

Übersicht Präsentation PSP/TSP/CMMI

- 1) Situation in der SW-Entwicklung / Wartung
- 2) Die Strategie von PSP/TSP
- 3) Die PSP Prozess Elemente
- 4) PSP Qualitätsmanagement
- 5) PSP Ausbildung
- 6) TSP Implementation & Vorgehen
- 7) PSP/TSP Implementation für SW Life Cycle Management
- 8) CMMI und PSP/TSP
- 9) Literaturhinweise

1) Situation in der SW – Entwicklung / Wartung

- Die Leistungsfähigkeit einer Entwicklungsabteilung wird durch diejenige seiner Teams bestimmt → deren Leistungsfähigkeit wird durch diejenige der Mitglieder bestimmt → und die Leistungsfähigkeit des Einzelnen wird durch dessen persönliches Fachkönnen und der Prozess Disziplin bestimmt
- Heute sind diese persönlichen Praktiken selten
 - definiert
 - gemessen
 - rapportiert
 - bewirtschaftet, geführt
- Nur wenige Organisationen wissen das Potential ihrer Mitarbeiter wirklich erfolgreich zu nutzen
- Angenommen ein Software Team definiert seine Prinzipien zur Qualitätsarbeit und folgt diesen dann auch → Welche Art von Arbeit würde dieses Team wohl vollbringen ?

2) Die Strategie von PSP / TSP

- Das grundsätzliche Ziel von PSP/TSP ist, Qualitätsarbeit als etwas natürliches und normales zu ermöglichen.

- Um dies zu tun, müssen die Entwickler
 - Ihre Arbeit planen und überwachen (rapportieren)
 - Die Qualität ihrer Produkte planen und messen
 - Den definierten und abgemachten Prozessen folgen

- Dies wiederum bedingt, dass die Entwickler
 - Besitzer ihrer eigenen Prozesse sind
 - An die gemachten Pläne glauben und dazu stehen
 - Danach streben, Qualitätsprodukte herzustellen

2) Die Strategie von PSP / TSP

- Die Annahme der Herausforderungen im Software Geschäft benötigt ein starkes Engagement auf allen Ebenen einer Organisation, speziell auch im Management.
- Der PSP und TSP führt zur Einführung
 - eines definierten, gemessenen und geplanten Entwicklungs Prozesses
 - sich selbst organisierenden und führenden Teams
 - effektiven Coaching- und Führungs Praktiken
 - eines auf Fakten basierten Management Systems
 - einer auf Qualität basierenden Organisations-Kultur

3) Die PSP Prozess Elemente / Basiselemente

Phase	Purpose	To guide you in developing module-level programs
	Inputs Required	<ul style="list-style-type: none"> Problem description PSP project plan summary form Time and defect recording logs Defect type standard Stop watch (optional)
1	Planning	<ul style="list-style-type: none"> Produce or obtain a requirements statement. Estimate the required development time. Enter the plan data in the project plan summary form. Complete the time log.
2	Development	<ul style="list-style-type: none"> Design the program. Implement the design. Compile the program and fix and log all defects found. Test the program and fix and log all defects found. Complete the time recording log.
3	Postmortem	<ul style="list-style-type: none"> Complete the project plan summary form with actual time, defect, and size data.
	Exit Criteria	<ul style="list-style-type: none"> A thoroughly tested program Completed project plan summary with estimated and actual data Completed defect and time logs

Scripts (Drehbücher)

Dokumentieren die Prozess Eingangs-Kriterien, Phasen/Schritte und die Prozess Ausgangs-Kriterien. Der Zweck ist die Unterstützung bei der Prozessanwendung.



Metriken

Messen den Prozess und das Produkt. Diese Informationen geben Einblick in die Arbeitsweise der Prozesse und den Arbeitsstand.

	Plan	Actual	To-Do
Summary			
Problem description			
PSP project plan summary form			
Time and defect recording logs			
Defect type standard			
Stop watch (optional)			
1. Planning			
2. Development			
3. Postmortem			
Exit Criteria			

Vorlagen

Bieten ein angepasstes und konsistentes Arbeitsgerüst zum Sammeln und Aufbewahren von Daten (Informationen).

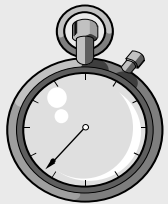


Standards

Bieten konsistente Definitionen zur Durchführung der Arbeit und dem Sammeln von Daten.

3) Die PSP Prozess Elemente / Metriken

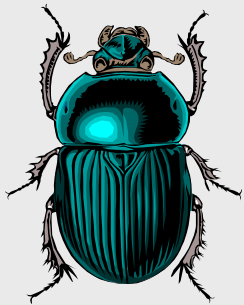
- Zur Steuerung des PSP Prozesses werden 3 Metriken eingesetzt



