



Prozessverbesserung

Sven Holzmann

Jochen Schwarz



Inhalt

CMM(I)

Einführung
Reifegrade
Statistische Daten
Studie

PSP

Einführung
Prozessebenen
Studie

ISO 15504

Einführung
ISO 12207
SPICE
Studie

Resümee



Historie

- 1986 SEI beginnt mit der Entwicklung eines Systems zur Bewertung von Softwareprozessen
- 1991 CMM 1.0
- 1993 überarbeitetes CMM 1.1
- 1997 CMM 2.0 kurz vor der Verabschiedung zurückgezogen
statt dessen CMMI – Projekt gestartet
- 2000 CMMI Pilotversion 1.0
- 2002 CMMI 1.1 freigegeben
- 2003 Unterstützung für CMM ausgelaufen
- 2005 CMMI 1.2 angekündigt
Lizenzen der Assessmentleiter für CMM laufen aus
- 2006 Herbst: Erscheinungstermin CMMI 1.2
- 2007 CMMI 1.1 läuft aus

Das Software Engineering Institute (SEI) der Carnegie Mellon University/Pittsburgh untersteht dem US-Verteidigungsministeriums (Department of Defense, DoD). Die Entwicklung des Bewertungssystems begann durch eine Initiative des DoD.

In der Pilotversion 1.0 steht das I für „Integrated“. Später wurde dies durch Integration ersetzt.

Da die Lizenzen der Assessmentleiter ausgelaufen sind gibt es nun keine offiziellen CMM – Assessments mehr.



CMM

Framework zur Prozessverbesserung

Abstrakte Formulierung

Konkrete Implementierung der Firma überlassen

Stufenmodell

Bewertung von Organisationen durch ausgebildete Assessoren

Unterscheidet in ausgereifte und unausgereifte Organisationen



Unausgereifte Organisationen

Können ausgezeichnete Ergebnisse erzielen

Durch „heldenhafte“ Taten von Individuen

Reproduzierbarer Erfolg wenn „die gleichen Individuen das gleiche tun“

Qualitätsmessung nicht vorhanden

Qualität schwer vorauszusagen

Reagieren auf Gegebenheiten

Aktivitäten zur Qualitätsverbesserung fallen weg



Ausgereifte Organisationen

Reproduzierbarer Erfolg

Organisationsweiter, wiederholbarer, standardisierter Softwareprozess

Wird zu Angestellten und neuen Mitarbeitern kommuniziert

Präskriptiv = deskriptiv

Wird verbessert

Verbesserungen werden getestet

Objektive Qualitätsmessung

Problemanalyse

Produkteckdaten können vorhergesagt werden

Vorhersagen treffen ein

Präskriptiv: Beschreibt den vorgegeben Prozess

Deskriptiv: Beschreibt den tatsächlich durchgeführten Prozess

Produkteckdaten: Kosten, Zeit, Funktionalität, Qualität



Konzepte

Prozessleistungsfähigkeit

Zu erwartende Ergebnisse, die durch den Prozess erreicht werden *können*

Prozessleistung

Ergebnisse, die durch Prozess erreicht werden

Ungleich Leistungsfähigkeit, z.B. durch Umgebung begrenzt

Prozessreife

Ausmaß, zu dem Prozess definiert, geleitet, kontrolliert und effektiv ist

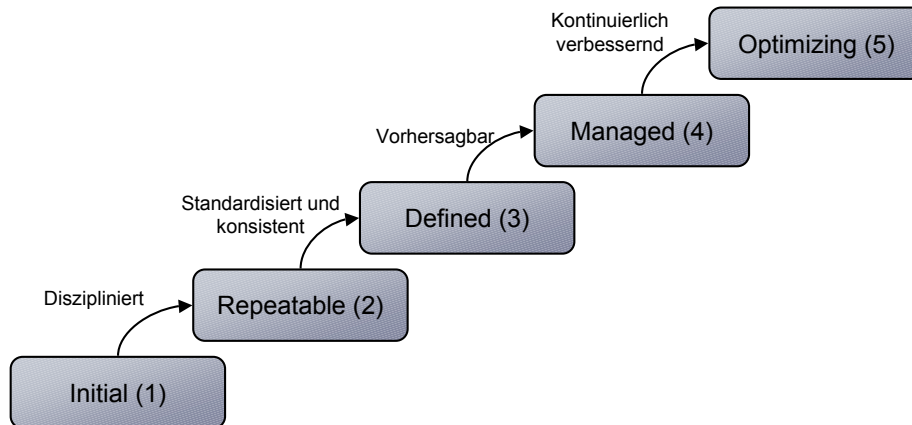
Impliziert Verbesserungspotential der Leistungsfähigkeit

Wird durch CMM gemessen

Die Messung der Reife erfolgt in sog. Assessments. Zur Messung wird ein System aus 5 evolutionären Stufen verwendet.



Reifelevel/Reifegrad



Repeatable: wiederholbar

Defined: definiert

Managed: geleitet

Optimizing: verbessernd



Initial

Prozess ist ad hoc, bisweilen chaotisch

Wenige/keine Prozesse definiert

Erfolg durch persönliche Anstrengung

Eckdaten nicht vorhersehbar

Keine stabile Entwicklungsumgebung

Gute Engineering – Praktiken haben unbestimmte Ergebnisse

Unbestimmte Planung und reaktionsgetrieben

Krisen: Code & Fix

Erfahrene Teamleiter widerstehen der Versuchung

Erfahrung geht mit dem Teamleiter



Repeatable

Projekt Management Prozesse eingeführt

Kosten, Zeitplan und Funktionalität werden verfolgt
Probleme der Einhaltung werden erkannt

Projektplanung und -leitung basieren auf Erkenntnissen

Effektiver Prozess

erfahren, dokumentiert, erzwungen, trainiert, gemessen, verbesserungsfähig

Standards für Prozesse eingeführt

Einhaltung wird kontrolliert

Instabil

Ein Softwareprozess des Reifegrades 2 wird als instabil beschrieben. Es wird sich zeigen, ob die Verbesserungen gegenüber des Grades 1 nur aufgesetzt sind oder ob sie tatsächlich verinnerlicht wurden. Je nachdem wird der Prozess mittelfristig auf Grad 3 aufsteigen oder auf Grad 1 zurückfallen.

Der Reifegrad 2 ist der einzige der 5 Reifegrade, auf den diese Beschreibung zutrifft.



