Versionsverwaltung macht jeder alles nach Gusto, niemand vertraut dem Plan, unter Druck bricht totale Hektik aus



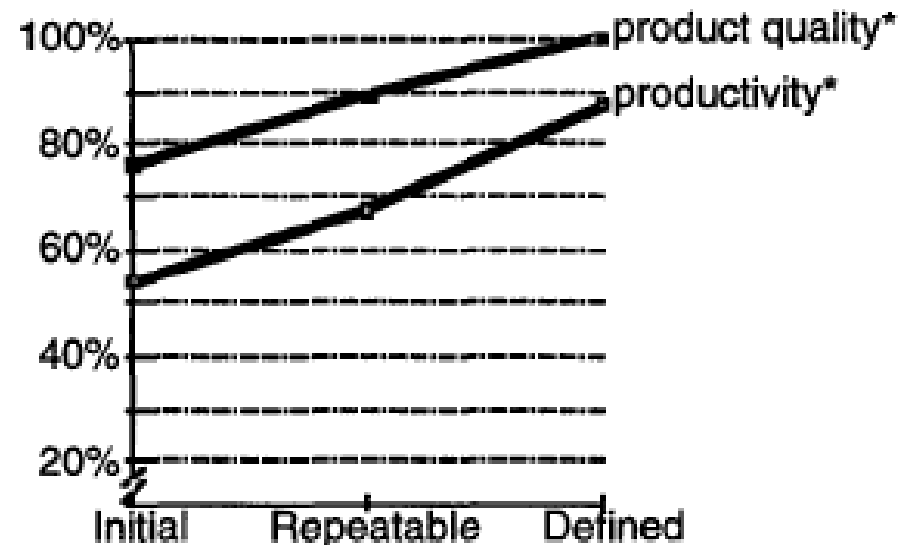
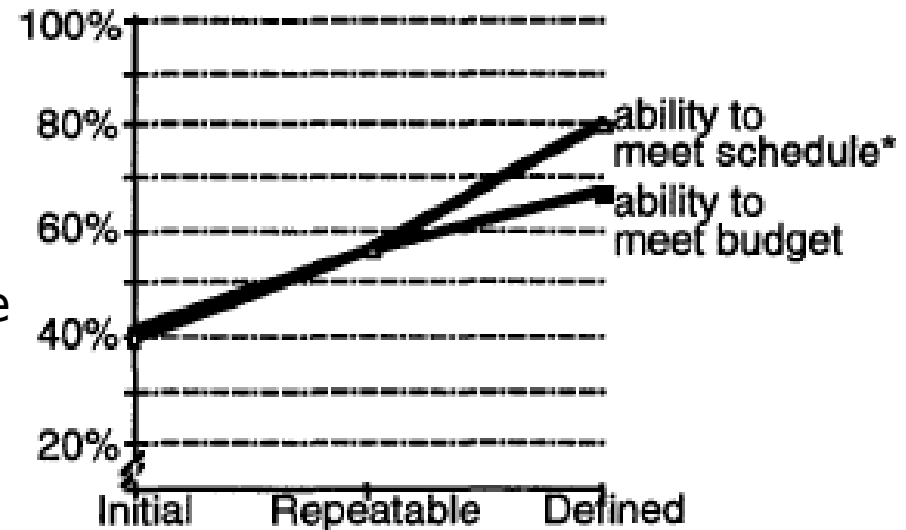
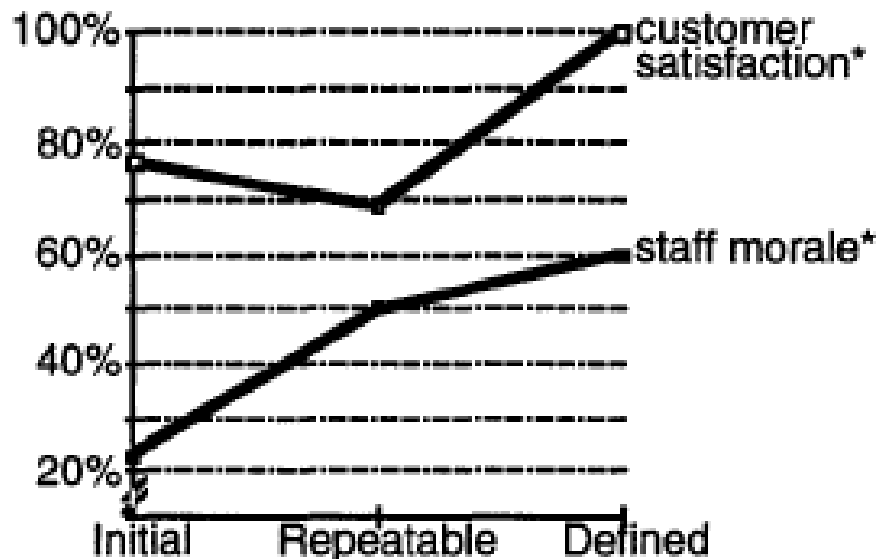
# Empirische Ergebnisse über CMM-Verwendung