Zeit benötigte Task Zeit pro Phase
=
Start Zeit - Ende Zeit – Unterbruchs Zeit

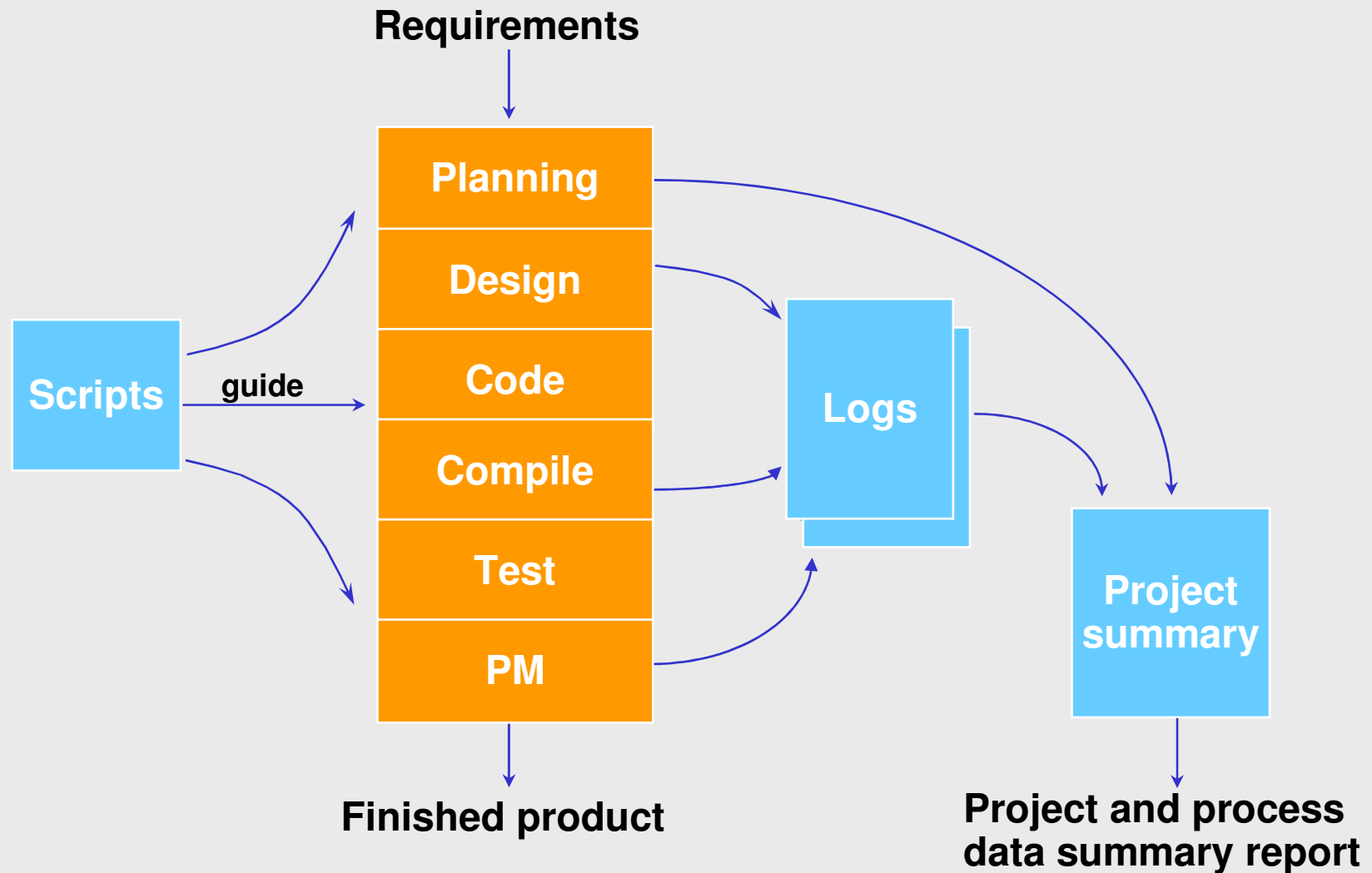


Grösse Messeinheit = SLOC (**S**ource **L**ines **O**f **C**ode)



Defekte Defekte produziert/behoben in jeder Phase
=
irgend etwas das angepasst, respektive verändert werden muss, damit das Produkt korrekt arbeitet

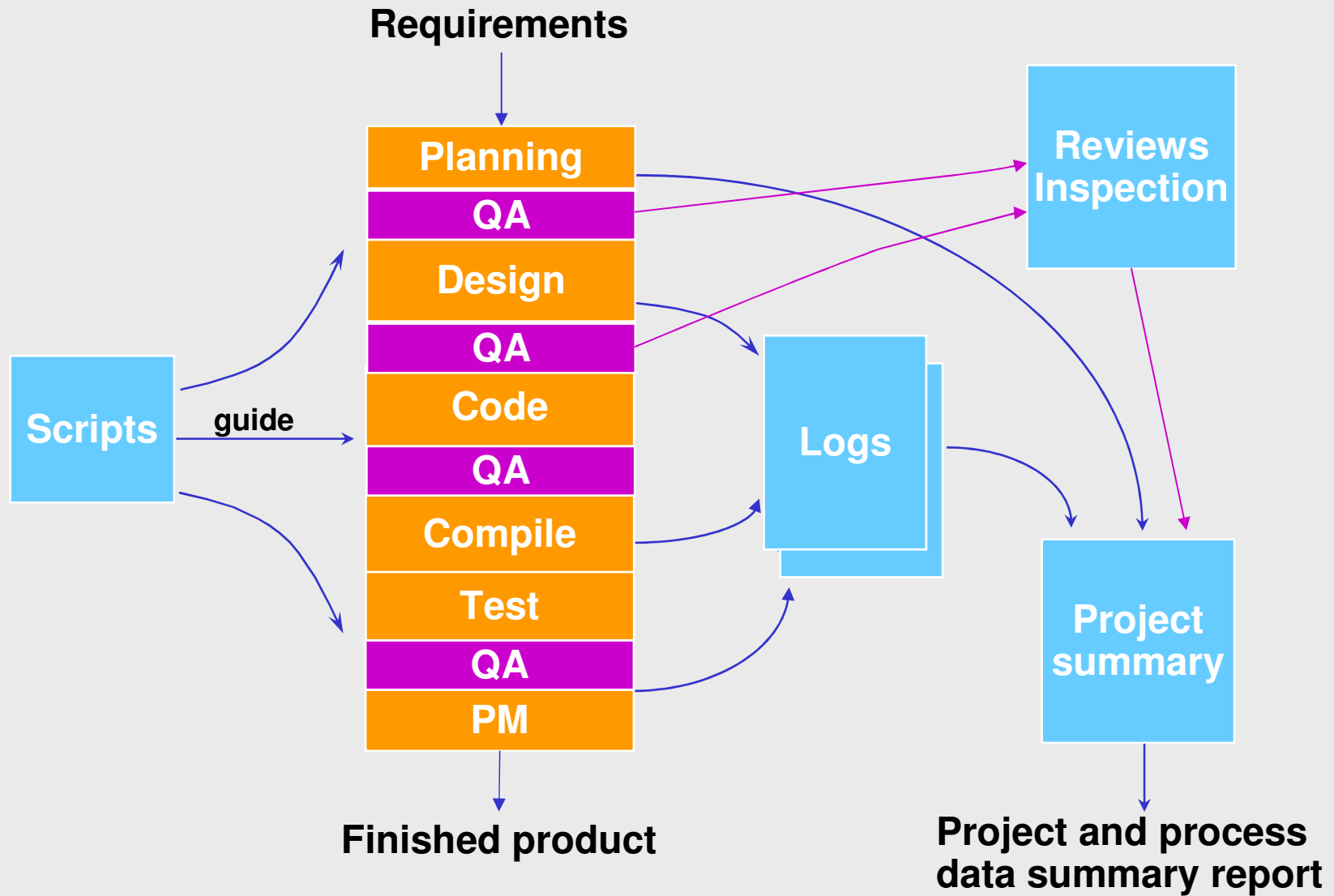
3) Die PSP Prozess Elemente / Basis-Phasen



4) PSP Q-Management / Was ist SW Qualität ?