Defined

Standardprozess

Organisationsweit, standardisiert und dokumentiert
Beinhaltet Prozesse für Management und Engineering
Effektive SE – Praktiken berücksichtigt
Einsicht des Managements in den technischen Fortschritt der Projekte

Zugeschnittener Prozess

Pro Projekt

Software Engineering Process Group (SEPG)

Kümmert sich um Prozessaktivitäten der Organisation

Trainingsprogramm

Personal und Management haben das Wissen und die Fähigkeit für ihre Rollen

SE: Softwareengineering



Managed

Organisationsweites Messprogramm

Messergebnisse von Softwareprozess und Produktqualität vorhanden
Prozesse und Produkte sind quantitativ verstanden und kontrolliert
Quantitative Ziele für Prozesse und Produkte

Datenbank

Enthält Messergebnisse zu spezialisierten Prozessen



Optimizing

Beständige Prozessverbesserung

Konzentration der Organisation auf Verbesserung

Prozessbewertung

Feedback nach Prozess – Veränderung

Innovative Ideen und Techniken in Pilottests

Organisation erkennt Stärken und Schwächen von Prozessen

Prozessanalyse

Effektivitätsdaten zur Kosten/Nutzen Rechnung von neuen Technologien

Erfolgreiche Innovationen werden in der Organisation verbreitet



Levelverteilung

Nur sehr wenige Organisationen auf Level 4 und 5

Charakteristiken schwer feststellbar

Analogie zu anderen Industriezweigen

Sobald mehr Daten vorhanden:

Anpassung der Definition von Level 4 und 5



Schlüsselgebiete

Auch Schlüsselprozesse

Bestimmen, auf was sich die Organisation konzentrieren soll

Für jeden Level gibt es Schlüsselgebiete

Werden alle abgedeckt, so wird der Level erreicht

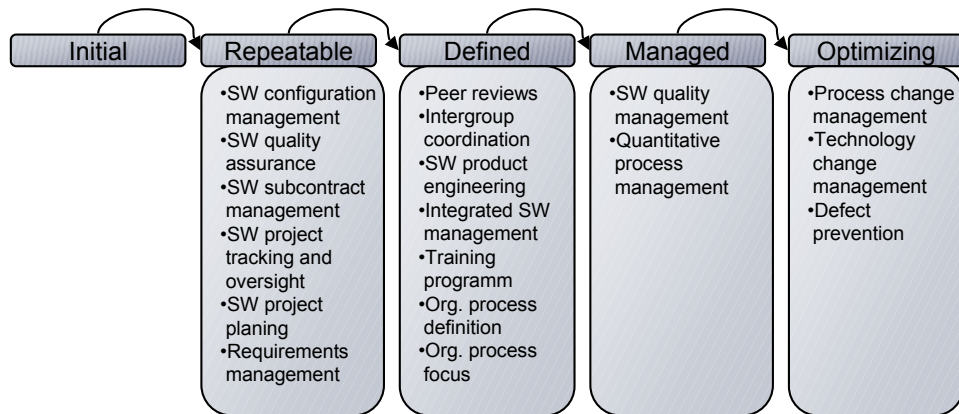
Gebiete definieren Ziele

Wenn die Ziele erreicht sind, ist das Gebiet abgedeckt

Ziele werden am besten durch Schlüsselpraktiken erreicht



Reifelevel/Reifegrad



Repeatable: wiederholbar

Defined: definiert

Managed: geleitet

Optimizing: verbessernd



Beispiel Schlüsselgebiet

Software Konfigurationsmanagement

Vorraussetzung für Level 2 (Repeatable)

Ziele

Software Konfigurationsmanagement Aktivitäten sind geplant

Ausgewählte SW – Arbeitsprodukte sind identifiziert, kontrolliert und verfügbar

Änderungen an identifizierten SW – Arbeitsprodukten sind kontrolliert

Betroffene Gruppen und Individuen sind informiert über den Status und

Inhalt der SW - Grundlinie



Statistische Daten zu CMM

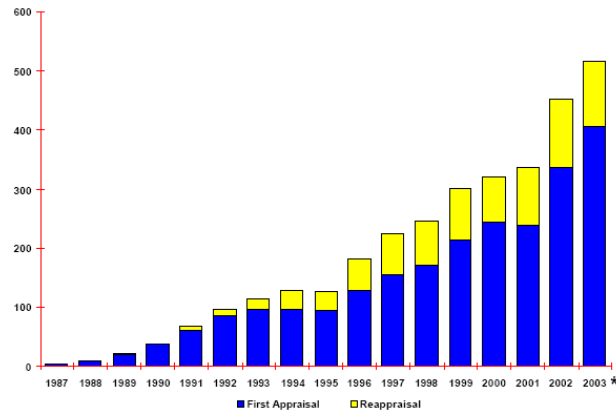
Process Maturity Profile – Software CMM 2003 Year End Update

Jährliche Studie vom SEI

<http://www.unf.edu/~ncoulter/cen6070/handouts/2004marSwCMM.pdf>

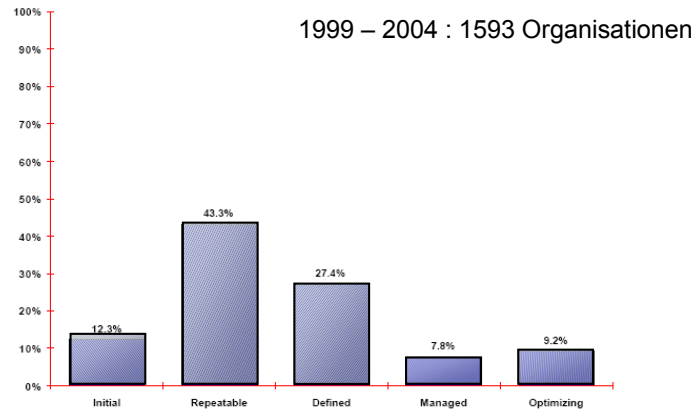


Durchgeführte Assessments



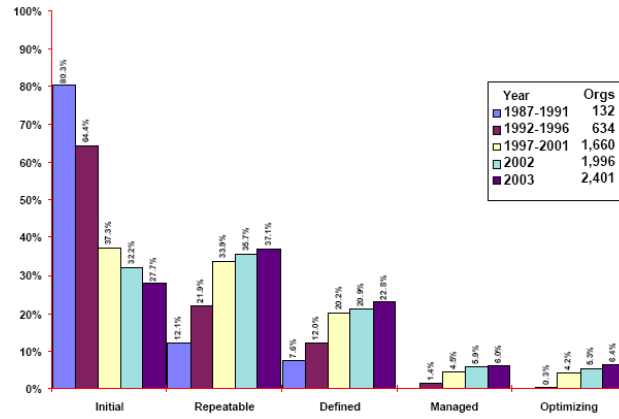


Verteilung der Reifegrade



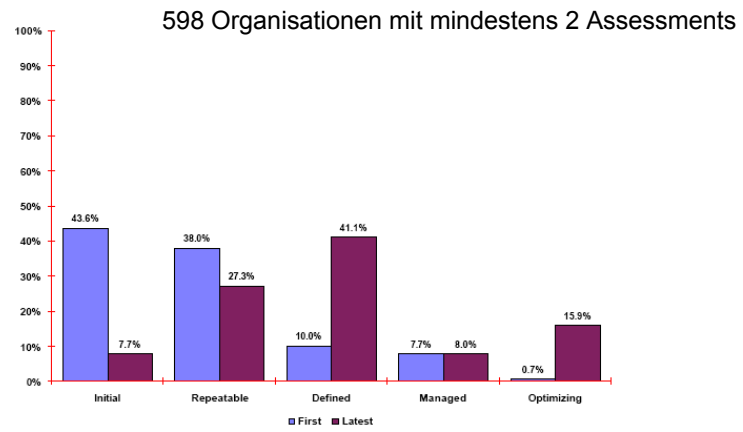


Trend des Reifegrades





Reifegrad bei erstem und letztem Assessment





Von CMM zu CMMI

CMM

Eigentlich: CMM – SW

Modell zur Verbesserung von Prozessen

Daneben: CMM – SE, CMM – People, CMM – Team, etc.