- Quelle:
  - James D. Herbsleb, Dennis R. Goldenson: *"A systematic survey of CMM experience and results"*, Proceedings of the 18th international conference on software engineering, May 1996
- Form der Studie: Umfrage
  - 138 Fragebögen ausgefüllt.
  - Teilnehmer zu gleichen Teilen Seniorentwickler, Projektmanager und Mitglieder einer Prozessverbesserungs-Gruppe (SEPG)
- Häufigkeit der Reifestufen:



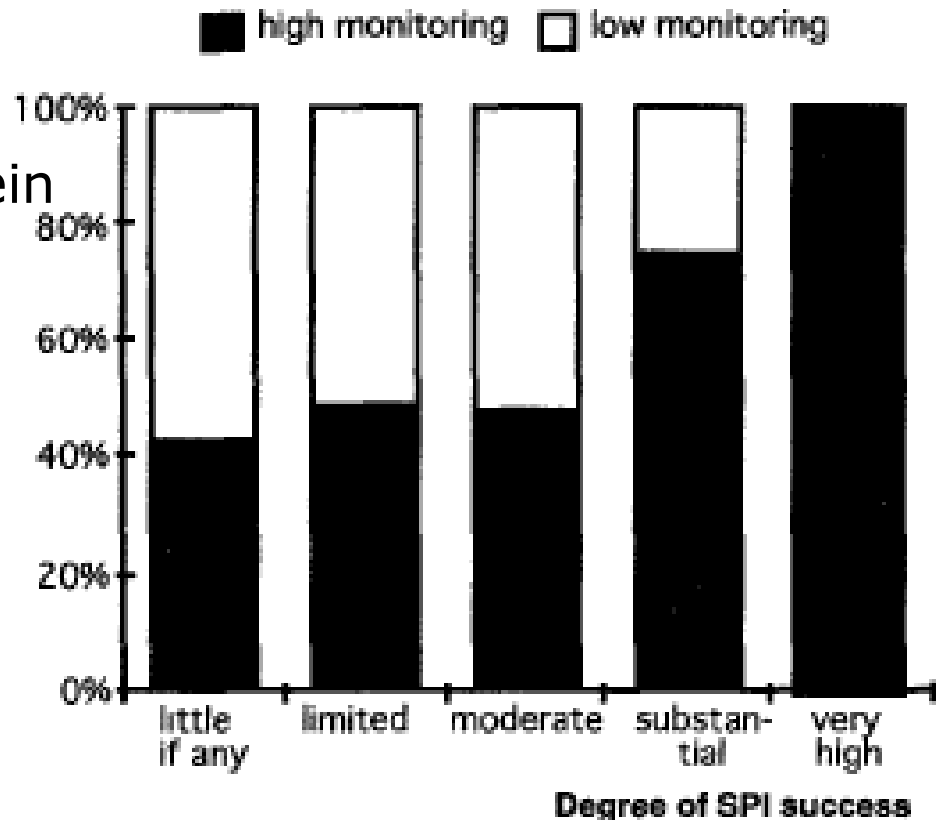
# Empirische CMM-Ergebnisse: Erfolge

- Qualitätsmerkmale "gut" oder "sehr gut"
  - im Gegensatz zu "mäßig" oder "schlecht"
  - in Abhängigkeit von CMM-Stufe
  - als Prozentsatz der Antworten