- Qualitativ hochstehende Produkte müssen alle Benutzer-Anforderungen erfüllen
- Qualitäts-Software muss im wesentlichen schon vor der finalen Testphase (Acceptance – Test) möglichst frei von Defekten sein
- Ein Defekt ist etwas unkorrektes innerhalb eines Programmes und muss daher behoben werden
- Defekte führen zu
 - unkorrekten, unsicheren, oder möglicherweise gefährlichen Abläufen
 - Installations Problemen
 - verwirrender oder falscher Dokumentation
 - Fehler behafteten Modifikationen und Verbesserungen

4) PSP Q-Management / QA-Phasen



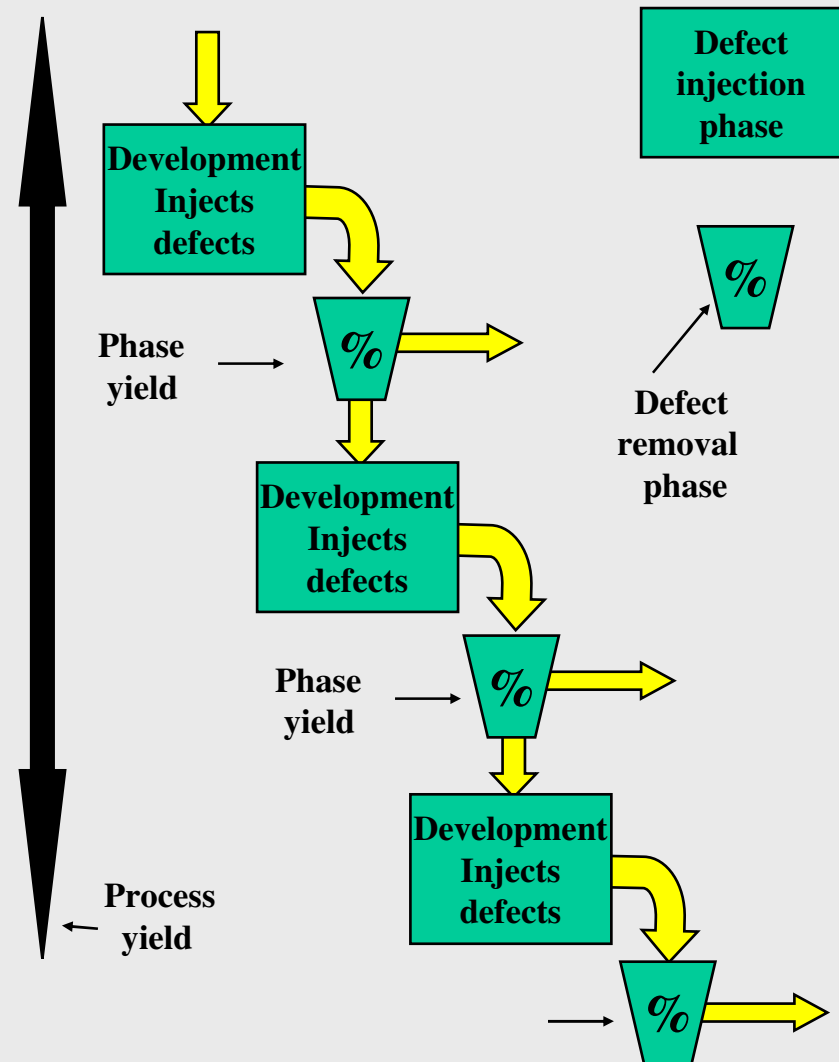
4) PSP Q-Management / Defect Yield Management

Der PSP erfasst/misst die Ausbeute (yield).

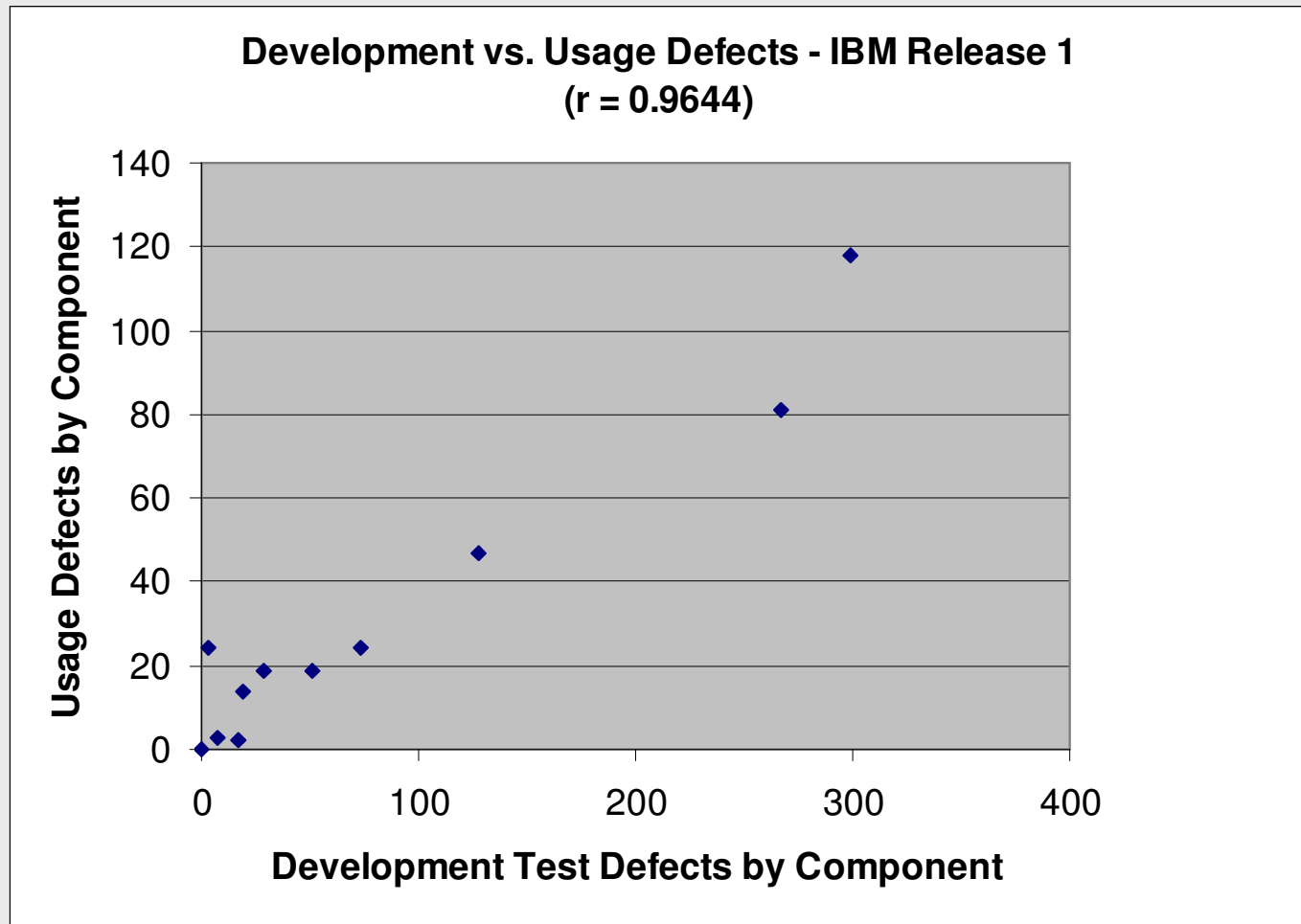
Phase yield (Ausbeute pro Phase) berechnet sich wie folgt: **(reparierte Defekts) / (Übernommene Defekts + produzierte Phasen Defekts) * 100**

