

Definition und Anwendung eines Prozess-Assessment- Modells zur Testprozessverbesserung

Abschlussarbeit

im Studiengang
Master of Computer Science

vorgelegt von

Sabine Link
Matrikelnummer: 6994806

Prüfer: Prof. Dr. F. Steimann
Betreuer: Prof. Dr. B. Hindel
Abgabe der Arbeit: 02.07.2007

„Qualität kommt von quälen.“
(Udo Bölts, Radrennfahrer)

„Der gemeinsame Nenner aller gefährdeten Projekte ist, dass diese Teams bestehende Erfahrungen nicht nutzen.“
(IBM Quality Assurance in einer Analyse gefährdeter Projekte)

Kurzfassung

Die Bedeutung von Softwaresystemen nimmt ständig zu und gleichzeitig wächst auch ihre Komplexität. Um den steigenden Qualitätsanforderungen genügen zu können, ist es wichtig die Software-Entwicklungsprozesse eines Unternehmens zu optimieren.

Reifegradmodelle sind Werkzeuge, um den Ist-Zustand dieser Prozesse zu bewerten und Verbesserungsmöglichkeiten zu identifizieren. SPICE (ISO/IEC 15504) hat sich hier zu einem der erfolgreichsten Modelle entwickelt und inzwischen sogar als internationaler Standard für die Vorgehensweise bei solchen Assessments etabliert.

Der Test nimmt bei der Qualitätssicherung in der Software-Entwicklung eine Schlüsselstellung ein. Da der Testprozess bei den gängigen Reifegradmodellen eher oberflächlich behandelt wird, wurden spezielle Testprozessverbesserungsmodelle entwickelt. TPI[®] (Test Process Improvement) von sogeti ist in diesem Bereich ein anerkanntes und praxiserprobtes Modell.

Die vorliegende Arbeit definiert aus den Ideen und Inhalten von TPI[®] ein so genanntes Prozess-Assessment-Modell entsprechend den Anforderungen der Norm ISO/IEC 15504. Dabei wurden eine Vorgehensweise zur Erstellung eines SPICE-konformen Referenzmodells und einige allgemeine Optimierungsvorschläge ermittelt.

Nach einem konzeptionellen Vergleich der beiden Modelle wurde ein Referenzmodell für TPI[®] konform zur SPICE-Norm erstellt. Einzelne Testprozesse mit konkreten Ergebnissen und Arbeitsprodukten wurden definiert und nötige Ergänzungen hinzugefügt. Die verschiedenen Ebenen von TPI[®] wurden auf die Reifegradstufen von SPICE abgebildet und in einer Matrix zugeordnet. Die Zahl der Entwicklungsstufen pro Prozess wurde dabei erweitert und vereinheitlicht. Zusammen mit dem Bewertungsrahmen hilft dies Assessoren die Stärken und Schwächen eines Testprozesses gezielt zu ermitteln und danach konkrete Verbesserungsmaßnahmen zu empfehlen.

Für das neu definierte Modell wurden im Prozess- und Projektmanagementwerkzeug project>kit der Firma method park Software AG das XML-Metamodell angepasst und anschließend die Inhalte eingefügt. Damit steht eine Browserdarstellung des Prozess-Assessment-Modells zur Verfügung, die es ermöglicht die Aktivitäten und Artefakte eines Testprojektes auf den Referenzprozess abzubilden und zu verknüpfen. Entscheidendes Kriterium bei der Verwendung eines jeden Modells ist die Akzeptanz bei den Benutzern. Dabei sind Werkzeuge ein gutes Mittel, um den Einsatz zu unterstützen.

Der Einsatz von project>kit erwies sich auch als große Hilfe beim durchgeführten Probe-Assessment, bei dem die Testprozesse eines realen Software-Entwicklungsprojektes mit dem erstellten Modell bewertet wurden. Die direkte Anwendung an einem Beispielprojekt diente zur Überprüfung und Verbesserung des Ergebnisses, um ein geeignetes und vollständiges Modell zu erhalten, das beim Einsatz in der Praxis hilfreich und gut anwendbar ist. Das Ergebnis des Assessments bestätigte hierbei sehr gut die Projekterfahrungen und die ermittelten Verbesserungsvorschläge wurden als zutreffend und sinnvoll empfunden.

Das erstellte Prozess-Assessment-Modell erfüllt den international anerkannten Standard von ISO/IEC 15504 und macht somit Bewertungen besser vergleichbar. Dies ist ein Beitrag, um den Testprozess und damit auch die Qualität bei Software-Entwicklungsprojekten weiter verbessern zu können.

Inhaltsverzeichnis

Kurzfassung	4
Inhaltsverzeichnis	5
1 Einleitung	7
1.1 Hintergrund.....	7
1.2 Aufgabenstellung.....	8
1.3 Aufbau der Arbeit.....	9
2 Grundlagen und Arbeitsmittel	10
2.1 Reifegradmodelle.....	10
2.1.1 Allgemeines	10
2.1.2 Überblick	11
2.1.3 SPICE	13
2.2 Testprozessverbesserungsmodelle.....	18
2.2.1 Testen in der Software-Entwicklung	18
2.2.2 Übersicht.....	19
2.2.3 TPI®	21
2.3 Werkzeugunterstützung mit project>kit	25
3 Umsetzung und Ergebnisse.....	28
3.1 Vergleich von TPI® und SPICE	28
3.2 Das Prozess-Referenzmodell.....	29
3.3 Der Bewertungsrahmen	32
3.4 Das Prozess-Assessment-Modell.....	36
3.5 Integration in project>kit	40
4 Praktische Anwendung	43
4.1 Das SPICE-Assessment	43
4.2 Durchführung	43
4.3 Assessment-Ergebnisse	44
4.4 Maßnahmen	45
4.5 Ausblick.....	46
5 Diskussion.....	47
5.1 Interpretation und Bewertung der Ergebnisse	47
5.2 Vergleich mit verwandten Arbeiten	50
6 Schlussbetrachtungen.....	53
6.1 Zusammenfassung	53
6.2 Ausblick.....	53
6.3 Fazit	54
Anhang A: Beispielprozesse für TPI® nach SPICE.....	55
Anhang B: Zuordnung SPICE-Levels - TPI®-Ebenen.....	58
Anhang C: Installationsanleitung für project>kit.....	60
Anhang D: Metamodell für project>kit	61
Anhang E: Beiliegende CD	63
Abkürzungsverzeichnis.....	64

Abbildungsverzeichnis	65
Tabellenverzeichnis	66
Literaturverzeichnis	67
Erklärung	71

1 Einleitung

1.1 Hintergrund

Die Anzahl und die Bedeutung der in vielen verschiedenen Bereichen verwendeten Softwaresysteme werden immer größer. Gleichzeitig nimmt die Komplexität der Software zu, während Entwicklungszeiten und Herstellungskosten abnehmen sollen. Wobei darunter die Softwarequalität nicht leiden sollte - und das nicht nur bei sicherheitskritischen oder gesundheitsgefährdenden Systemen. Die hohen Erwartungen an die erstellten Produkte werden jedoch längst nicht immer erfüllt. Viele Projekte in der Software-Entwicklung scheitern sogar ganz [WeRA06] und bringen die gesamte Branche in Verruf.

Das Testen ist deshalb im Entwicklungsprozess unverzichtbar, um so genannte „Bananensoftware“ zu vermeiden, bei der das Produkt erst beim Kunden reift und dort die Fehler gefunden werden. Dabei sollte der Test nicht nur als eine Phase am Projektende kurz vor Auslieferung stattfinden, sondern projektbegleitend von Anfang an. Testen macht bis zu 50 Prozent des Gesamtaufwandes eines Projektes aus [PKSp02], so dass eine Strukturierung dieses wichtigen Prozesses entscheidend zum Projekterfolg beiträgt. Leider wird das oft versäumt.