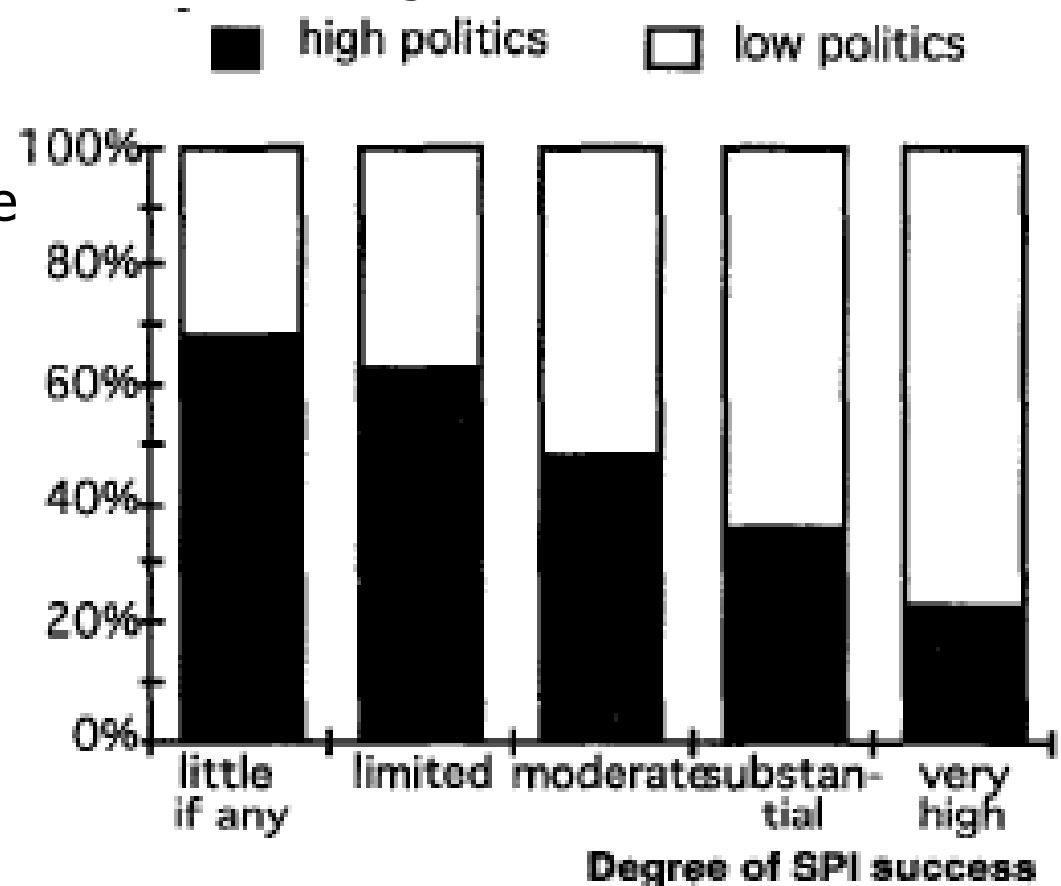
- Verfolgt das obere Management die SPI-Aktivitäten (software process improvement) sorgfältig?
  - Prozentsatz der "ja"-Antworten je nach wahrgenommener Erfolgsstufe bei der Prozessverbesserung

- Hohes Interesse des oberen Managements ist also wohl ein wichtiger Erfolgsfaktor



- Gibt es in der Firma ausgeprägte "Politik" (Machtspiele)?
  - Prozentsatz der "ja"-Antworten je nach wahrgenommener Erfolgsstufe bei der Prozessverbesserung

- Ausgeprägte Machtspiele sind also wohl ein wichtiger Misserfolgswfaktor



- Prozessverbesserung ist aufwendig
  - Es sind Investitionen nötig, die sich nicht sofort auszahlen
  - Der Aufstieg um eine CMM-Stufe dauert meist 1–3 Jahre
- Kleine Organisationen müssen vom CMM abweichen
  - und zum Teil eigene Lösungen zur Prozessverbesserung finden
  - da einige vorgesehene Prozesse und Institutionen nur für große Organisationen tragbar sind
- Hohe Reifestufen können risikoscheu machen (Tom DeMarco)
  - Bei einem sehr innovativen Projekt (viel radikales Vorgehen) sind
    - bisherige Messungen nicht mehr aussagekräftig und
    - definierte Prozesse oft nicht anwendbar
  - Deshalb fällt eine Organisation dann meist auf Stufe 2 zurück
  - Falls Reifestufen zum "Wert an sich" geworden sind, wird die Organisation tendenziell vor solchen Projekten zurückschrecken