Stufenmodell



Von CMM zu CMMI

CMMI

Integration: Vereinheitlicht die einzelnen CMM's
Verwendung der Erfahrungen mit CMM
Sehr abstrakte Punkte wurden (etwas) konkretisiert

Modell 1: Stufenmodell

Ähnlich wie bei CMM
Neue Levelbezeichnungen: Managed (2) und Quantitatively Managed (4)
Neue Schlüsselgebiete, Aufteilung von Gebieten, Entfernung von SW vor Gebietsbezeichnungen

Modell 2: kontinuierliches Modell

Vergleich zu SPICE
Prozessbereiche werden von 0 bis 5 bewertet

CMMI verwendet 2 verschiedene Bewertungsmodelle. Das Stufenmodell erlaubt den Vergleich mit CMM, das kontinuierliche Modell ermöglicht den Vergleich mit SPICE.



Beschreibung

Umfrage bei mehreren Firmen aus USA und Kanada

Assessments durchgeführt 1992 und 1993

Assessments liegen zwischen 1 und 3 Jahren zurück

Zusammenstellung der Teilnehmer September 1994

155 Firmen aus PAIS

61 Firmen kontaktiert (40%)

1/3 der Firmen mit mehr als 200 Mitarbeitern

1/3 der Firmen mit weniger als 70 Mitarbeitern

PAIS: Process Appraisal Information System

Datenbank des SEI, die Daten über bisher durchgeführte Assessments enthält.



Teilnehmer

Befragt wurden

Projektmanager

Senior developer mit der meisten Erfahrung

SEPG Manager auf Organisationsebene (falls vorhanden)

(Senior Manager = Sponsor des Assessments)

167 Personen für 61 Assessments

138 Fragebogen vollständig ausgefüllt (85%)

Gehören zu 56 Assessments (92%)

SEPG: Software Engineering Process Group

Senior Manager: Wurden in dieser Studie nicht befragt, da die Fragen eher technischer Natur waren.

Alle Teilnehmer sollten mit dem durchgeführten Assessment vertraut sein.



Ziele

Was passiert typischerweise nach einem CMM Assessment

Bezogen auf Prozessverbesserungen

Verstehen, warum manche Verbesserungsanstrengungen besser sind als andere

Den Zusammenhang zwischen Reifegrad und Performance der Organisation erkennen

Interessanteste Ziel



Fragebogen

Fragen so zusammengestellt, dass Sichten bedeutungslos werden

Entwickler sehen Dinge anders als Manager

75 Fragen in 4 Kategorien

Die Kategorien und ihre Aufteilung waren wie folgt:

- Das Assessment: 13 Fragen
- Prozessverbesserung: 39 Fragen
- Die Organisation: 18 Fragen
- Persönlicher Hintergrund: 5 Fragen

Der komplette Fragebogen findet sich im Anhang von [2].



Ergebnisse

Kaum Unterschiede zwischen verschiedenen Rollen

Guter Fragebogen

Prozessverbesserung zahlt sich durch bessere Organisationsperformance aus

Wahrscheinlichkeit zur guten Einschätzung von Performance steigt

Kein erheblicher Einfluss von

Organisationsbereich (Regierung/Verteidigung und andere)

Organisationsgröße

Prozessverbesserung zahlt sich durch bessere Performance der Organisation aus: Diese Annahme wurde nach Auswertung von 2 Fragen getroffen. Der Teilnehmer wurde nach einer Einschätzung des aktuellen CMM Levels gefragt. Danach wurde die Einschätzung verschiedener Performanceindikatoren erfragt (Zeiteinhaltung, Einhaltung von Budget, Produktqualität, Mitarbeiterproduktivität, Kundenzufriedenheit, Mitarbeitermoral). Bei Firmen mit höherer Levelschätzung wurden die Fragen nach der Performance häufiger mit gut oder exzellent beantwortet.

In Organisationen, die nicht für Regierung oder Verteidigungsministerium arbeiten, ist CMM noch nicht so lange eingeführt.



Positive Reaktionen

Assessments sind sehr genau und nützlich um nachfolgende Anstrengungen zur Prozessverbesserung zu leiten

Anstrengungen werden größtenteils durch CMM Einschätzung bestimmt
Nur 10% denken, dass aufgrund von CMM wichtige Verbesserungen vernachlässigt wurden

Bewertung durch CMM zeigt Stärken und Schwächen der Organisation auf

Manager können aktiv den Fortschritt der Prozessverbesserung aufzeigen

Können deren Ziele benennen
Passende Ressourcen auswählen



Negative Reaktionen

¼: Es hat sich nicht viel getan

¼: Empfehlungen zu ehrgeizig

In angemessener Zeitspanne kaum zu erreichen

Diese Firmen führten weniger Verbesserungen durch

40%: Krisen führten zu Vernachlässigung von Verbesserungen

Fast ¾: Verbesserung leider an Mangel von Zeit und Ressourcen

Über ¾: Verbesserungen brauchen länger als erwartet

Über 2/3: Verbesserungen kosten mehr als erwartet



Bewertung

Keine Kontrollgruppe

Nur Firmen aus CMM Datenbank befragt

Nur 7 Fragen zur Performance

Eigene Einschätzung zur Reife wird bewertet

Ausgereift \leftrightarrow Unausgereift vs. Optimistisch \leftrightarrow Pessimistisch

Alter der Studie

Zeigt: CMM führt zu Verbesserungsanstrengungen

Validierung \leftrightarrow Verifikation

Vom SEI selbst durchgeführt

Die Interpretation der Interviewer lautet, dass ausgereifte Firmen eine bessere Performance melden. Eine andere Interpretation wäre, dass Optimistische Personen bessere Performance berichten. Diese würden auch eine höhere Einschätzung der Reife liefern.

Die Studie wurde 1995 durchgeführt. Wie man den statistischen Daten entnehmen kann, hat sich die Verteilung der Reifegrade seither Grundlegend verschoben.

Die Studie verifiziert CMM. Das Assessment scheint sich tatsächlich dazu zu eignen, Prozessverbesserungen (im Sinne von CMM) anzustoßen. Ignoriert wird die Validierung, d.h. wie ist die Korrelation Reifegrad/Performance.



PSP – The Personal Software Process

Idee von Watts Humphrey (1995):

Anwendung der Prozessverbesserung auf den einzelnen Programmierer
→ Prozessmanagement und Kontrolle auf individueller Ebene

Eigenschaften

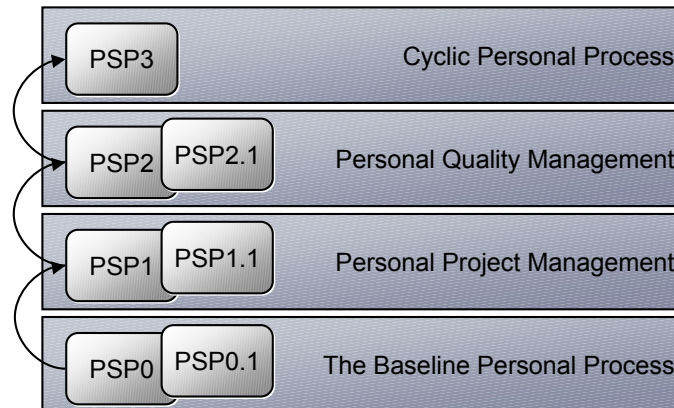
- Definierter Prozess
- Planen
- Messen
- Überwachen

Idee: Das Prinzip der Prozessverbesserung ist gut, da Die Software letzten Endes aber von einzelnen Personen erstellt wird, ist es wichtig, dass auch diese Personen Einen „guten“ Prozess haben.

Eigentlich für die Planung und Entwicklung von Softwaremodulen und kleineren Programmen gedacht, kann aber auch auf andere individuelle Aufgaben angepasst werden.



PSP – Prozessebenen



Die einzelnen Prozessebenen bauen aufeinander auf und werden dementsprechend auch in einem Kurs gelehrt.