IBM Quality Assurance hat in einer Analyse gefährdeter Projekte festgestellt: „Der gemeinsame Nenner aller gefährdeten Projekte ist, dass diese Teams bestehende Erfahrungen nicht nutzen.“ [Nöss04]

Der Grundstein für die Qualität der Softwareprodukte kann also durch die Verwendung von bewährten Entwicklungsprozessen gelegt werden. Aus der Qualität des Software-Entwicklungsprozesses lassen sich Rückschlüsse auf die Produktqualität ziehen [HDHM06]. Um nun die Güte dieser Prozesse objektiv beurteilen und vergleichen zu können, gibt es eine Reihe von Vorgehensweisen und Modellen, mit denen auch gezielt Verbesserungsmöglichkeiten aufgezeigt werden können. Kompetenz und Erfahrung können somit konkret weitergegeben und genutzt werden.

Die Entwicklung von Reifegradmodellen begann etwa Mitte der Achtziger Jahre und dauert immer noch an. Diese lange Entwicklungsphase verdeutlicht den Anspruch und die Schwierigkeiten dieser Thematik [Isel05]. In den letzten Jahren hat sich hier SPICE (ISO/IEC 15504) [ISO05] als internationaler Standard etabliert, weswegen inzwischen viele Hersteller von ihren Zulieferern SPICE-konforme Prozesse verlangen: SPICE ist sozusagen der erwartete Stand der Technik [WeRa06].

Nachdem in den bestehenden Reifegradmodellen der Softwaretest meist etwas zu kurz kommt, entstanden speziell für den Testprozess verschiedene Bewertungs- und Verbesserungsmodelle. Das verbreitetste ist hier das aus Erfahrungen in der Praxis entstandene TPI[®] (Test Process Improvement) [PKSp02] der Firma sogeti.

Aus dieser Situation heraus entstand die Idee ein Referenzmodell für Testprozessverbesserung gemäß ISO/IEC 15504 zu erstellen, das sowohl die bewährten Inhalte von TPI[®] enthält, als auch das standardisierte Bewertungsschema und Vorgehen von SPICE beinhaltet.

1.2 Aufgabenstellung

Ziel dieser Arbeit ist es ein Prozess-Assessment-Modell (PAM) für TPI[®] nach SPICE zu definieren und anzuwenden. Mit diesem Modell soll es möglich sein, den Reifegrad eines Testprozesses in einem Unternehmen nach einem standardisierten Vorgehen objektiv zu ermitteln und gezielt Verbesserungsvorschläge zu machen.

Folgende Arbeitsschritte sind dabei zu absolvieren:

- Einführung in die Thematik der Reifegradmodelle
- Vergleich von Aufbau und Vorgehensweise bei SPICE und TPI[®]
- Beschreiben der nötigen Anforderungen zur Erreichung der SPICE-Konformität
- Definition eines Prozess-Assessment-Modells für TPI[®] nach SPICE
- Umsetzung mithilfe des Projekt- und Prozessmanagementwerkzeugs project>kit
- Pilotierung dieses Modells in einem Probe-Assessment eines Beispielprojekts
- Bewertung und Diskussion der Ergebnisse und Erfahrungen

Das zu entwickelnde Modell soll konzipiert und eine erste Version erstellt werden. Eine vollständige Validierung ist allerdings im Rahmen dieser Arbeit nicht möglich, da dazu eine Vielzahl an Testprojekten assessiert werden muss, um die Akzeptanz und Anwendbarkeit in der Praxis gründlich zu ermitteln.

Verwandte Arbeiten

Im Umfeld dieser Thematik sind einige andere Arbeiten erschienen. So wurden allgemeine Ansätze zur Verbesserung der Prozessqualität wie TQM, ISO, CMMI oder SPICE verglichen [Balz98], [RaFB04], und ebenso die bestehenden Modelle für Testprozessverbesserung einem Vergleich unterzogen [Swin00].

Zum Standard SPICE (ISO/IEC 15504) gibt es Anpassungen an verschiedene branchenspezifische Anforderungen. So sind SPICE4SPACE [CVWC01] für die Raumfahrtindustrie, und Automotive-SPICE [CHKK04] für die Automobilbranche entstanden. Ein MediSPICE für den Medizintechnikbereich ist in Arbeit.

Auch eine Gegenüberstellung von SPICE und der Safety Norm ISO 61508 wurde erarbeitet [MeSc06], was das allgemeine Interesse an diesen Standards und deren Verwendung in der Praxis weiter unterstreicht.

Für Automotive TPI[®] [Soge04], [NoBr04], [Dorn04] wurden in das TPI[®]-Modell Erweiterungen und Beispiele speziell für die Automobilindustrie eingefügt. Außerdem wurde der Inhalt des Automotive TPI[®] mit den testbezogenen Inhalten des SPICE-konformen Modells basierend auf ISO/IEC 12207 (Information Technology) verglichen [Gros04]. Dieses Modell wird für Assessments im Softwarebereich verwendet und ist als Beispielmodell in der SPICE-Norm ISO/IEC 15504 enthalten. Es enthält neben vielen anderen Prozessen auch einige Testprozesse.

Im Folgenden soll nun ein SPICE-konformes Prozess-Assessment-Modell basierend auf dem Referenzmodell TPI[®] speziell für den Testprozess bei der Software-Entwicklung erstellt werden. Zu dieser konkreten Problemstellung wurde bisher keine Literatur veröffentlicht, so dass mit ähnlichen Konzepten und Lösungsansätzen nicht verglichen werden kann. Diese Arbeit kann somit als Grundlage für weitere Überlegungen in diesem Bereich dienen.

1.3 Aufbau der Arbeit

In der Einleitung wurde bereits erklärt aus welcher Motivation heraus das Thema dieser Arbeit entstand und welche konkrete Aufgabenstellung sich daraus ergibt. In diesem Zusammenhang wurden verwandte Arbeiten kurz vorgestellt.

Das folgende Kapitel 2 führt in die Grundlagen der Modelle und Arbeitsmittel ein, die in den anschließenden Kapiteln verwendet werden. Es gibt einen Überblick der bestehenden Reifegrad- und Testprozessverbesserungsmodelle und geht besonders auf SPICE und TPI[®] ein. Außerdem stellt es das bei der Umsetzung verwendete Werkzeug project>kit vor.

Kapitel 3 beschreibt die Erstellung des Prozess-Assessment-Modells von TPI[®] nach SPICE. Hier finden sich ein Vergleich der beiden Modelle, einige Vorüberlegungen und schließlich die Erarbeitung einer Lösung des vorgegebenen Problems mit den Ergebnissen.

Um die Anwendbarkeit des Ansatzes zu überprüfen, und die Arbeitsweise bei einem Assessment zu zeigen, wird in Kapitel 4 das entwickelte Modell bei einem Beispielprojekt eingesetzt und erprobt.

In Kapitel 5 folgen die Interpretation und eine Bewertung des gewählten Lösungsansatzes und der Erfahrungen beim Praxiseinsatz. Hier werden auch Vor- und Nachteile des ermittelten Lösungsansatzes diskutiert.

Die gesamte Arbeit wird in Kapitel 6 kurz zusammengefasst und auf verbleibende Aufgaben und Herausforderungen der bearbeiteten Thematik verwiesen.

Mehrere Anhänge ergänzen die Arbeit. In Anhang A befinden sich Beispielprozesse aus dem erstellten Prozess-Assessment-Modell.

Anhang B beinhaltet Beispiele für die Zuordnung von SPICE-Levels und TPI[®]-Ebenen.

Die Anhänge C und D enthalten eine Installationsanleitung und das Metamodell für die konkrete Umsetzung in project>kit.

Die beigefügte CD in Anhang E beinhaltet eine mit dem Werkzeug project>kit erstellte HTML-Version des Referenzmodells und Tabellen, die zur Assessment-Unterstützung erzeugt wurden.

Nachdem die Umsetzung des entwickelten Modells für die Browserdarstellung im Werkzeug project>kit auf Englisch erfolgte, sind auch die Dateien im Anhang in Englisch gehalten. In der restlichen Arbeit wurde versucht zur besseren Lesbarkeit und leichterem Verständnis nach Möglichkeit deutsche Ausdrücke und Inhalte zu verwenden. Um einen Bezug zwischen beiden Begriffen herstellen zu können, werden die englischen Namen anfangs in Klammern hinter dem deutschen Wort angegeben.