Process yield (Prozess Ausbeute) berechnet sich wie folgt: **Ist die Ausbeute aller Phasen bis zum aktuellen Stand des Prozesses**

Compile process yield (Ausbeute des Kompilations Prozesses): **ist das Total der Ausbeute bis zur Phase Compile**



4) PSP Q-Management / Testabdeckung



*Kaplan, *Secrets of Software Quality – 40 Innovations from IBM*, page 173

4) PSP Q-Management / Beispiele Resultate

- Kosteneinsparung durch frühzeitige Fehlerbehebung
 - ✓ weniger Testaufwand (85% weniger; IBM)
 - ✓ weniger Nacharbeit nach Integrationstest
 - ✓ weniger Aufwand zur Fehlerbehebung nach Produktionsfreigabe
- Zeiteinsparung: 20% bis 35% durch frühere Freigabe
- Produktivitätsverbesserung (bis Faktor 2; IBM)
- weniger Ärger durch Verzögerungen und Probleme
- Einsparungen als Resultat der Prozeßverbesserungen auf der Basis der Inspektions- respektive Review-Daten und Ursachen-Analyse

Siehe auch Erfolgstory von PSP/TSP Implementation bei Microsoft
<http://www.sei.cmu.edu/publications/news-at-sei/features/2004/2/feature-1-2004-2.htm>

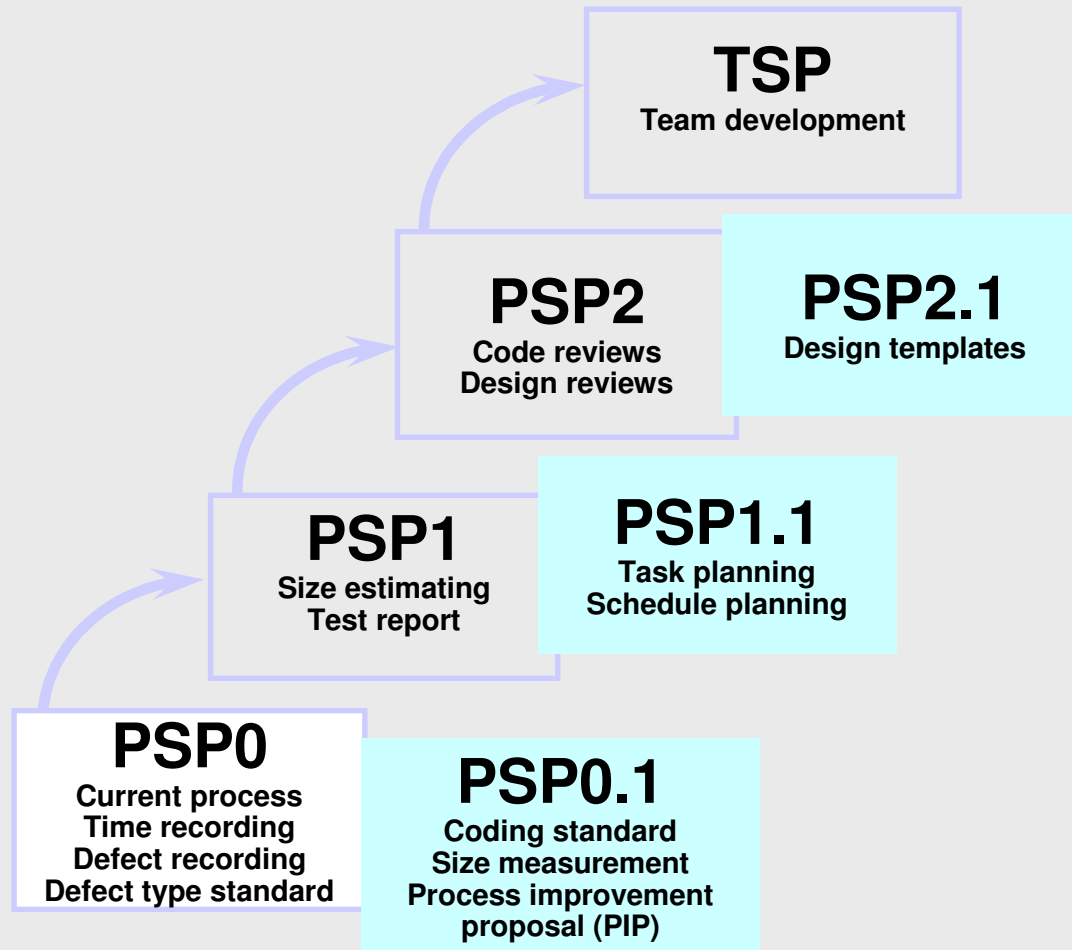
5) PSP Ausbildung / Vorgehen

Der PSP Prozess wird in sechs aufwärts kompatiblen Schritten trainiert:

- Sie schreiben pro Schritt eines oder mehrere Programme, in Modulgrösse.
- Sie sammeln und analysieren die Daten Ihrer Arbeit.
- Sie benutzen die Resultate der Analyse zur Verbesserung Ihrer persönlichen Leistungsfähigkeit.

- PSP0: Sie etablieren eine ermittelte Leistungs-Baseline.
- PSP1: Sie erstellen Pläne bezüglich Grösse, Ressourcen und Schedule (Arbeitsplan).
- PSP2: Sie erlernen das Anwenden des Defekt und Yield-Managements.

5) PSP Ausbildung / Vorgehen



5) PSP Ausbildung / Resultate

- Sie werden praktische Erfahrungen mit den zentralen Elementen eines industriell erprobten Software Entwicklungs Prozesses gemacht haben.*
- Sie werden verstehen, welche Methoden für Sie am effektivsten sind.
- Sie werden die Grundlagen haben, um eine bessere Arbeit zu machen.
- Sie werden langfristige Ziele zur Verbesserung haben.
- Sie sind bereit, als aktives Mitglied in einem TSP Team zu wirken

*Diese entsprechen generell CMMI level 5 Prozessen.