Ähnlich wie bei CMM/ISO 15504 wird die Vorgehensweise zunehmend strukturierter



PSP – The Baseline Personal Process

PSP0

Step	Phase	Description
1	Plan	Planen und Dokumentieren
2	Design	Entwerfen des Programms
3	Code	Implementieren des Designs
4	Compile	Kompilieren, Fehler entfernen und protokollieren
5	Test	Testen, Fehler entfernen und protokollieren
6	Postmortem	Zeit, Fehler und Größenangaben protokollieren

PSP0.1

Prozessverbesserungsvorschläge



PSP – Personal Project Management

PSP 1

Größen- und Auwandsabschätzung
→ PROBE (**Pro**xy-**B**ased **E**stimating)

PSP 1.1

Planung und Überwachung
→ Earned Value Methode

Bei PROBE wird ein „Proxy“ geschätzt (z.B. ein Objekt):

- Kategorisierung der Objekte
- Bestimmung der relativen Größe (very small, small, medium, large, very large)
- Konvertierung in LOC anhand von Daten die in vorigen Projekten gesammelt wurden
- Aufsummierung der Größen über alle Objekte
- Vorhersage mittels linearer Regression

Earned Value:

- Zuweisung eines Wertes (value) zu jeder Aufgabe (nach Anteil am Gesamtaufwand)
- Wird eine Aufgabe erledigt, wird ihr Wert ein earned value
- Darüberlässt sich der momentane Stand des Projektes überprüfen



PSP – Personal Quality Management

PSP2

- Design Reviews
- Code Reviews
- Maße für die Prozess- und Produktqualität

PSP2.1

- Design Notation
- 4 Designvorlagen
- Techniken zur Verifizierung des Designs

Ziel ist, alle Fehler schon vor dem ersten Kompilieren zu finden

Yield: Prozentualer Anteil der vor dem Kompilieren gefundenen Fehler an den gesamten Fehlern

Design/Code Review: selbstständig durchgeführt, strukturiert, datengetrieben, mit Checklisten (aus persönlicher „Fehler-Geschichte“)

Design: Abschätzen der verursachten/beseitigten Fehler

Implementierung: Vergleich der verursachten/beseitigten Fehler mit den geplanten Werten. → 60%-70% Yield können erreicht werden

PSP2.1

Sichten (Spezifikationen):

- Abläufe
- Funktionen
- Zustände
- Logik



PSP – Cyclic Personal Process

PSP3

- High-Level Design
- High-Level Design Review
- Planung von Zyklen
- Zyklen in der Entwicklung

PSP3:

Die Produktivität eines Entwicklers ist am höchsten bei Programmen/Modulen deren Größe in einem bestimmten Bereich liegt.

- Darüber leidet die Qualität (der Prozess skaliert dann nicht mehr gut)
- Darunter leidet die Produktivität (wegen festen Kosten, die durch Overhead verursacht werden)

Das Gesamtprogramm wird in PSP2.1-Zyklen aufgeteilt, die integriert und getestet werden



Auswirkungen

- Bessere Aufwandsabschätzung
- Leichteres Planen und Überwachen der eigenen Arbeit
- Schutz vor "Überverantwortung"
- Persönlicher Einsatz für Qualität
- Einbindung in Prozessverbesserung

Aufwandsabschätzung: Besser durch Vergleich mit vorherigen Projekten

Planen & Überwachen: Gegenwärtiger Stand kann jederzeit mit den gesteckten Zielen verglichen werden

"Überverantwortung": Programmierer weiss, wie viel er leisten kann → wird mehr verlangt, kann er begründet und fundiert ablehnen

Einsatz für Qualität: Qualität ist nichts Abstraktes

Prozessverbesserung: Programmierer hat nicht das Gefühl, dass "von oben" etwas diktiert wird → bessere Resultate



Studie

Daten von 298 Teilnehmern an 23 PSP-Schulungen

Daten nicht immer vollständig, für jede der Analysen standen jedoch mindestens 170 Fälle zur Verfügung.

Auswertung der Daten durch Varianzanalyse (ANOVA)

Die Studie wurde 1997 von Will Hayes und James W. Over durchgeführt (angestellt am Software Engineering Institute, Carnegie Mellon University), also



Bereiche der Untersuchung

Schätzung der Programmgröße

Schätzung des Arbeitsaufwandes

Fehler

- Fehlerdichte
- Yield

Produktivität



Datensätze bezogen auf Bereiche der Untersuchung

Assignment	1	2	3	4	5	6	7	8	9	10
Effort	277	276	271	262	248	239	226	219	209	152
Size	265	274	271	262	248	239	226	219	209	152
Defects	277	273	267	259	244	235	224	214	202	150



Nicht von allen Gruppen durchgeführt

Quelle [10]

Die für die Bereiche der Untersuchung zur Verfügung stehenden Datensätze unterscheiden sich, da nicht von allen Testpersonen 100% der Daten erstellt wurden.

Aufgabe 10 wurde von vielen Gruppen gar nicht durchgeführt



Datenbasis je Phase

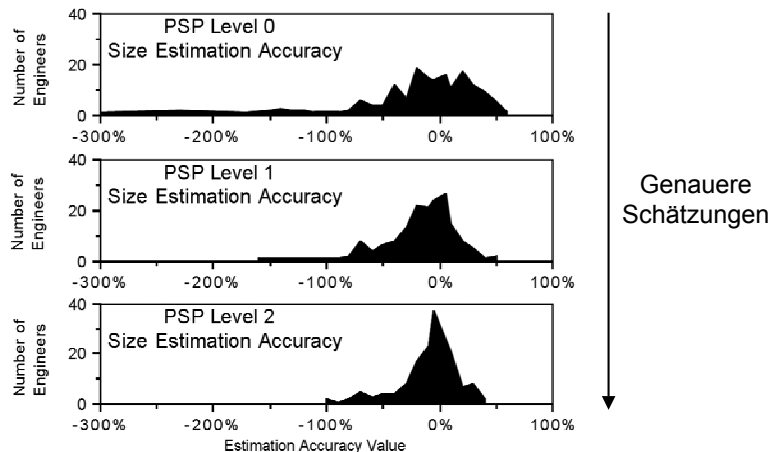
Assignment	1	2	3	4	5	6	7	8	9	10
Planning	268	271	267	261	248	239	226	218	209	152
Design	266	265	264	251	243	233	224	213	208	152
Design Review	2	4	6	4	13	15	138	132	124	149
Code	277	276	271	262	248	239	225	219	209	152
Code Review	2	7	8	10	20	22	223	219	208	152
Compile	277	276	271	260	246	239	222	214	209	151
Test	276	276	269	262	247	239	226	219	209	152

Quelle [10]

Design- und Codereviews werden erst mit Aufgabe 7 (PSP2) eingeführt, vorherige Eingaben wurden von Testpersonen mit Vorerfahrung in Eigeninitiative erstellt.