2 Grundlagen und Arbeitsmittel

Um die Bedeutung von Reifegrad- und Testprozessverbesserungsmodellen in der Software-Entwicklung besser beurteilen zu können, wird nun ein kurzer Überblick der wichtigsten Modelle gegeben. Außerdem werden die Modelle SPICE und TPI[®] genauer erklärt, um die folgenden Konzepte der Arbeit besser verstehen zu können.

Am Ende dieses Kapitels wird noch das verwendete Werkzeug project>kit kurz vorgestellt.

2.1 Reifegradmodelle

2.1.1 Allgemeines

Bei Software-Entwicklungsprojekten besteht immer die Herausforderung, in der vorgegebenen Zeit und mit dem vorgesehenen Budget die geforderte und zugesagte Qualität liefern zu können.

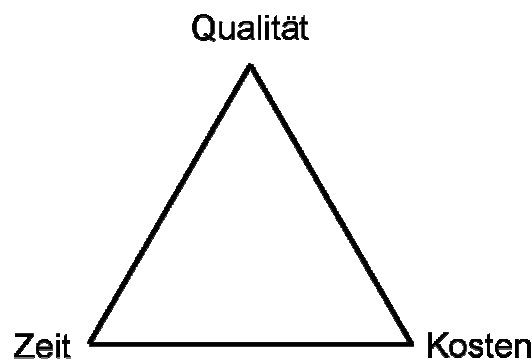


Abbildung 2.1: Das Dreieck zu Kosten, Zeit und Qualität (s. [Kneu03])

Wie das Dreieck in Abbildung 2.1 zeigt, ist es nicht möglich ein Optimum bei allen drei Anforderungen zu erreichen. So erfordert eine möglichst schnelle Lösung entweder mehr Ressourcen und damit höhere Kosten, um die Qualität zu halten, oder gleich bleibende Kosten gehen zu Lasten der Qualität. [Kneu03]

Ein Ansatz um diese Problematik zu reduzieren besteht darin, die Prozesse und Abläufe in einer Organisation genauer zu betrachten. Mit optimierten Software-Entwicklungsprozessen ist es dann möglich bessere Ergebnisse im Spannungsfeld von Qualität, Zeit und Geld zu erzielen [Isel05]. Da Fehler vermieden werden, lässt sich eine aufwändige Nacharbeit verhindern [HDHM06].

Um nun die Qualität der Prozesse objektiv bewerten und quantifizieren zu können, ist ein Konzept nötig, so dass sich jederzeit wiederholbare und vergleichbare Ergebnisse erreichen lassen. Dazu wurden von Standardisierungsgremien in Zusammenarbeit mit Industrieunternehmen Reifegradmodelle entwickelt, die eine einheitliche Vorgehensweise ermöglichen. Die Grundidee bei diesen Reifegradmodellen ist, bewährte und erfolgreiche Praktiken zur Verfügung zu stellen, so dass sich Unternehmen bei der Gestaltung der eigenen Prozesse daran orientieren können.

Bei einer Standortbestimmung wird zunächst die gelebte Praxis in den Projekten mit den Anforderungen des verwendeten Modells verglichen. Die allgemein gültigen Praktiken helfen dann die Prozesse zu verbessern. [HDHM06]

Die Verwendung von Reifegradmodellen ermöglicht mit guten Prozessen hochqualitative Produkte zu erhalten. Verlässlichere Schätzungen und höhere Planungssicherheit werden gefördert, da Erfahrungswerte aus anderen Projekten genutzt werden. Zudem wird es leichter gesetzliche Auflagen zu erfüllen. [HDHM06]

2.1.2 Überblick

Es eine Menge an Reifegradmodellen, von denen die wichtigsten hier kurz vorgestellt werden.

Meist wird die deutsche Übersetzung „Reifegrad“ sowohl für „capability level“, als auch für „maturity level“ verwendet. Dabei bezieht sich „maturity“ eher auf die Reife der gesamten Organisation [Isel05], „capability“ hingegen steht für die Reife eines einzelnen Prozessbereiches, um bei der Ausführung die erwarteten Ergebnisse zu erzielen. Ein hoher Reifegrad einer Organisation impliziert somit hohe Fähigkeitsgrade der Prozesse. [Huyn02]

In dieser Arbeit wird oft der geläufigere Begriff „Reifegrad“ in beiden Bedeutungen, also für Prozess und Organisation, verwendet.

Die folgende Tabelle 2.1 zeigt was „Reife“ bei Unternehmen bedeutet:

Unreife Organisation	Reife Organisation
Prozesse sind improvisiert und chaotisch	Prozesse sind definiert, dokumentiert, beherrscht und angewandt
Erfolg ist zufällig, nicht zwingend wiederholbar und eine Leistung Einzelner	Erfolg ist vorhersagbar und wird gemeinsam erzielt
Termine werden nicht eingehalten, Kosten überschritten, Funktionalität nicht erreicht	Termin- und Budgettreue, Funktionalität wird umgesetzt
Stresssituation, Krisenmanagement nötig	Aktives Management, Zeit wird effektiv genutzt
Technologie wird nicht optimal verwendet	Technologie wird effizient eingesetzt

Tabelle 2.1: Reife und unreife Organisationen (nach [RoSF05])

TQM

Das Total Quality Management wird auch als umfassendes Qualitätsmanagement bezeichnet und wurde ursprünglich in Japan entwickelt. In Deutschland ist das TQM-Konzept der EFQM (European Foundation for Quality Management) verbreitet [Wiki07c]. Bei dem ganzheitlichen Ansatz sind alle Mitarbeiter zuständig („Total“). Die Qualität steht im Mittelpunkt („Quality“) und das Konzept muss bewusst geplant und durchgeführt werden („Management“). [SRWL06]

TQM ist sehr allgemein und wenig konkret gefasst und nicht speziell auf die Software-Entwicklung ausgerichtet. [Balz98]

ISO 9000

Das ISO 9000 Normenwerk beschreibt Anforderungen an Qualitätssicherungssysteme und liefert damit einen allgemeinen Rahmen für das Auftraggeber-Lieferanten-Verhältnis. Aus der Reihe der ISO 9000 Standards ist vor allem ISO 9001 für die Herstellung und Wartung von Produkten und Dienstleistungen relevant [Wiki07b]. Da dieses Regelwerk für die Software-Entwicklung nicht so leicht anzuwenden ist, wurde der Leitfaden ISO 9000-3 entwickelt, der Richtlinien für den Einsatz von ISO 9001 für Software-Entwicklung enthält. [Balz98]

Ebenso wie TQM ist die ISO Norm jedoch ein sehr allgemeines Qualitätsmodell. Es beschreibt eher Anforderungen und gibt keine detaillierten Verbesserungsmöglichkeiten.

CMM

Das amerikanische Verteidigungsministerium (*Department of Defence*) hatte 1986 die Entwicklung des Capability Maturity Model (CMM) initiiert, nachdem es Softwarequalitätsprobleme bei seinen Zulieferern gab. Das Modell zur Bewertung der Reife von Softwareprozessen wurde vom Software Engineering Institute (SEI) der Carnegie-Mellon Universität in Pittsburgh entwickelt. 1991 erschien die Version 1.0 und wurde bis 1993 bis zur Version 1.1 weitergeführt [Kneu03].

Es entstanden verschiedene CMMs, neben dem bekannten CMM für Software zum Beispiel eines für Systementwicklung und ein PeopleCMM [Brab06] zur Mitarbeiterentwicklung. Nachdem all diese Modelle unterschiedlich strukturiert waren und nur schwer gemeinsam eingesetzt werden konnten, wurde ein neues Projekt mit dem Namen CMMI (Capability Maturity Model Integration) gestartet, um die verschiedenen Modelle zu integrieren. CMM wurde durch CMMI ersetzt und wird vom SEI nicht mehr weitergeführt. [Isel05]

Bootstrap

Das Bootstrap-Modell wurde von einigen europäischen Organisationen im EU-Forschungsprojekt ESPRIT (European Strategic Program for Research in Information Technology) entwickelt. [Isel05]. Es ist weitgehend kompatibel mit CMM, wurde allerdings mehr auf die Bedürfnisse der europäischen Software-Industrie abgestimmt. Außerdem werden die Anforderungen von ISO 9001 und ISO 9000-3 erfüllt. [Prec99]

Bootstrap wurde 1993 veröffentlicht, das EU-Programm wurde inzwischen allerdings eingestellt [SRWL06].