## Anmerkung: Andere CMMs, CMMI

- CMM wurde in den 1980er Jahren hauptsächlich von einem ehemaligen IBM-Manager namens Watts Humphrey am Software Engineering Institute (SEI) in Pittsburgh entwickelt
- Als sich das Modell zu bewähren begann, entstanden andere CMMs:
  - People-CMM: CMM für Personalmanagement
  - CMM-SE: CMM für Systems Engineering (HW + SW)
- Schließlich wurden diese Modelle zusammengeführt im **CMMI** (CMM Integration)
  - Die 5 Stufen sind aber in allen diesen CMMs analog
    - allerdings heißt "repeatable" im CMMI "**managed**" und "managed" heißt im CMMI "**quantitatively managed**"



- Prozessverbesserung bedeutet immer Veränderungen
  - insbesondere, wenn man damit gerade erst beginnt
- Veränderungen erzeugen in Organisationen stets Widerstände
  - (Sie erinnern sich noch an "Anwendungssysteme"?)
- Typische "Argumente"
  - *"Das haben wir ja noch nie so gemacht"*
  - *"Das zahlt sich doch nie aus"*
  - *"Das klappt niemals"* (selbsterfüll. Prophezeiung)
- Erfolgreiche Prozessverbesserung benötigt als Grundlage stets ein Qualitätsbewusstsein
  - und zwar auf allen Ebenen: Mitarbeiter, mittleres Management, Topmanagement



© Kruemi

# Empfehlung

- Suchen Sie immer zunächst nach möglichst einfachen Lösungen
- Andere Sicht: Viele kleine Schritte führen auch zum Ziel
  - Dies ist das Prinzip von TQM:





# Qualitätsbewusstsein: Total Quality Management (TQM)

- Definition TQM (laut ISO 8402):
  - "Auf der Mitwirkung aller ihrer Mitglieder basierende Führungsmethode einer Organisation, die Qualität in den Mittelpunkt stellt und durch Zufriedenheit der Kunden auf langfristigen Geschäftserfolg sowie auf Nutzen für die Mitglieder der Organisation und für die Gesellschaft zielt."
- Entwickelt in Japan in den 1980er Jahren für Fabrikation
- Integriert Interessen von Kunden, Mitarbeitern, Unternehmen und Lieferanten
- Ziel ist Qualität aus Kundensicht

- Primat der Qualität:
  - Alle Arbeit soll sofort und immer 100% richtig getan werden
  - wo immer nötig werden dafür Prozesse verbessert
- Zuständigkeit aller:
  - Jeder Mitarbeiter ist voll für die Endqualität mit verantwortlich
- Ständige Verbesserung (Kaizen):
  - Viele kleine Schritte führen allmählich zu stets guter Qualität
- Internes Kunde/Lieferanten-Verhältnis:
  - Tut Mitarbeiter A etwas, das Mitarbeiter B benutzt, so ist A ein Lieferant und B sein Kunde
  - A soll für hohe Kundenzufriedenheit bei B sorgen
  - Der Erfolg jedes Teams wird daran gemessen

改 = KAI = CHANGE

善 = ZEN = GOOD  
(FOR THE BETTER)

改善 = KAIZEN  
= CONTINUAL  
IMPROVEMENT

- TQM wurde entwickelt im Bereich der Güterfabrikation
  - z.B. Automobile
- Es eignet sich dort sehr gut, um ein Qualitätsbewusstsein zu schaffen und zu erhalten
- Gelangt bei SW-Herstellung nicht ganz zu voller Blüte, weil es dort so schwierig ist, niedrigste Mängelhäufigkeiten zu erreichen
  - Aber dennoch ein lohnenswerter Ansatz

- Normen für zertifiziertes Qualitätsmanagement
  - Für SW relevant: ISO 9001 und ISO 9000-3
  - Beschreibt Mindestanforderungen an ein Qualitätsmanagement
    - Hauptsächlich Dokumentations- und Prüfpflichten
  - Grobe Norm: Nur einige Dutzend Seiten
    - CMM-SW hat 600 Seiten!
  - Zertifikat wird von Auditor vergeben und ist 3 Jahre gültig
- ISO 9000-3: Inhaltsrichtlinien für diverse Dokumentarten
  - Vertrag zwischen Auftraggeber und Lieferant
  - Spezifikation
  - Qualitätssicherungsplan
  - Wartungsplan
  - Konfigurationsmanagementplan



- Ein voller Einstieg ist stets recht aufwendig (siehe CMM-SW)
- Außerdem gibt es viele Leute, die davon ohnehin nichts halten
- Vermutlich sind die **TQM-Prinzipien** der beste Ansatz für den Start:
  1. Qualität (Kundenzufriedenheit) ist wichtig:  
Dem kann niemand widersprechen
  2. Nur kleine Schritte sind verlangt: Machbar!
  3. Alle machen mit: Das ist der wichtigste Trick
- Größtes Problem:
  - Die Bedingungen der Kundenzufriedenheit sind bei SW oft sehr schwierig zu begreifen

Ist Prozessmgmt. nur etwas für große SW-Organisationen?