Ergebnisse – Schätzung der Programmgröße



Quelle [10]

44

Hypothese:

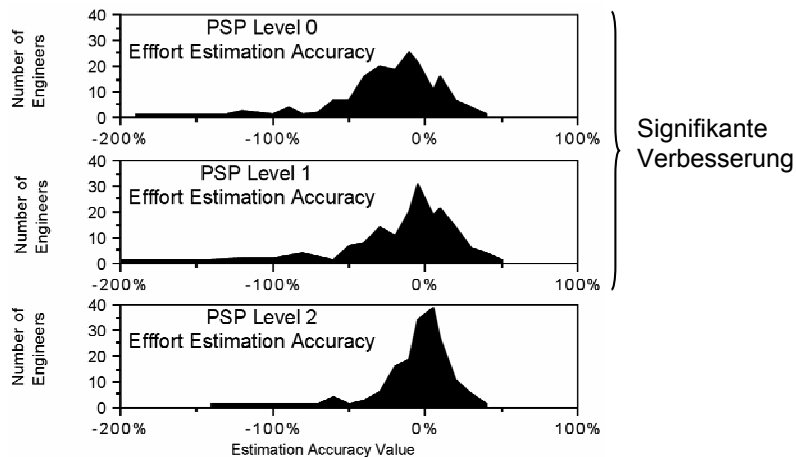
„As engineers progress through the PSP training, their size estimates gradually grow closer to the actual size of the program at the end. More specifically, with the introduction of a formal estimation technique for size in PSP level 1, there is a notable improvement in the accuracy of engineers' size estimates”

Schätzung der Größe eines Programms:

- Daten von 170 Testpersonen
- Schätzungen werden zunehmend genauer
- Über- und Unterschätzungen gleichen sich bei PSP-geschulten Testpersonen eher aus



Ergebnisse – Schätzung des Arbeitsaufwandes



Quelle [10]

45

Hypothese:

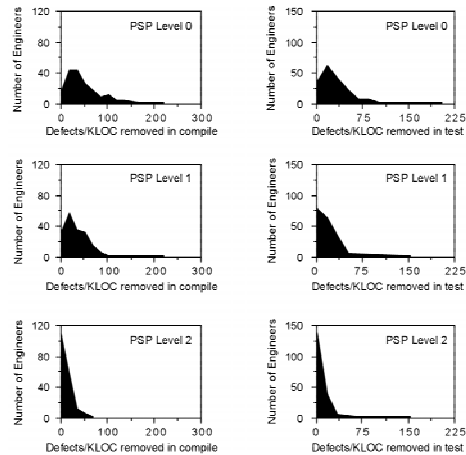
“As engineers progress through the PSP training, their effort estimates grow closer to the actual effort expended for the entire life cycle. More specifically, with the introduction of a statistical technique (linear regression) in PSP level 1, there is a notable improvement in the accuracy of engineers' effort estimates.”

- Signifikante Verbesserung zwischen PSP0 und PSP1 (Einführung der Schätzung)
- Nicht signifikant zwischen PSP1 und PSP2

→ 50% der Testpersonen reduzierten den Schätzfehler um einen Faktor von 1.75 oder mehr



Ergebnisse – Fehlerdichte



Fehlerdichte nimmt
(signifikant) ab

Quelle [10]

46

Hypothese 1:

„As engineers progress through PSP training, the number of defects injected and therefore removed per thousand lines of code (KLOC) decreases.“

Daten von 181 Testpersonen

Ergebnisse:

Signifikante Unterschiede zwischen den PSP-Ebenen bezüglich Fehlern in Test-Phase, Compile-Phase

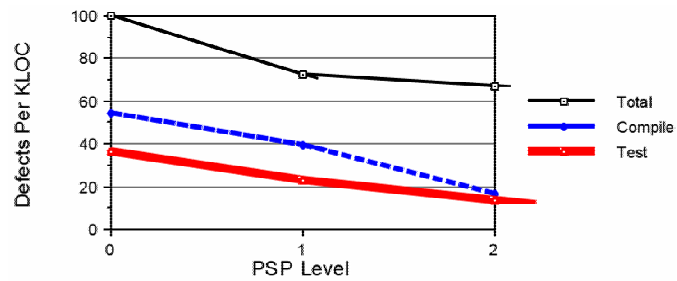
Reduktion der gesamten Fehler um Faktor 1.5 (Median)

Reduktion der Fehler in der Compile-Phase um Faktor 3.7 (Median)

Reduktion der Fehler in der Test-Phase um Faktor 2.5 (Median)



Ergebnisse – Fehlerdichte



Quelle [10]

47

Hypothese 2:

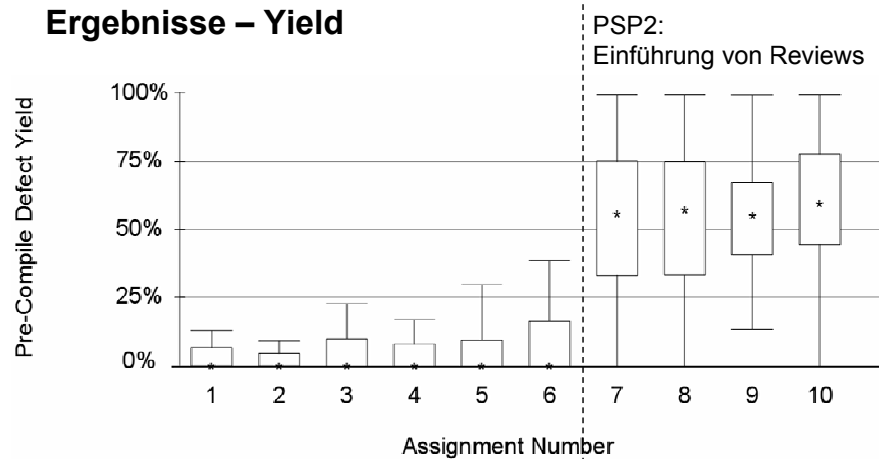
“With the introduction of design and code reviews in PSP level 2, the defect densities of programs entering the compile and test phases decrease significantly.”

Ergebnis:

Die Gesamtfehlerhäufigkeit sinkt zwischen PSP1 und PSP2 nicht signifikant, jedoch die Fehlerhäufigkeit in Compile- und Testphase



Ergebnisse – Yield



Quelle [10]

48

Hypothese:

“As engineers progress through the PSP training, their yield increases significantly. More specifically, the introduction of design review and code review following PSP level 1 has a significant impact on the value of engineers’ yield.”

Daten von 188 Testpersonen

Ergebnisse:

Mindestens die Hälfte der Testpersonen entfernte keine Fehler vor dem Kompilieren (vgl. Median (Stern in der Box))

Signifikanter Anstieg des Yield in PSP2. Dies führen die Autoren auf die Einführung von Code- und Designreviews zurück.

Mittleren Erhöhung des Yield um 50%



Ergebnisse – Produktivität

PSP Level	Average Productivity
0	23.927 LOC/Hr
1	22.958 LOC/Hr
2	24.899 LOC/Hr

Quelle [10]

49

Hypothese:

As engineers progress through the PSP training, their productivity increases. That is, the number of lines of code designed, written, and tested, per hour spent increases between the first and last assignments.

Daten (LOC) waren durch die Vorgaben von PSP vorhanden.

Ergebnisse:

Die Produktivität steigt **nicht** signifikant.

Zwischen PSP0 und PSP1 sinkt sie sogar.

Problem: Kein Vergleich mit Arbeit **ohne** PSP!



Bewertung der Studie

Untersuchung unter Laborbedingungen

- Sind die Voraussetzungen für die ANOVA gegeben?
- Können die Ergebnisse auf die Realität übertragen werden?

Autoren sind/waren am SEI beschäftigt

- objektiv?