6) TSP Implementation

- Softwareprodukte bestehen aus Software Modulen und/oder Komponenten.
- Diese Module sind gestaltet, gebaut, integriert, und getestet durch ein Team von Software-Entwicklern.
- Das Fachkönnen, der Geist, die Disziplin und Hingabe des Teams beeinflussen die Resultate.

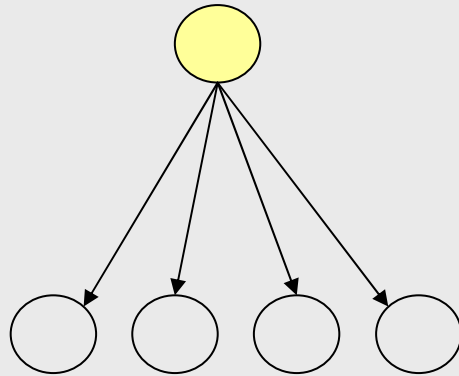
- **Ein SW Produkt ist das Resultat einer Team Anstrengung**



6) TSP Implementation / erfolgreiche TeamPraktiken

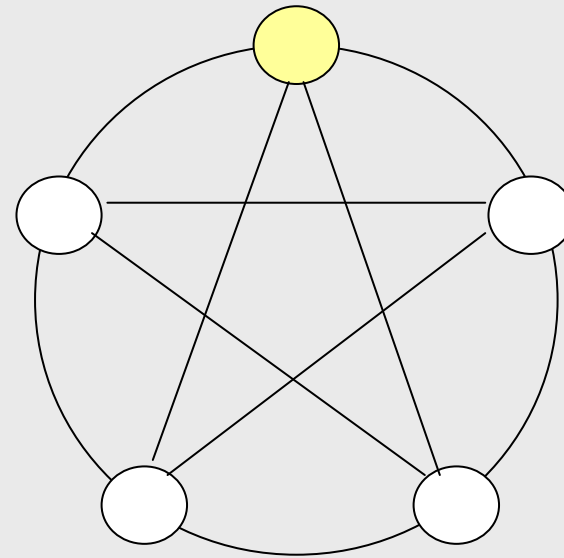
- Arbeitet ein Entwicklungsteam gut, so entstehen üblicherweise auch erfolgreiche Projekte.
- Um erfolgreich zu sein, braucht ein Team
 - klare Ziele
 - beschriebene und etablierte Rollen
 - definierte Prozesse
 - gemeinsam vereinbarte Pläne
 - Einen guten “Geist”
- Teams müssen zudem motiviert sein, und jedes Mitglied muss sich persönlich den getroffenen Abmachungen verpflichten.
- Ohne diese Rahmenbedingungen respektive Werte wird ein Team kaum erfolgreich sein.

6) TSP Implementation / sich selbst führende Teams



Traditionelles Team

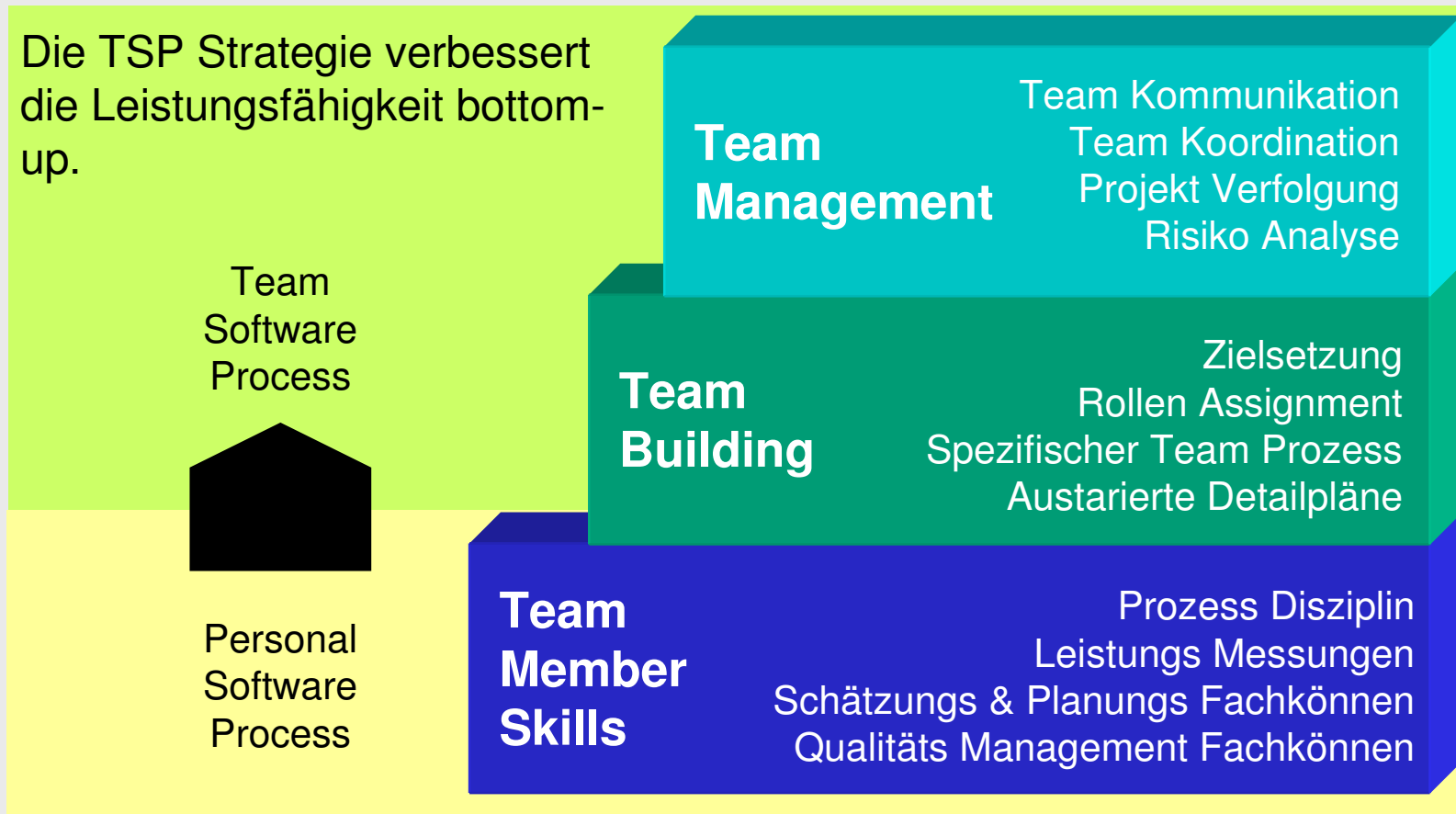
Der Leader plant, gibt Anweisungen und überwacht den Fortschritt der vergebenen Aufträge.