CMMI

2000 wurde die Pilotversion 1.0 von CMMI, das damals noch Capability Maturity Model - Integrated hieß, herausgegeben. 2002 folgte dann die überarbeitete Version 1.1, jetzt unter dem Namen Capability Maturity Model Integration. Die aktuelle Version 1.2 wurde 2006 veröffentlicht. [Wiki07c]

Im CMMI sind unterschiedliche Vorgängermodelle integriert. Diese haben zwar die gleichen Ziele, unterscheiden sich aber im Aufbau. Dadurch sind zwei Formen des Modells entstanden: es gibt eine stufenförmige und eine kontinuierliche Darstellung. [Kneu03]

In der stufenförmigen Variante gibt es fünf Reifegrade, die die unterschiedliche Reife einer Organisation, wie in Abbildung 2.2 zu sehen, von 1 bis 5 beschreiben:

Außer der Stufe 1 sind allen Reifegraden jeweils konkrete Prozessgebiete mit bestimmten Anforderungen zugeordnet, die erfüllt sein müssen, um den Reifegrad zu erreichen. Dadurch wird die Reihenfolge der Prozessbereiche vorgegeben, die sich bei der

schrittweisen Verbesserung einer Organisation bewährt haben. Beispielsweise wird auf Stufe 2 zwingend vorgeschrieben Projekt- und Konfigurationsmanagement zu optimieren [Tarn06].

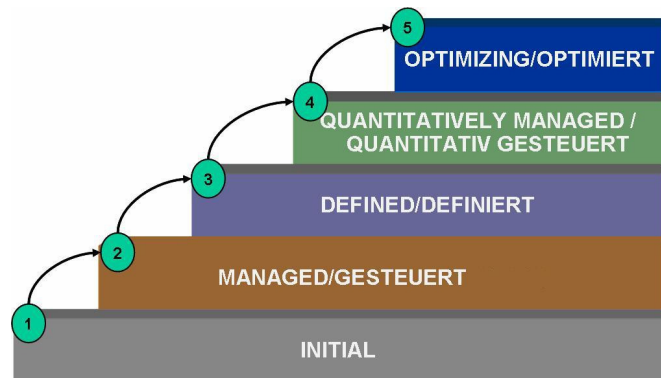


Abbildung 2.2: Die Reifegradstufen bei CMMI (nach [Kneu03])

Während sich beim stufenförmigen CMMI der Reifegrad (*maturity level*) auf die Gesamtheit der Prozessgebiete bezieht, gibt es beim kontinuierlichen Modell so genannte Fähigkeitsgrade (*capability level*), die jeweils ein einzelnes Prozessgebiet beschreiben. Damit ist eine genauere Beschreibung der Prozesse möglich [Kneu03].

Beim kontinuierlichen Modell gibt es noch einen Fähigkeitsgrad 0 „unvollständig“. Der Fähigkeitsgrad 1 heißt hier nicht „initial“, sondern „durchgeführt“ [Kneu03].

Die Bewertung einer Organisation geschieht durch die Assessmentmethode SCAMPI (Standard CMMI Appraisal Method for Process Improvement) [Kneu03].

Sowohl das Reifegradmodell CMMI, als auch das Bewertungsverfahren SCAMPI erfüllen die Anforderungen der Norm ISO/IEC 15504 (SPICE), die nun als nächstes genauer beschrieben wird.

2.1.3 SPICE

Historie

In Europa wurden ebenfalls Probleme mit der Qualität von Softwarelieferungen erkannt. Deshalb entstand 1992 das SPICE-Projekt, um eine Norm für Prozessbewertungen zu erarbeiten. SPICE stand zunächst für „Software Process Improvement and Capability Evaluation“. Da aber die Übersetzung von „Capability Evaluation“ im Französischen eine andere Bedeutung bekommen hätte, wurde „Software Process Improvement and Capability dEtermination“ gewählt und die Abkürzung belassen. [HDHM06]

Es dauerte bis 1998 bis der „Technical Report“ ISO/IEC TR 15504 als Vorstufe zum Internationalen Standard (IS) veröffentlicht wurde. 2006 hat der letzte Teil der Norm IS-Standard erreicht. Der Name des Projektes SPICE wird inzwischen als Synonym für das Modell verwendet.

Die folgende Abbildung 2.3 zeigt die Entwicklung von SPICE im Vergleich zu den anderen bereits erwähnten Reifegradmodellen.

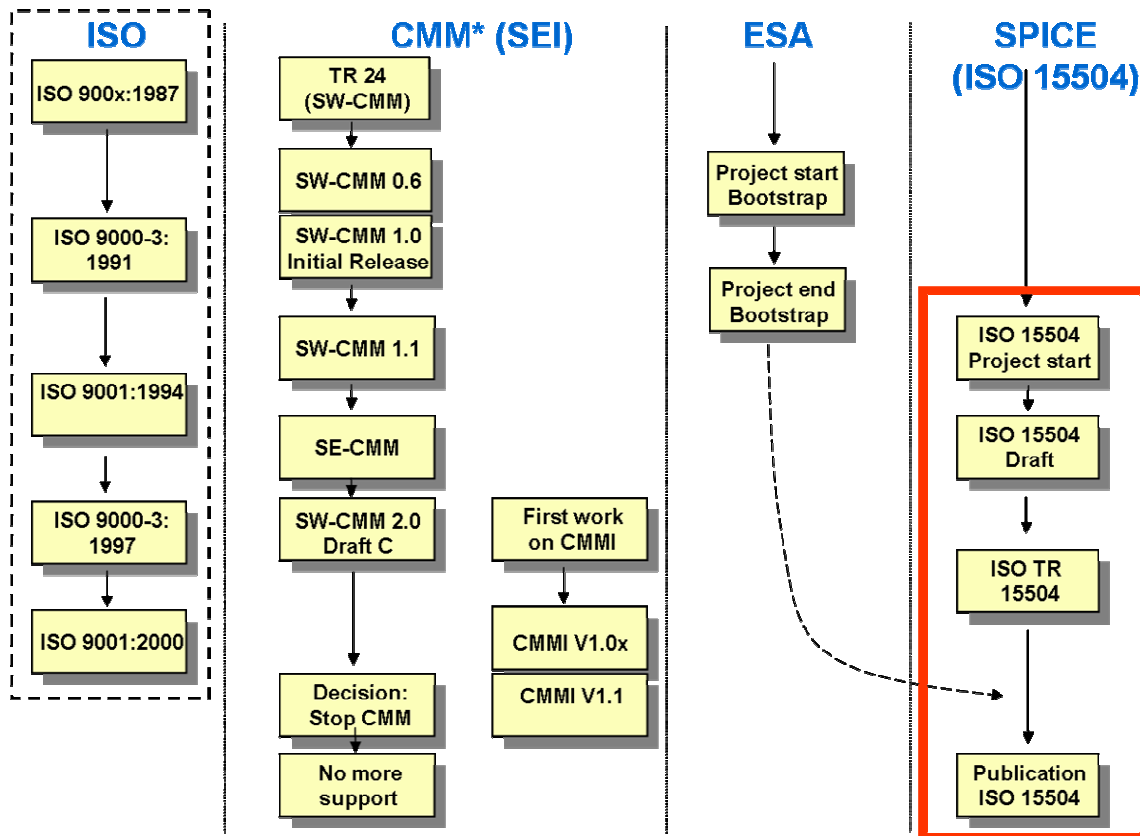


Abbildung 2.3: Übersicht der Reifegradmodelle (u.a. nach [Balz98])

Aufbau

Im Vergleich zum TR besteht die Norm ISO/IEC 15504 nicht mehr aus neun Teilen, sondern aus fünf. Nur der zweite Teil ist normativ, also verpflichtend für SPICE-konforme Modelle vorgeschrieben. Die anderen vier Teile haben mehr informativen und interpretativen Charakter [ISO05]:

- Teil 1: Konzepte und Vokabular
Allgemeine Einführung und Glossar
- Teil 2: Durchführung eines Assessments
Mindestanforderungen an ein Assessment, um konsistente Bewertungen zu erhalten
- Teil 3: Leitfaden zur Durchführung von Assessments
Leitfaden zur Interpretation von Teil 2
- Teil 4: Leitfaden zur Nutzung bei Prozessverbesserung und Prozessbewertung
Einsatzmöglichkeiten der Norm innerhalb einer Organisation
- Teil 5: Ein exemplarisches Prozess-Assessment-Modell
Beispiel eines PAM basierend auf ISO/IEC 12207 Amendment 1 & 2 nach den Anforderungen aus Teil 2

Als nächstes ist Teil 6 geplant, der ein beispielhaftes Assessment-Modell für Systementwicklungsprozesse basierend auf ISO/IEC 15288 enthalten wird.