Unstimmigkeiten innerhalb der Studie

Vorraussetzungen für ANOVA:

- repräsentative und zufällig ausgewählte Stichprobe
 - Das ist sie wohl nicht (geben die Autoren aber auch teilweise zu)
- Die Untersuchungen müssen unabhängig sein
 - Besonderes Problem: Die Leiter der Schulungen ändern diese teilweise ab
- Normalverteilung der abhängigen Variablen
 - Verteilung ist schief (die Autoren behaupten, das in den Griff bekommen zu haben – müsste man sich also genauer ansehen)

Laborbedingungen:

- nur kleine Programme, kein Druck, etc.,
- Beispiel: Größenabschätzung: nur wenige Programme, die sich ähneln → Verbesserung ist zu erwarten

Unstimmigkeiten innerhalb der Studie:

Bsp.: Die Autoren sagen, dass für die Größenschätzung nur Daten von Testpersonen berücksichtigt wurden, die Ergebnisse zu allen Übungen abgegeben haben (ausschließlich der ersten, in der keine Schätzung durchgeführt werden musste). Das sind max. 152 (da nur 152 bei Übung 10). Ausserdem wurden bei Aufgabe 1 265 Bewertungen abgegeben?



PSP in der Praxis

72 von 78 Kursteilnehmern wollen PSP nicht weiter benutzen. Über die restlichen 6 ist nicht bekannt, ob sie es einsetzen! [12]

PSP erfordert viel Selbstdisziplin. Viele bringen diese nicht auf. [13]

Großer „overhead“ und „context-switch“ [14]

Fazit:

Zu umständlich.

Softwareentwickler und Management sind oft nicht bereit

Zusammenarbeit und Toolunterstützung kann helfen



ISO/IEC 15504

Internationaler Standard für Software Prozess Assessments

Aktuelle Version (2003-2005)

Anforderungen an Prozessmodell

- z.B.: ISO 12207 (Prozesse im Software-Lebenszyklus)

Anforderungen an Assessmentmethode

- z.B.: SPICE (Assessment-Methode für ISO 12207)

Projekt wurde 1993 von ISO und IEC gestartet

1998: Technical Report

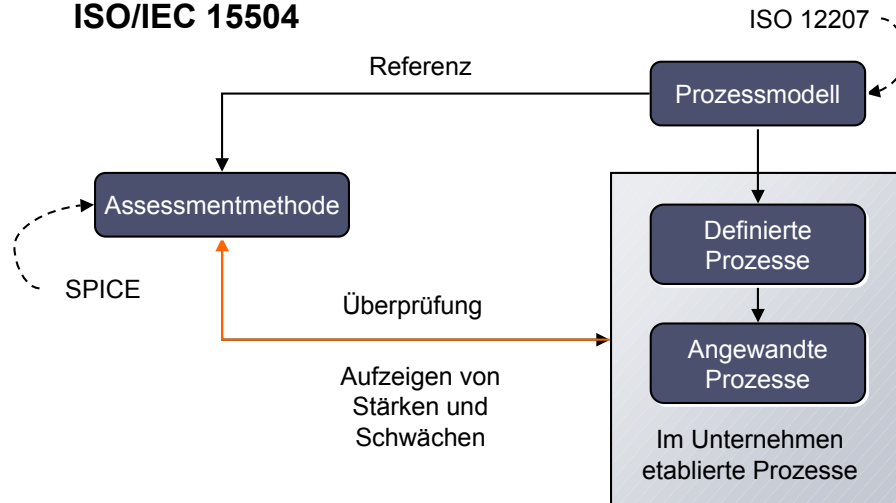
2003-2005: Revision → International Standard

Restrukturierung:

TR war auf Software Prozesse beschränkt, in der neuen Version (ähnlich wie bei CMMI) ist dies erweitert worden, indem verschiedene Prozessmodelle verwendet werden können



ISO/IEC 15504



Ausgehend von einem Prozessmodell wie z.B. ISO 12207 (oder auch 15288 für System-Entwicklung) werden in einem Unternehmen Prozess definiert und angewandt (ohne die tatsächliche Umsetzung ist die Definition von Prozessen sinnlos).

Das Prozessmodell wird von einer Assessmentmethode wie z.B. SPICE verwendet, um Stärken und Schwächen der im Unternehmen etablierten Prozesse aufzuzeigen. SPICE beruft sich hierbei auf ISO 12207 als Prozessmodell.

Umgekehrt kann auch die Assessmentmethode selbst überprüft werden.



ISO 12207 – Prozesskategorien



Primärprozesse

Beschaffung

Beschaffens von Software und Dienstleistungen

Aktivitäten des

Entwicklung

Softwareentwicklers

Aktivitäten des

Lieferung

Liefers von Software und Dienstleistungen

Aktivitäten des

Betrieb

Systemeinführung, -test, Benutzerunterstützung und Wartung

Aktivitäten wie

Organisatorische Prozesse

Management

Risiko-, Projektmanagement...

Qualitäts-,

Infrastruktur

Bereitstellung der notwendigen Infrastruktur aber auch und Schulungsmaßnahmen

Aktivitäten zur

Optimierung

Überprüfen und Verbessern der Lebenszyklusprozesse

Messen,

Wiederverwendung

managen

vorhandene Komponenten

Unterstützende Prozesse

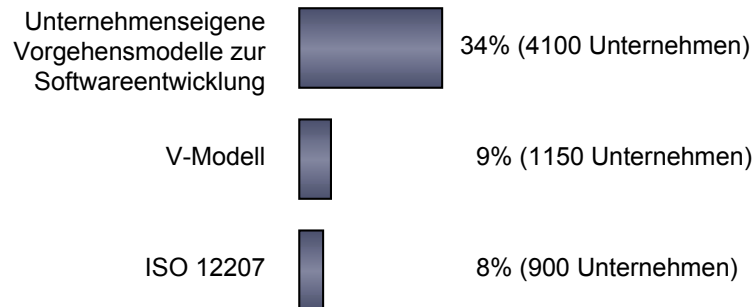
Konfigurationsmanagement

Problembhebung

Dokumentation,



Einsatz von Vorgehensmodellen



Quelle: [16]

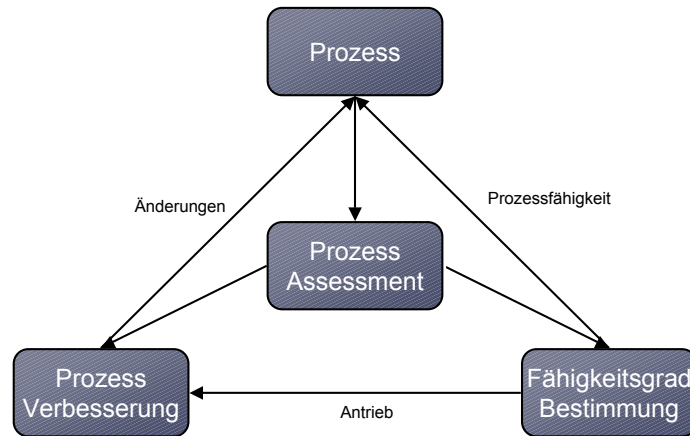
Basis der Umfrage waren 12.165 Unternehmen, die 2000 im Rahmen einer vom Bundesministerium für Bildung und Forschung in Auftrag gegebenen Studie zur Analyse und Evaluation der Softwareentwicklung in Deutschland von der GfK Marktforschung GmbH, IESE und ISI befragt wurden.

Interessant ist, dass nur ca. 50% überhaupt angeben, ein Vorgehensmodell anzuwenden.

Neben den dominanten unternehmenseigenen Modellen sind V-Modell und ISO 12207 gleich vertreten.



SPICE



Die Idee hinter Spice ist, dass das Prozess-Assessment immens wichtig als Antrieb für SPI ist .