Selbst führendes Team

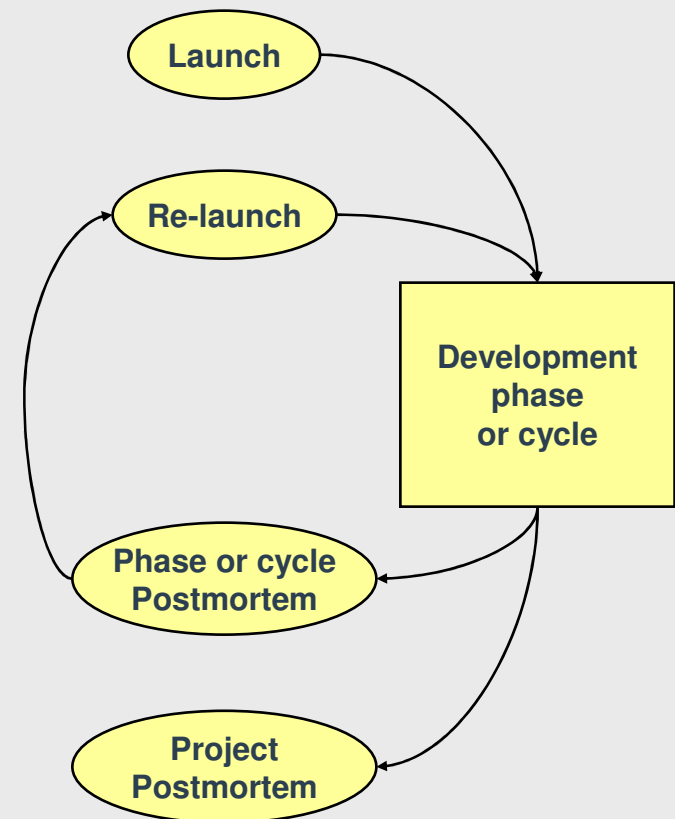
Die Teammitglieder beteiligen sich aktiv an der Planung, der Führung und der Fortschrittsüberwachung der zugewiesenen wie auch der Teamaufgaben.

6) TSP Implementation / formieren von Teams



6) TSP Implementation / Vorgehen

- Ein TSP Projekt kann grundsätzlich in jeder Phase oder Zyklus gestartet werden.
- Jede Phase oder Zyklus startet mit einem Launch oder Re-Launch und endet mit einem Postmortem.
- Die Strategie zielt auf das
 - Schrittweise Entwickeln
 - Benützen mehrerer Zyklen
 - Voraussesehendes Arbeiten
- TSP erlaubt eine den geschäftlichen und technischen Bedürfnissen angepasste Prozess-Struktur.