Außerdem soll Teil 7 erstellt werden, mit dem es möglich sein wird einen Reifegrad für eine ganze Organisation zu vergeben, wie es mit dem CMMI Stufenmodell der Fall ist.

Konzept

Der Aufbau von ISO/IEC 15504 ist in Abbildung 2.4 zu sehen. Zentrales Element bei SPICE ist das Prozess-Assessment-Modell, das sich aus dem Bewertungsrahmen und dem Prozess-Referenzmodell zusammensetzt. Die einzelnen Komponenten werden nun im Folgenden genauer erklärt.

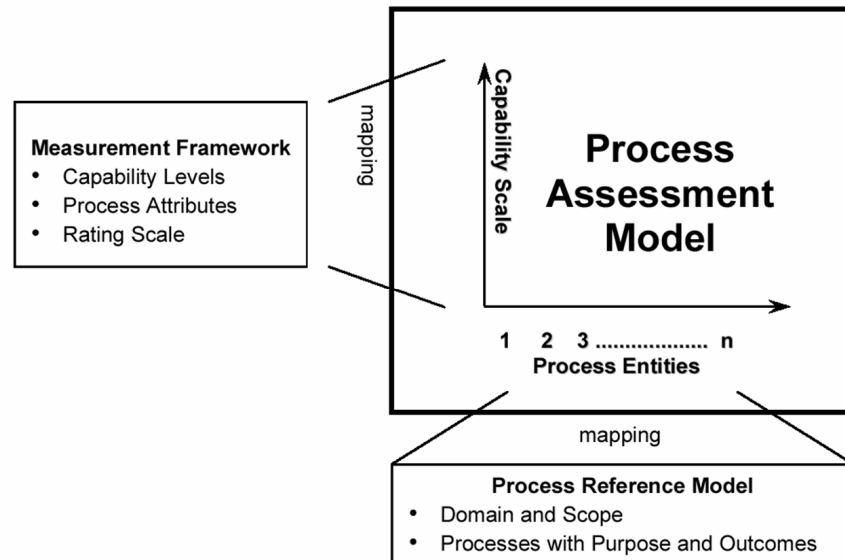


Abbildung 2.4: ISO/IEC 15504 im Überblick (aus [ISO05])

Prozess-Referenzmodell

Ein Prozess-Referenzmodell (PRM) beschreibt eine Menge von Prozessen jeweils mithilfe eines Prozesszwecks (*purpose*) und erwarteten Ergebnissen (*outcomes*). Die Prozessbeschreibungen müssen eindeutig sein, so dass es keine zwei Prozesse mit dem gleichen Namen oder den gleichen Ergebnissen gibt. Das PRM enthält die Anwendungsdomäne (*domain*) und den vorgesehenen Nutzungskontext (*scope*), außerdem müssen die Beziehungen zwischen den definierten Prozessen enthalten sein.

Bewertungsrahmen

Mit dem Bewertungsrahmen (*measurement framework*) lässt sich der Fähigkeitsgrad von Prozessen bestimmen. Im Gegensatz zu CMMI hat SPICE nur eine kontinuierliche Darstellung, das heißt der Reifegrad einer Organisation wird damit nicht bewertet.

Es werden sechs Fähigkeitsstufen (*capability level*) beschrieben, die aufeinander aufbauen und nicht übersprungen werden können. Die Fähigkeitsstufe leitet sich ab von der Bewertung von neun Prozessattributen (*process attributes, PA*), die den Stufen zugeordnet werden können, wie in der folgenden Abbildung 2.5 im linken Teil ersichtlich wird. Diese Attribute repräsentieren generische Fähigkeiten von Prozessen.

Je nach Erfüllung der Anforderungen können vier Bewertungen (*ratings*) erreicht werden:

- N: Nicht erfüllt (not achieved): 0-15%
- P: Teilweise erfüllt (partially achieved): 16-50%
- L: Überwiegend erfüllt (largely achieved): 51-85%
- F: Vollständig erfüllt (fully achieved): 86-100%

Damit eine Fähigkeitsstufe als erreicht gilt, müssen die Prozessattribute dieser Stufe mindestens mit L und die der darunter liegenden Stufen mit F bewertet sein [HDHM06].

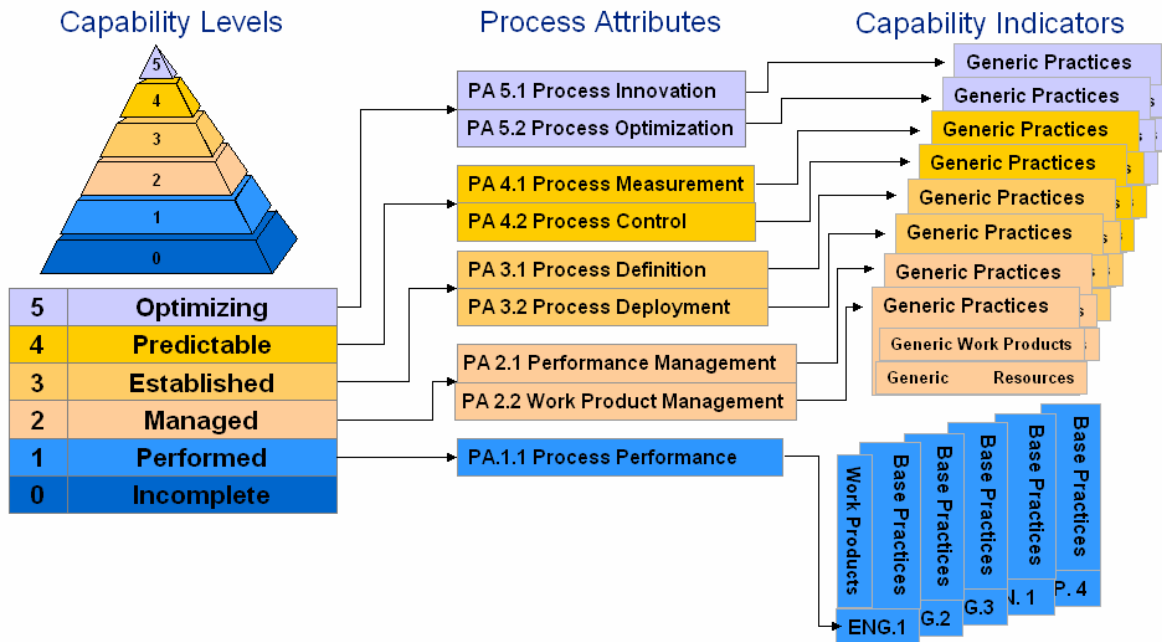


Abbildung 2.5: Fähigkeitsstufen und Prozessattribute bei SPICE (nach [ISO05])

Prozess-Assessment-Modell

Das Prozess-Assessment-Modell (PAM) bildet nun die Verknüpfung zwischen Prozess-Referenzmodell und Bewertungsrahmen. Ein PAM bezieht sich auf ein oder mehrere PRM, zu einem PRM können auch mehrere PAM existieren [HDHM06].