Anfangspunkt ist ein (im Unternehmen vorhandener) Prozess. Dieser Prozess wird durch ein SPICE-Assessment bewertet. Das Ergebnis ist eine Bestimmung des Fähigkeitsgrades und Verbesserungen des Prozesses sowohl direkt durch die Ergebnisse, als auch durch den Antrieb, einen bestimmten Reifegrad zu erreichen.



Prozessfähigkeit / Process Capability

umfasst

- Prozessplanung
- Prozessdurchführung
- Prozesslenkung & Prozessmessung
- Prozessverbesserung

Bessere Prozessfähigkeit

→ bessere Vorhersagbarkeit der Ergebnisse des Prozesses

Bewertung mittels Prozess-Assessments



Fähigkeitsgrade/Stufen und zugeordnete Attribute



58

Den Stufen sind bestimmte Aktivitäten zugeordnet, die dazu führen, dass die Ergebnisse systematisch erarbeitet werden und am Ende des Prozesses in der definierten Qualität vorliegen.

Unvollständig

Chaotischer Prozess

Durchgeführt
durchgeführt

Artefakte sind vorhanden, Prozesse werden intuitiv

Gesteuert

Verantwortlichkeiten sind definiert

Prozesse und Artefakte gemanaged,

Definiert

angepasst

Vordefinierte Prozesse werden eingesetzt und

Vorhersagbar

Leistung derselben kontrollierbar

Messen der Prozesse macht die Resultate und die

Optimierend

Verbesserung der Prozesse eingesetzt

Quantitative Messungen werden für die Optimierung und



Stufe 0 – Unvollständig

chaotischer Prozess

keine definierten Abläufe

keine klar erkennbaren Produkte





Stufe 1 – Durchgeführt

Prozessdurchführung

Es sind Produkte von Aktivitäten vorhanden
Die Prozesse werden intuitiv durchgeführt





Stufe 2 – Gesteuert

Management der Prozesse

Einschränkungen bezüglich Ressourcen und Zeit werden berücksichtigt

Management der Produkte

Anforderungen an die Ergebnisse sind spezifiziert
Art der Dokumentation
Abhängigkeiten zwischen Ergebnissen
Verwaltung von Änderungen
Verantwortlichkeiten sind definiert



Darüberhinaus wird erwartet, dass die Ziele formuliert sind und welche Einschränkungen bezüglich Ressourcen und Zeit bei der Arbeit zu berücksichtigen sind.

Prozess ist nachvollziehbar (z.B. falls Probleme auftreten)

→ Entspricht in etwa ISO 9001 Zertifizierung



Stufe 3 – Definiert

Definition der Prozesse

Einheitlicher Entwicklungsprozess

Verteilung der Prozesse

Vordefinierte Prozesse werden eingesetzt und angepasst





Stufe 4 – Vorhersagbar

Messung der Prozesse

Vordefinierte Metriken

Qualitätsbestimmung der Prozesse und Produkte

Kontrolle der Prozesse

Genau Formulierung und Kontrolle von Zielvorgaben

5. Optimierend	<input type="checkbox"/>
4. Vorhersagbar	<input type="checkbox"/>
3. Definiert	<input type="checkbox"/>
2. Gesteuert	<input type="checkbox"/>
1. Durchgeführt	<input type="checkbox"/>
0. Unvollständig	<input type="checkbox"/>

Anhand vordefinierter Metriken die Qualität der Prozesse und deren Produkte laufend ermittelt und werden die gesammelten Daten regelmässig analysiert, damit Zielvorgaben genau formuliert werden können



Stufe 5 – Optimierend

Prozessoptimierung

Änderungsbedarf wird vorzeitig erkannt
Maßnahmen werden präventiv eingeleitet

Prozessinnovation

Pilotprojekte und neue Technologien



Es findet eine **ständige** Prozessoptimierung statt



Bestimmung der Fähigkeitsgrade

Prozesse (z.B. nach ISO 12207)	Prozesse/Stufen	0	1	2	3	4	5
Bestimmung des Fähigkeitsgrades für jeden einzelnen Prozess	ENG.1						
	CUS.1						
Gesamtheit aller Prozesse → Stärken-Schwächen-Profil	SUP.1						
	SUP.7						
Nächsthöherer Reifegrad → Möglichkeiten zur Verbesserung	...						
	...						

Der Reifegrad **wird für jeden Prozess einzeln** bestimmt. (Wesentlicher Unterschied zu CMM)

Die Bewertung erfolgt anhand der oben vorgestellten 5-stufigen Skala.

Während der Bewertung muss objektiv nachgewiesen werden, dass die Anforderungen auf der entsprechenden Stufe erfüllt werden. Dieses erfolgt zum Beispiel anhand von Arbeitsprodukten, welche als Ergebnisse aus den Prozessen hervorgehen.



Studie

Datenquelle

Ergebnisse der Assessments von 29 Organisationen

Fragebögen (z.B. Produktivität, Kostenschätzung) zu 56 Projekten

Hypothesen

Eine höhere Prozessfähigkeit in der Software-Anforderungsanalyse führt zu:

- einer Steigerung der Produktivität
- kürzeren Entwicklungszeiten

Dauer der Studie: 2 Jahre

Unternehmen hauptsächlich aus Europa und Australien/Südostasien

Fragebögen wurden von den Verantwortlichen der Assessments beantwortet



Maße

Performanz

Customer:	Kundenzufriedenheit
Schedule:	Fähigkeit, Kostenzusagen einzuhalten
Budget:	Fähigkeit, Zeitzusagen einzuhalten
Requirements:	Fähigkeit, spezifizierten Anforderungen zu genügen
Productivity:	Produktivität der Mitarbeiter
Morale:	Betriebsklima / Zufriedenheit der Mitarbeiter

Größe der Organisationseinheit (OU)

Unterscheidung zwischen kleinen (<=50 IT Mitarbeitern) und großen OU

Auswertung in Fragebögen zu jedem Projekt:

Frage: „Wie würden sie die Performanz des Projektes sehen hinsichtlich...“

Antwortmöglichkeiten: „Exzellent“, „Gut“, „Zufriedenstellend“, „Schlecht“, „Unbekannt“



Auswertung für kleine OU

	Results from Repeated Imputations										Combined Results		
	Imputation 1		Imputation 2		Imputation 3		Imputation 4		Imputation 5		\bar{r}	\bar{Q}	\sqrt{T}
	\hat{Q}_1	$\sqrt{U_1}$	\hat{Q}_2	$\sqrt{U_2}$	\hat{Q}_3	$\sqrt{U_3}$	\hat{Q}_4	$\sqrt{U_4}$	\hat{Q}_5	$\sqrt{U_5}$			
BUDGET	0.011	0.045	-0.013	0.042	-0.015	0.041	0.021	0.052	-0.015	0.043	-0.017	-0.002	0.0484
SCHEDULE	-0.055	0.044	-0.06	0.047	-0.095	0.044	-0.066	0.052	-0.028	0.051	-0.274	-0.060	0.0544
CUSTOMER	0.045	0.033	0.013	0.033	0.028	0.032	0.037	0.036	0.039	0.031	0.2134	0.032	0.0357
REQUIREMENTS	0.028	0.029	0.037	0.029	0.043	0.033	0.053	0.033	0.054	0.03	0.296	0.043	0.0331
PRODUCTIVITY	0.011	0.032	-0.012	0.03	-0.012	0.025	-0.012	0.025	0.019	0.032	-0.021	-0.001	0.0333
MORALE	0.046	0.039	0.002	0.034	0.044	0.036	0.044	0.036	0.009	0.035	0.1726	0.029	0.0431

Quelle [15]

68

Keiner der Korrelationskoeffizienten ist größer als der a priori festgelegte Schwellenwert von 0,3. (nur für die Requirements-Variable **fast**)

Die Autoren schließen deshalb, dass kein signifikanter Zusammenhang zwischen höherer Prozessfähigkeit in der SRA und einer der verwendeten Maße besteht.