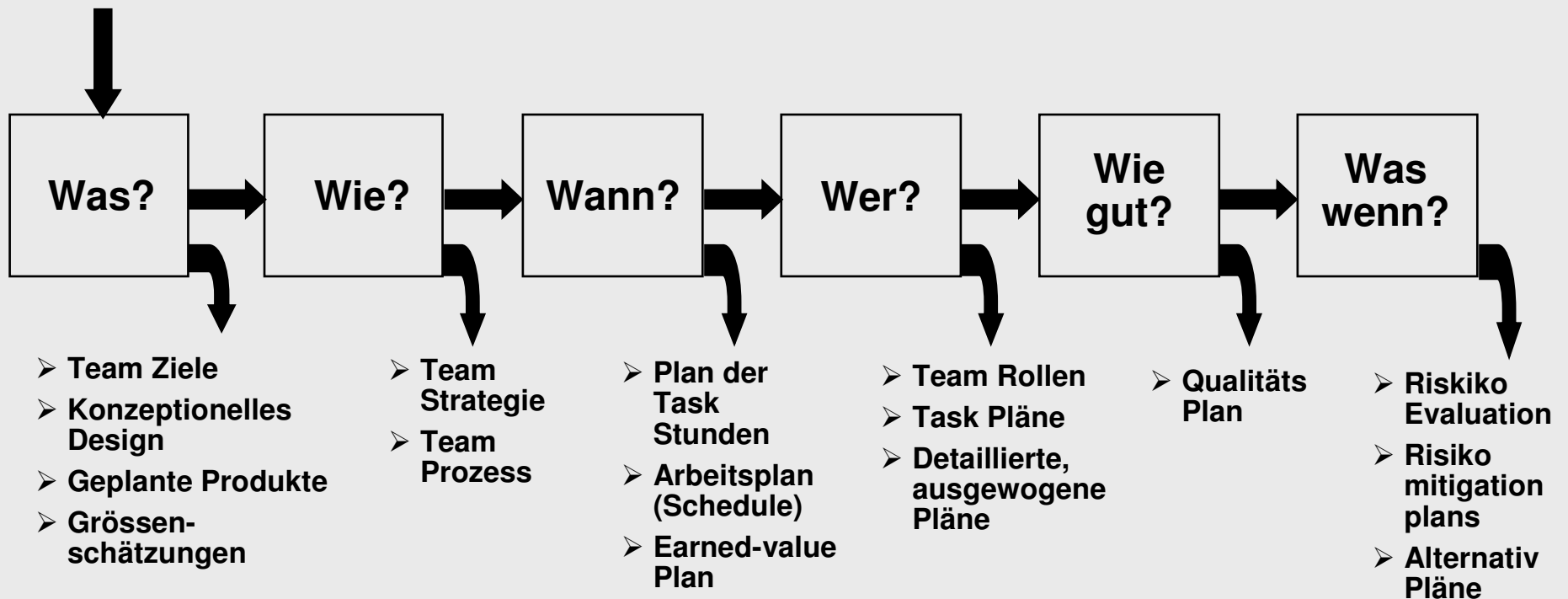
6) TSP Implementation / Launch Prozess



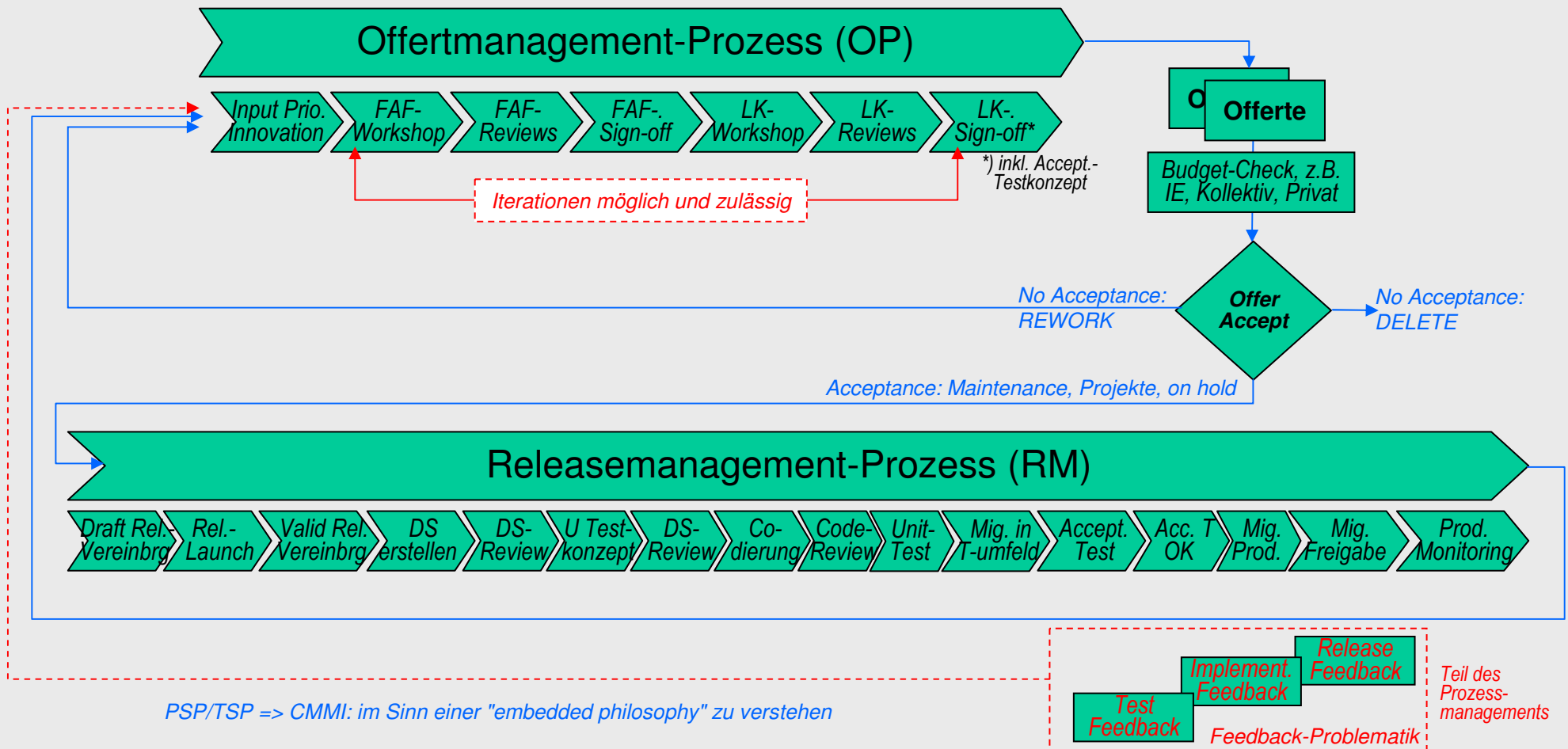
Ein qualifizierter TSP Team Coach führt das Team durch einen definierten Prozess zur Entwicklung eines ausführungsfähigen Planes der mit dem Management Team, sofern notwendig, verhandelt wird.