**Nein!**

- Die meisten Elemente von CMM taugen auch für kleine Org.
  - Manche der Vorgaben aus dem CMM sind allerdings für kleine Organisationen irrelevant oder unpraktikabel
- ISO 9000 taugt auch für kleine Org.
- TQM taugt auch für kleine Org.
  
- Grundsätzlich: Kleine Organisationen sind fast genau so sehr auf gutes Prozessmgmt. angewiesen wie große, wenn sie hohe Qualität und Effizienz erzielen wollen

- Im Entwicklungsalltag gilt vor allem die Regel  
**Do the right thing!**
- Viele Leute verhalten sich aber eher so:



- ...und manche sogar so:





## Analytische QS:

- **Dynamische Verfahren (Test)**
  - **Defekttest**
    - **Wie wählt man Zustände und Eingaben aus?**
    - **Wer wählt Zustände und Eingaben aus?**
    - **Wie wählt man Testgegenstände aus?**
    - **Wie ermittelt man das erwartete Verhalten?**
    - **Wann wiederholt man Tests?**
    - Wann/wie kann und sollte man Tests automatisieren?
  - ...
- **Statische Verfahren**
  - ...

## Konstruktive QS:

- Test- und Durchsichtsmgmt.
- Prozessmanagement
- Prozessmodelle, Projektmanagement, Risikomanagement

### Analytische QS:

- Dynamische Verfahren (Test)
  - Defekttest
    - **Testautomatisierung**
  - **Benutzbarkeitstest**
  - **Lasttest**
  - **Akzeptanztest**
- **Statische Verfahren**
  - **Manuelle Verfahren**
    - **Durchsichten, Inspektionen**
  - **Automatische Verfahren**
    - **Modellprüfung**
    - **Quelltextanalyse**

### Konstruktive QS:

- Test- und Durchsichtsmgmt.
- Prozessmanagement
- Prozessmodelle, Projektmanagement, Risikomanagement

## Analytische QS:

- Dynamische Verfahren (Test)
  - Defekttest
  - Benutzbarkeitstest
  - Lasttest
  - Akzeptanztest
- Statische Verfahren
  - Manuelle Verfahren
    - Durchsichten, Inspektionen
  - Automatische Verfahren
    - Modellprüfung
    - Quelltextanalyse

## Konstruktive QS:

- Test- und Durchsichtsmgmt.
- **Prozessmanagement**
  - **CMM-SW**
  - **TQM**
  - **ISO 9000**
- Prozessmodelle, Projektmanagement, Risikomanagement

ab nächste Stunde!



- In vielen Fällen sind wichtige Qualitätsmerkmale auch
  - **Sicherheit (safety)**      Schutz der Umwelt vor dem System
  - **Schutz (security)**      Schutz des Systems vor der Umwelt
- Diese haben wir hier nicht einzeln berücksichtigt, weil das unseren Rahmen sprengen würde
  - Sie erfordern aber eigentlich nochmal ganz separate Überlegungen und Maßnahmen

## Weitere Quellen

- Chris Sauer, Ross Jeffery, Lesley Land, Philip Yetton: *"The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research"*, IEEE Transactions on Software Engineering 26(1), Januar 2000
  - Überblick über Stand der Forschung zu Durchsichten
  - Zugriff v. Uninetz über [www.computer.org](http://www.computer.org) (IEEE Comp. Society)
- <http://www.stickyminds.com>
  - QS-Portal. Umfangreiche Informationen über Qualitätssicherung, Anforderungsbestimmung u.a.
- <http://www.software-kompetenz.de>
  - SWT-Portal. Allgemeine Informationen zu SWT, insbesondere hilfreich zu Inspektionen/Durchsichten
- <http://www.kaner.com>
  - Cem Kaner. Diverse Artikel über Testautomatisierung. Weitere Verweise.

**Danke!**