Das PAM ist die Grundlage für ein Assessment und ist in zwei Dimensionen organisiert: Die Prozessdimension enthält für alle relevanten Prozesse die Indikatoren, die sich auf das Prozess-Referenzmodell beziehen. Die Reifegraddimension enthält für alle relevanten Prozessattribute die Indikatoren zur Bewertung der Prozessreife. Die Indikatoren liefern den Nachweis zur Einstufung der Reife. In Abbildung 2.4 werden die beiden Dimensionen durch die „Process Entities“ auf der horizontalen und die „Capability Scale“ auf der vertikalen Achse im Diagramm dargestellt.

Im exemplarischen Prozess-Assessment-Modell basierend auf ISO/IEC 12207 im Teil 5 der SPICE-Norm werden verschiedene Indikatoren unterschieden. „Process Performance Indicators“ (PPI) werden für Level 1 benutzt. Hier gibt es Basispraktiken (*base practices* - BP) und Arbeitsprodukte (*work products* - WP). Die „Process Capability Indicators“ (PCI) werden eingeteilt nach Generischen Praktiken (*generic practices* - GP), Generischen Ressourcen (*generic resource* - GR) und Generischen Arbeitsprodukten (*generic work products* - GWP), vgl. Abbildung 2.5, rechter Teil.

Nachdem dieser Teil der Norm als Anleitung für weitere Prozess-Assessment-Modelle dient, werden diese Indikatoren für viele Assessments im Softwarebereich übernommen. [Isel05]

Assessment-Durchführung

Das Assessment ist das zentrale Mittel, um den Reifegrad eines Prozesses zu bestimmen und Prozessverbesserung zu ermöglichen, wie Abbildung 2.6 zeigt.

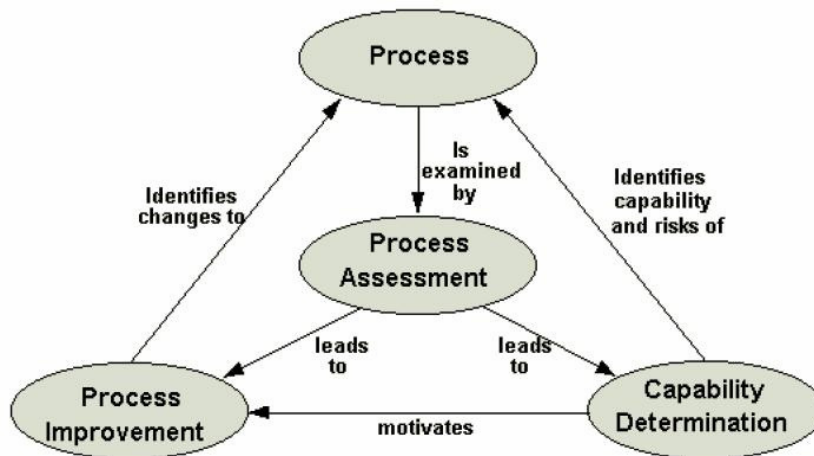


Abbildung 2.6: Die zentrale Rolle von Prozess-Assessments (nach [ISO05], [Balz98])

In ISO/IEC 15504 wird außerdem genau beschrieben, wie die Durchführung eines Assessments aussehen und dokumentiert werden muss. Die folgenden Aktivitäten müssen mindestens enthalten sein:

- Planung
- Datensammlung
- Datenvalidierung
- Bewertung der Prozessattribute
- Berichtswesen.

In den Eingaben (*input*) für das Assessment werden Zweck und Umfang bestimmt, also welche Prozesse bis zu welchem Fähigkeitsgrad untersucht werden sollen. In der Ausgabe (*output*) wird das Ergebnis präsentiert.

Abbildung 2.7 zeigt noch mal die wichtigsten Eigenschaften eines SPICE-Assessments im Überblick:

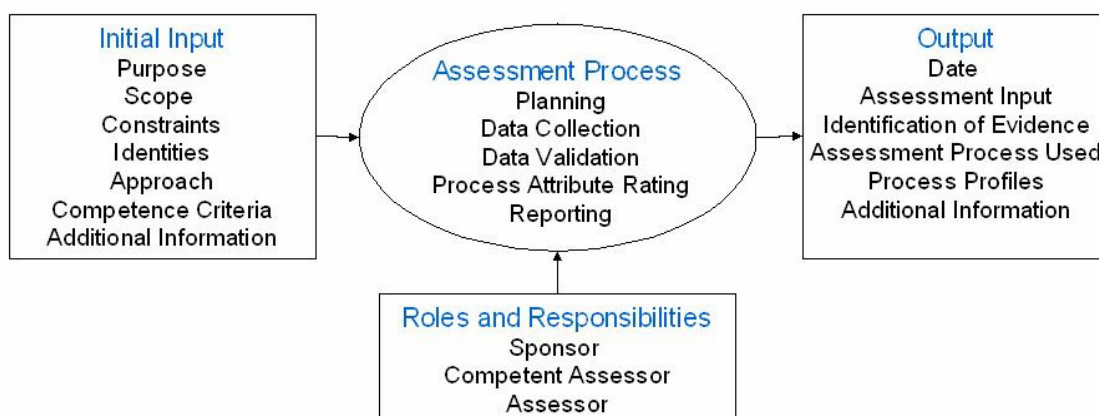


Abbildung 2.7: Bestandteile eines SPICE-Assessments (nach [ISO05])

Ausblick

Während CMMI mehr im asiatischen und nordamerikanischen Raum Verbreitung findet, hat SPICE den größeren Marktanteil in Europa. Im Gegensatz zu CMMI ist SPICE allerdings inzwischen nicht nur ein proprietärer Standard - die Eigentumsrechte von CMMI

liegen beim SEI - sondern ein ISO-Standard zu dem CMMI kompatibel ist. [WeRa06] [Tarn06]

Das Modell ist flexibel anpassbar und wurde mittlerweile in verschiedenen Branchen wie der Raumfahrt (SPICE4SPACE) oder der Medizintechnik (MediSPICE) verwendet. Eine besonders große Rolle spielt SPICE in der Automobilindustrie durch Automotive SPICE. Die großen Automobilhersteller haben sich in der Herstellerinitiative Software (HIS) organisiert und verlangen von ihren Zulieferern SPICE-Bewertungen. [Tarn06] Nicht zuletzt dadurch hat das Reifegradmodell in den vergangenen Jahren stark an Verbreitung zugenommen.

Auf dem Weg zum verabschiedeten Standard wurde der normative Anteil immer abstrakter formuliert, so dass SPICE heute ein sehr schlankes Modell ist und eher ein Rahmenwerk für Prozess-Assessments als ein Reifegradmodell darstellt [Isel05]. Deshalb ist es auch möglich ein Prozess-Assessment-Modell für TPI[®] nach SPICE zu entwickeln, was Thema dieser Arbeit ist.

In der Literatur wird der Begriff SPICE oft als Synonym für das exemplarische Prozess-Assessment-Modell (nach ISO 12207) in Teil 5 von ISO/IEC 15504 benutzt. Eigentlich steht SPICE aber für die allgemein formulierten, verpflichtenden Anforderungen an Modelle in Teil 2 der Norm. Deshalb wird dieser Begriff auch im Folgenden in diesem Sinne, also gleichbedeutend zu ISO/IEC 15504, verwendet werden.

2.2 Testprozessverbesserungsmodelle

2.2.1 Testen in der Software-Entwicklung

Die Definition des Testens in der Software-Entwicklung, hat sich im Laufe der Zeit stark gewandelt. So ist Testen nicht mehr nur der Prozess zum Nachweis, dass ein Softwaresystem funktioniert (nach Hetzel, 1973) und hat auch nicht den alleinigen Zweck Fehler zu finden (Myers, 1979). Testen bedeutet auch die Softwarequalität zu messen (Hetzel, 1984). [Veen02]

Eine jüngere Erklärung versteht unter Testen den „Prozess des Planens, der Vorbereitung und der Messung, mit dem Ziel, die Eigenschaften eines IT-Systems festzustellen und den Unterschied zwischen dem tatsächlichen und dem erforderlichen Zustand aufzuzeigen.“ (Pol et al., 1995) [PKSp02].