6) TSP Implementation / Launch Produkte

- Geschäfts Bedürfnisse
- Management Ziele
- Produkt Anforderungen



7) PSP/TSP Implementation für SW Life Cycle Management

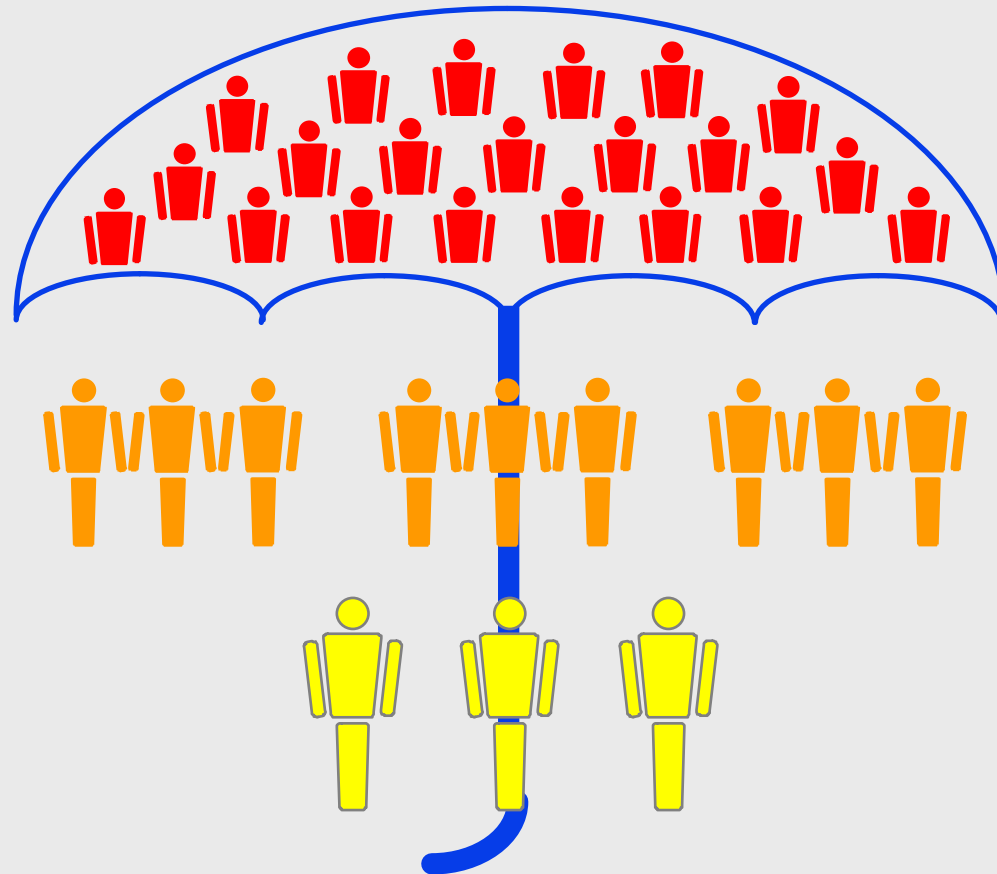


8) CMMI und PSP/TSP drei operative Ebenen

**CMMI - für
organisationsweite
Fähigkeit**

**TSP – für Team
Qualitätsprodukte
gemäss Plan und
vereinbarten
Kosten**

**PSP – für
individuelles
Fachkönnen und
Disziplin**



8) CMMI und PSP/TSP

Grundsätzliche Ziele von CMM/CMMI

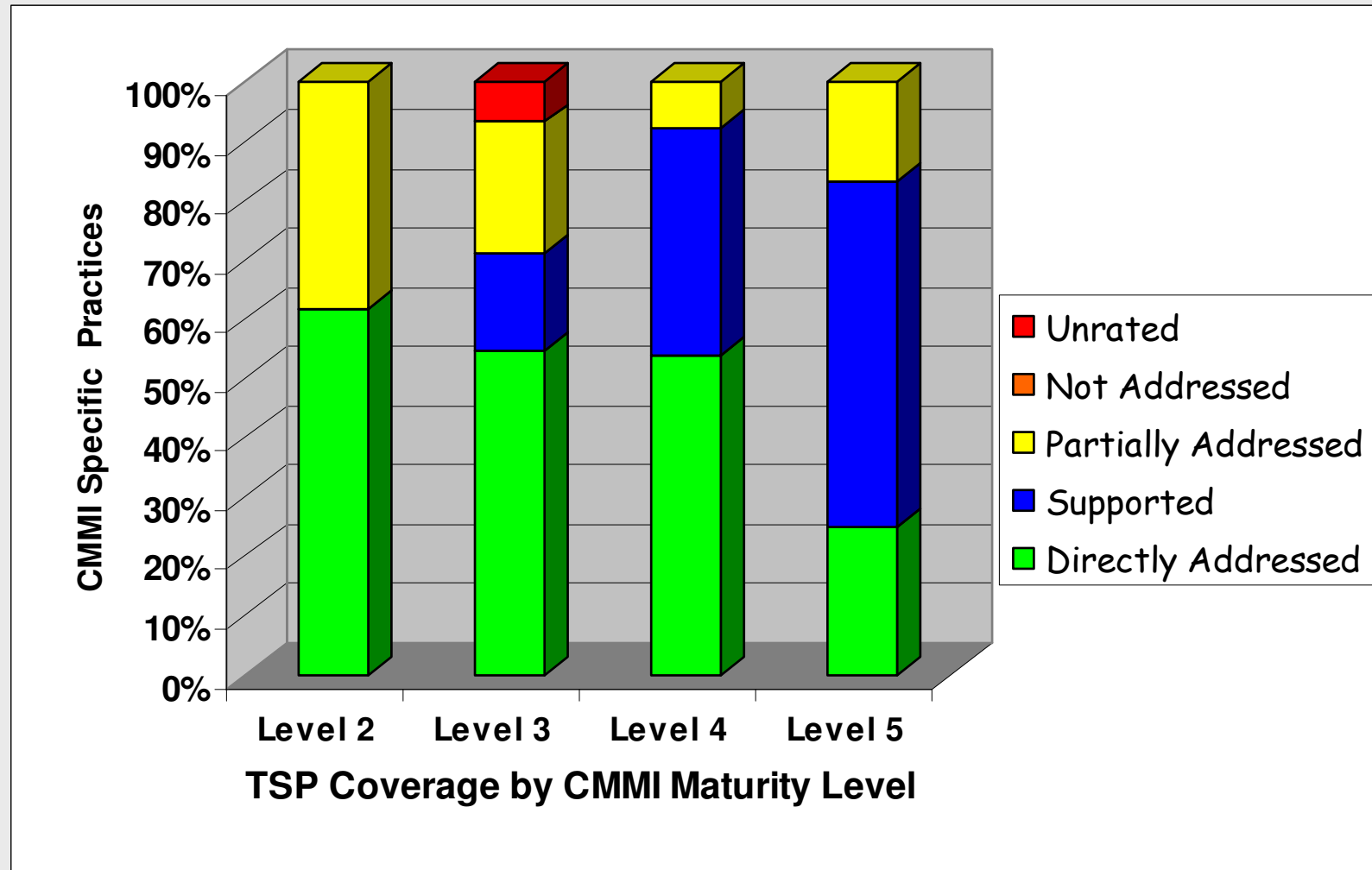
- Die ursprünglichen Ziele von CMM haben sich für und mit CMMI nicht verändert.
 - Produzieren von Qualitätsprodukten
 - Gemäss vereinbartem Plan
 - Zu den tiefst möglichen Kosten

- CMMI versteht, dass diese Ziele für den gesamten Engineering Lebenszyklus und nicht nur für den Software Entwicklungs Zyklus gelten.

- PSP und TSP wurden entwickelt, um die Erreichbarkeit der Ziele von CMM/CMMI auf individueller, respektive Team Ebene zu unterstützen und zu ermöglichen.

- PSP/TSP ermöglichen das beschleunigte Erreichen von CMMI-Level 4 (Spezialreport NAVAIR, Dez-05)

8) CMMI und PSP/TSP , Kompatibilitätsgrad



Technical Report CMU/SEI-2004-TR-014 „Mapping TSP to CMMI“ James McHale & Daniel S. Wall

9) Literaturhinweise

PSP/TSP

„Introduction to the Team Software Process“	by Watts S. Humphrey	ISBN 0-201-47719-X
„Winning with Software“	by Watts S. Humphrey	ISBN 0-201-77639-1
„A Discipline for Software Engineering“	by Watts S. Humphrey	ISBN 0-201-54610-8
„Introduction to the Personal Software Process“	by Watts S. Humphrey	ISBN 0-201-54809-7
„PSP A Self-Improvement Process for SW-Eng.“	by Watts S. Humphrey	ISBN 0-321-30549-3
„TSP Leading a Development Team“	by Watts S. Humphrey	ISBN 0-321-34962-8
„TSP Coaching Development Teams“	by Watts S. Humphrey	ISBN 0-201-73113-4

CMM/CMMI

„Practical insight into CMMI“	by Tim Kasse	ISBN 1-58053-625-5
„Real process improvement using the CMMI“	by Michael West	ISBN 0-8493-2109-3
„CMMI Guidelines for Process Integration and Product Improvement“	by Mary Beth Chrissis Mike Konrad Sandy Shrum	ISBN 0-321-15496-7

Für weitere Informationen

Hans-Peter Pfister - hans-peter.pfister@helsana.ch oder hpp@bluewin.ch

Jim McHale – jdm@sei.cmu.edu

TSP website – <http://www.sei.cmu.edu/tsp/>

Contact a PSP or TSP transition partner

<http://www.sei.cmu.edu/collaborating/partners/trans.part.psp.html>

Contact SEI customer relations

Software Engineering Institute

Carnegie Mellon University

Pittsburgh, PA 15213-3890

Phone, voice mail, and on-demand FAX: 412/268-5800

E-mail: customer-relations@sei.cmu.edu

Danke für Ihre Aufmerksamkeit

Fragen ?

Ein spezielles Dankeschön an **Frau Yveta Dvornik** für ihre wertvolle Mitarbeit