Auswertung für große OU

	Results from Repeated Imputations										Combined Results		
	Imputation 1		Imputation 2		Imputation 3		Imputation 4		Imputation 5		\bar{r}	\bar{Q}	\sqrt{T}
	\hat{Q}_1	$\sqrt{U_1}$	\hat{Q}_2	$\sqrt{U_2}$	\hat{Q}_3	$\sqrt{U_3}$	\hat{Q}_4	$\sqrt{U_4}$	\hat{Q}_5	$\sqrt{U_5}$			
BUDGET	0.007	0.049	-0.014	0.049	-0.074	0.049	0.074	0.043	0.011	0.052	0.009	0.0008	0.0758
SCHEDULE	0.061	0.049	0.068	0.05	0.106	0.049	0.09	0.049	0.002	0.049	0.2242	0.0654	0.0656
CUSTOMER	-0.011	0.031	-0.023	0.031	0.016	0.033	-0.043	0.03	0.009	0.037	-0.062	-0.01	0.0417
REQUIREMENTS	0.036	0.041	0.054	0.038	0.059	0.037	0.086	0.038	0.052	0.037	0.2554	0.0574	0.0430
PRODUCTIVITY	-0.162	0.034	-0.151	0.034	-0.162	0.038	-0.162	0.034	-0.162	0.036	-0.625	-0.159	0.0356
MORALE	-0.052	0.028	-0.083	0.028	-0.029	0.029	-0.038	0.029	-0.063	0.027	-0.312	-0.053	0.0365

Quelle [15]

69

Die Korrelationseffizienten von Produktivität und Moral sind über dem Schwellenwert von 0,3.

Weitere statistische Tests (Bonferroni angepasstes alpha Level!) haben ergeben, dass nur **Produktivität** signifikant mit der SRA Prozessfähigkeit zusammenhängt.

Bei **Budget** und **Schedule** wurde nicht erwartet, einen starken Zusammenhang zu finden, da diese von anderen Faktoren als SRA abhängen.

Die Kundenzufriedenheit (**Customer**) hängt (wahrscheinlich) mehr von eingehalteten Zeit- und Kostenzusagen, als der Anforderungsanalyse ab.

Interessant ist, das **Requirements** nicht stark abhängt. Andere Faktoren im Laufe des Lebenszyklus haben hier also (mit) entscheidenden Einfluss..



Ergebnisse

Eine höhere Prozessfähigkeit in der Anforderungsanalyse führte zu einer Steigerung der Produktivität der Projekte und somit zu einer Kostenreduktion.

ABER: Nur bei großen Unternehmen (mit mehr als 50 Mitarbeitern)!

Interpretation:

- Messung des Reifegrades funktioniert bei kleinen Betrieben nicht
- Die Prozessfähigkeit in der Requirement Analyse hat bei kleineren Organisationen einen geringeren Effekt auf die projektbezogene Leistung

Die Autoren denken, dass die Produktivitätssteigerung dadurch zustande kommt, dass weniger nachgearbeitet werden muss.



Bewertung der Studie

Es wird explizit in der Studie darauf hingewiesen, dass oft nur die Studien publiziert werden, die eine These / ein Modell stützen.

Es wird in der Studie auch auf andere mit dem Überprüfen von Assessmentmodellen verbundenen Problematiken eingegangen.

Die verwendeten statistischen Methoden werden detailliert erklärt.

Die Studie macht einen statistisch fundierten Eindruck

Die Studie eignet sich auch als Ausgangspunkt, um sich kritisch mit Prozessverbesserung auseinander zu setzen.



Resümee

„Egal mit welchem Prozess Sie beginnen,
Es wird nicht der Prozess sein,
Der bei Ihnen wirklich funktioniert.
Sie müssen die Verantwortung für Ihren Prozess übernehmen,
Ihn dauernd beobachten
Und ihn an die Gegebenheiten anpassen.
Letztlich muss es **Ihr Prozess** werden,
Alle anderen Etiketten sind sekundär“
(Martin Fowler)



CMM

- [1] **Process Maturity Profile – Software CMM 2003 Year End Update**
<http://www.unf.edu/~ncoulter/cen6070/handouts/2004marSwCMM.pdf>
- [2] **After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success**, Dennis R. Goldenson, James D. Herbsleb, August 1995
- [3] **Capability Maturity Model for Software, Version 1.1**, Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, 1993
- [4] **A Systematic Survey of CMM Experiences and Results**, J.D. Herbsleb, D.R. Goldenson.; Proceedings of ICSE-18, 1996, S.323f
- [5] **Software Quality and the Capability Maturity Model**, J.D. Herbsleb, D.R. Goldenson et. al.: Communications of the ACM, Juni 1997/Vol.40 No.6, S.30f



CMM

- [6] **A Systematic Survey of CMM Experiences and Results**, J.D. Herbsleb, D.R. Goldenson; Technical Report CMU/SEI-95-TR-009, 1995;
<http://www.sei.cmu.edu>

- [7] **A Critical Look at Software Capability Evaluations**, T.B. Bollinger, C. McGowan; IEEE Software 8, Juli 1991, S.25f

- [8] http://de.wikipedia.org/wiki/Capability_Maturity_Model

- [9] de.wikipedia.org/wiki/Capability_Maturity_Model_Integration



PSP

- [10] **The Personal Software Process (PSP): An Empirical Study of the Impact of PSP on Individual Engineers**, Will Hayes, James W. Over, December 1997
- [11] http://de.wikipedia.org/wiki/Personal_Software_Process
- [12] **Teaching PSP: Challenges and lessons learned**, J. Borstler, D. Carrington, G. Hislop, et al., IEEE Software, 19(5), September 2002
- [13] **Experience Report: Teaching and Using the Personal Software Process (PSP)**, Prechelt, Submission to ESEC 1997.
- [14] **Beyond the PSP: Metrics collection and analysis for the differently disciplined**. Johnson, Kou, Agustin et al., Proceedings of the 2003 International Conference on Software Engineering. Portland, Oregon, 2003



ISO/IEC 15504 – SPICE – ISO 12206

- [15] **Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Capability**, Khaled El Emam, Andreas Birk, Februar 1999
- [16] **Analyse und Evaluation der Softwareentwicklung in Deutschland**, GfK Marktforschung GmbH, IESE, ISI
- [17] http://de.wikipedia.org/wiki/ISO_15504
- [18] http://www.isospice.typepad.com/isospice_home/
- [19] **Nach CMM und BOOTSTRAP: SPiCE, Die neue Norm für Prozessbewertungen**, Stienen, Informatique 6, 1999
- [20] http://de.wikipedia.org/wiki/ISO_12207



Prozessverbesserung

- [21] http://www.tantara.ab.ca/a_isorel.htm#15504
- [22] Improving Software Process Improvement, Conradi R., Fugetta A., IEEE Software July/August 2002
- [23] **Eine geführte Tour durch die Landschaft der Software-Prozesse und Prozessverbesserung**, Glinz, Informatique 6, 1999