Testen ist ein Prozess, der zeitgleich mit dem Projektstart beginnen sollte. Die Erfahrung hat gezeigt, dass es umso teurer wird, je später Fehler in der Software entdeckt werden. Abbildung 2.8 zeigt die Fehlerkosten in Abhängigkeit des Zeitpunkts, zu dem der Fehler entdeckt wird. So sind Mängel, die man in der Anforderungsphase findet mit wesentlich weniger Folgekosten verbunden, als Fehler, die erst nach Auslieferung des Produktes entdeckt werden und Schaden bei den Benutzern und auch am Image des Herstellers hervorrufen können.

Qualität lässt sich in ein Produkt nicht „hineintesten“, vielmehr ist Testen eine qualitätssichernde Maßnahme. Man kann mit statischen Testtechniken (wie z.B. Reviews) und dynamischen Testmethoden Fehler finden. Außerdem kann damit die Qualität eines Systems beurteilt werden. Testen ermöglicht eine Verbesserung der Software. Die eigentliche Fehlerbehebung, das so genannte Debugging, ist Aufgabe der Implementierung.

Für detaillierte Informationen zum Thema Softwaretest sei an dieser Stelle auf weitere Literatur wie [Veen02], [SpLi03] oder [SRWL06] hingewiesen.

Das Testen kommt in der Softwareentwicklung oft zu kurz, wird schlecht geplant und unsystematisch durchgeführt. Ebenso wie beim gesamten Entwicklungsprozess muss eine gute Relation zwischen Zeit, Kosten und Qualität erreicht werden. [PKSp02]

Deshalb ist eine Reihe von Modellen zur Testprozessverbesserung entstanden, die im Folgenden kurz vorgestellt werden.

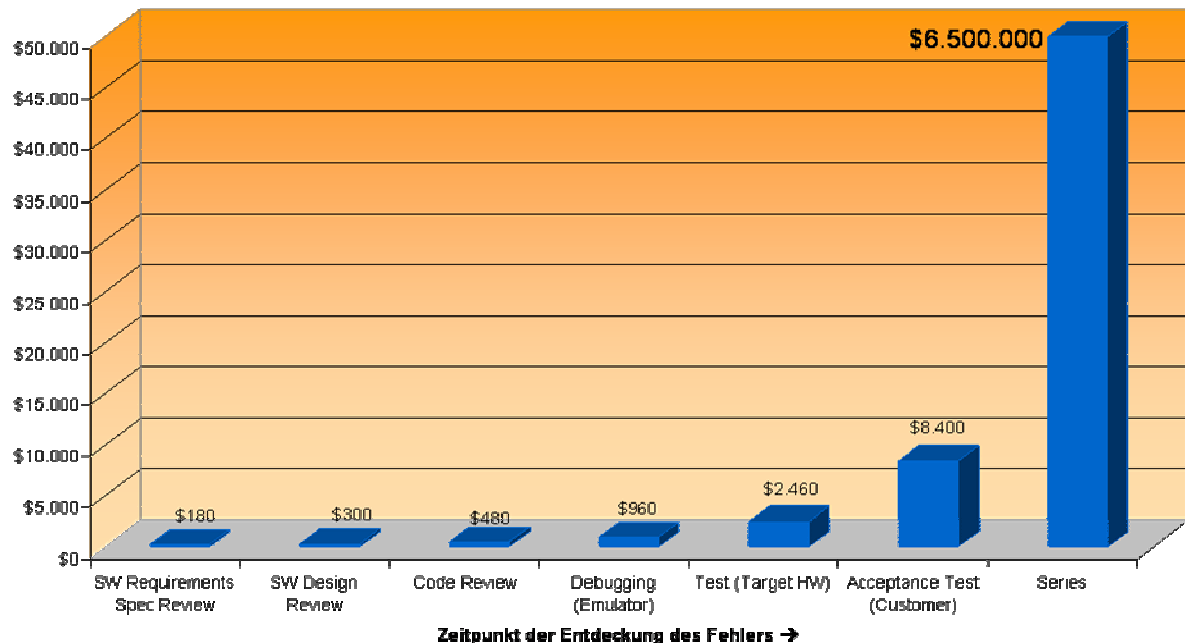


Abbildung 2.8: Fehlerkosten in verschiedenen Projektphasen (nach [Balz98])

2.2.2 Übersicht

Die meisten Modelle zur Testprozessverbesserung sind Mitte der neunziger Jahre entstanden. Hier ein kurzer Überblick der wichtigsten Modelle.

Maturity Model for Automated Software Testing (MMAST)

Dieses Modell wurde 1994 von Mitchel H. Krause speziell für die Hersteller von Medizingeräten entwickelt, die ihren Testprozess automatisieren wollten [Swin00]. Somit liegt der Fokus hier auf der Testautomatisierung und nicht auf dem gesamten Testprozess. In diesem Modell werden keine Verbesserungsvorschläge und Assessmentanweisungen gegeben.

Testing Assessment Programme (TAP)

1995 wurde dieses Modell von „Software Futures Ltd. and IE Testing Consultancy Ltd.“ entworfen und basiert auf CMM. Es werden vier Kernbereiche untersucht, nämlich Ziele, Personen, Management und Methoden. Spezielle Techniken, Werkzeuge oder Vorgehensweisen werden nicht empfohlen, detaillierte Informationen sind ebenfalls nicht vorhanden. [Swin00]

Testing Capability Maturity Model (TCMM)

TCMM wurde 1996 von Susan M. Burgess und Rodger D. Drabick konzipiert. Das Modell ist ebenfalls an CMM und dessen fünf Reifegradstufen angelehnt. Weitere Inhalte, wie der dazugehörige Fragebogen, sind allerdings nach [Swin00] nicht verfügbar.

Testability Support Model (TSM)

David Gelperin hat 1996 das Testability Support Model entwickelt, das auch als Testability Maturity Model bekannt ist. Es besteht nur aus einem Reifegradmodell und beschreibt drei Stufen. Mit TSM werden Möglichkeiten zur Verbesserung der Testbarkeit eines Systems aufgezeigt. [Swin00]

Test Organization Maturity Model (TOM)

Die Firma Systeme Evolutif (heute: gerrard consulting) hat TOM konzipiert. Ausgangspunkt war die Erfahrung, dass die Hauptprobleme in Projekten organisatorischer Natur sind und nicht etwa technische Schwierigkeiten. Der Fokus liegt hier auf Managementexpectationen und der Motivation von Mitarbeitern, geht aber nicht auf alle Bereiche des Testens ein.

Mithilfe eines Fragebogens können Verbesserungsvorschläge ermittelt werden. Dabei werden zunächst Prioritäten vergeben, je nachdem welche Aspekte im Test als besonders wichtig erachtet werden. Dann wird mit 20 so genannten Symptomen der Entwicklungsgrad eines Bereiches festgestellt und danach Verbesserungsvorschläge ermittelt [Gerr07]. Ein spezielles Reifegradmodell existiert nicht. Dieses Modell wurde schon in verschiedenen Branchen verwendet. [Swin00]

Test Improvement Model (TIM)

TIM wurde 1996 von Ericson, Subotic und Ursing entworfen. Das Modell enthält ein Reifegradmodell mit fünf Stufen und eine Beschreibung zum Assessmentvorgehen und baut auf CMM und Gelperins TSM auf. Hier werden fünf Kernbereiche beschrieben:

- Organisation
- Planung und Tracking
- „Testware“
- Testfälle
- Reviews.

Für jeden Kernbereich werden Aktivitäten für jede Reifegradstufe vorgegeben. [ErSU96] Beim Testing Improvement Model werden vor allem in den höheren Reifegradstufen wenig detaillierte Informationen angegeben, Fehlervermeidung und Qualitätskontrolle werden gar nicht erwähnt. [Swin00]

Testing Maturity Model (TMM)

1996 wurde am Illinois Institute of Technology von Burnstein et al. das Testing Maturity Model als Ergänzung des Capability Maturity Model entwickelt, da CMM nur wenig auf den Testprozess eingeht.

Die Definition der Reifegradstufen in TMM basiert auf dem Testmodell von Gelperin und Hetzel von 1988. Darin wurden vier historische Phasen des Testens von 1950 bis 1980 identifiziert: zunächst gab es die reine Fehlerbehebung ohne eine Trennung von Test und Debugging, dann kam die Fehlerfindung mit einer geplanten Testphase, anschließend fand eine Integration des Testens in den Softwarelebenszyklus statt und schließlich ging es um

die Optimierung des Testprozesses mit Messen und Kontrolle der Qualität. Aufgrund der Annahme, dass jede Organisation diese historischen Phasen durchläuft, wird nach TMM die entsprechende Phase identifiziert und der Testreifegrad der Organisation festgestellt. Im Gegensatz zu TIM ist TMM ein Stufenmodell, bei dem es für jede der fünf Reifegradstufen separate Prozesse mit Zielen und Aktivitäten gibt, d.h. ein Prozess ist nicht in allen Stufen definiert, sondern einer speziellen Stufe zugeordnet [BuSC96]. Die Übersicht in Tabelle 2.2 verdeutlicht dies.

TMM enthält ein Assessment-Modell mit Vorgehensweisen und Fragebogen. Weiterhin werden Beispiele, Vorlagen und empfohlene Literatur beschrieben. [Veen02]

	Maturity Level	Process areas
1	Initial	-
2	Definition	Test Policy and Goals Test Planning Test Techniques and Methods Test Environment
3	Integration	Test Organisation Test Training Program Test Life Cycle and Integration Control and Monitor
4	Management and Measurement	Peer Reviews Test Measurement Software Quality Evaluation
5	Optimisation	Defect Prevention Test Process Optimisation Quality Control

Tabelle 2.2: Die Reifegradstufen im TMM (nach [Veen02])

Das Testing Maturity Model deckt einen Großteil des Testprozesses ab, allerdings werden einige wichtige Bereiche nicht betrachtet. Das sind vor allem die Testumgebung, der Testarbeitsplatz, die Fehlerbehandlung, Fehler- und Testwaremanagement. [Swin00]

2.2.3 TPI[®]

Erfahrungen mit TMap

Das Testprozessverbesserungsmodell TPI[®] wurde 1996 von IQIIP entworfen (heute ist es eine eingetragene Marke von sogeti) [KoPo99]. Grundlage dafür waren die Erfahrungen mit der Testmethodik TMap (Test Management Approach), die ebenfalls von IQIIP im Jahr 1995 erstellt wurde.

Das strukturierte Testkonzept TMap besteht aus vier Pfeilern, diese sind:

- Phasenmodell mit Entwicklungszyklus und zeitlichem Ablauf der Testaktivitäten
- organisatorische Einbettung
- Infrastruktur mit entsprechenden Werkzeugen
- benötigte Techniken.

Die Pfeiler beschreiben was wann, durch wen, wie, wo und womit getestet werden sollte [KoBa05]. Die Kenntnisse aus dem Praxiseinsatz von TMap bei der DV-Unterstützung von Geschäftsprozessen sind in TPI[®] eingeflossen. [PKSp02]

Überblick

TPI[®] bietet einen Bezugsrahmen, um mit Kontrollpunkten die Stärken und Schwächen des Testprozesses eines Unternehmens zu bestimmen und gezielt Verbesserungsvorschläge zu machen [PKSp02]. Einen Überblick des Modells gibt Abbildung 2.9:

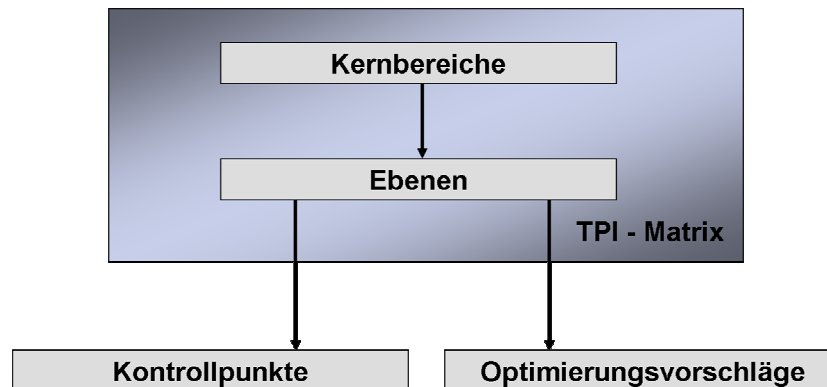


Abbildung 2.9: Aufbau des TPI-Modells (nach [PKSp02])

Kernbereiche

Das TPI[®]-Modell umfasst 20 Kernbereiche (*key areas*), die vor allem auf High-Level-Tests wie z. B. System- und Abnahmetest, abgestimmt sind. Für Low-Level-Tests (Modul-, Integrationstest) und Prüfen wurden gesonderte Bereiche gebildet.

Die 20 Kernbereiche enthalten Themen zu den vier Pfeilern von TMap. Zum Phasenmodell gibt es „Einsatz des Phasenmodells“ oder „Zeitpunkt der Beteiligung“. Bei den Techniken sind etwa „Statische Testtechniken“ und „Metriken“ zu finden. Es gibt Bereiche zur Infrastruktur, beispielsweise „Testumgebung“ oder „Testarbeitsplatz“ und zur Organisation sind „Kommunikation“ oder „Testwaremanagement“ vorhanden.

Somit werden alle Aspekte des Testprozesses beleuchtet.

Ebenen

Die 20 Kernbereiche sind in verschiedene Ebenen (*level*) A bis D unterteilt, wobei Ebene A die niedrigste ist. Um eine höhere Ebene zu erreichen, müssen auch die Anforderungen für die unteren erfüllt sein. Die Anzahl der Ebenen pro Bereich ist sehr unterschiedlich: Es können vier, aber auch nur eine vorhanden sein.

Die höheren Ebenen im TPI[®]-Modell zeichnen sich dadurch aus, dass eine Optimierung bei Kosten, Zeitbudget und Qualität des jeweiligen Teilbereiches stattfindet. Wichtig sind ein früherer Teststart, die Automatisierung und Professionalisierung des Testprozesses, aber auch eine bessere Kommunikation und vor allem eine projektübergreifende Testkonzeption.

Für den Kernbereich „Dokumentation der Abweichungen“ sind beispielsweise folgende Ebenen definiert:

- A: Interne Dokumentation der Abweichungen
- B: Umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten
- C: Dokumentation der Abweichungen wird im gesamten Projekt eingesetzt

Um eine objektive Bewertung des Testbereiches sicherzustellen, ist jede Ebene mit Kontrollpunkten versehen, die erfüllt werden müssen, um eine Ebene zu erreichen. Außerdem wird eine Vielzahl von Optimierungsvorschlägen gemacht, die die Verbesserung zu unterstützen.

Matrix

Nicht jeder Kernbereich ist zunächst gleich wichtig, zudem bestehen Abhängigkeiten zwischen den verschiedenen Bereichen und Ebenen. Diese Zusammenhänge werden in der so genannten TPI[®]-Matrix in Abbildung 2.10 verdeutlicht:

Kernbereich / Stufe	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A					B				C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A		B							
7 Metriken						A			B			C		D
8 Testautomatisierung					A			B			C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A				B						C		
12 Testfunktionen und Ausbildung				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A			B							C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
		beherrschbar					effizient					optimierend		

Abbildung 2.10: Die TPI[®]-Matrix (aus [PKSp02])

In dieser Matrix sind vertikal alle Kernbereiche aufgelistet und horizontal befinden sich 13 Entwicklungsstufen (*scales*), auf die die Ebenen verteilt sind. Auf der untersten Stufe 0, der Startstufe sind bei einem Bereich noch keine Anforderungen erfüllt. Die leeren Matrixfelder zeigen, dass das Erreichen eines Kernbereiches vom Entwicklungsgrad anderer Kernbereiche abhängt. So macht es keinen Sinn schon Projektmetriken (Kernbereich „Metriken“, Ebene A) einzuführen und auszuwerten, wenn noch gar keine Daten zu Abweichungen und Testaufwänden gesammelt und aufgeschrieben werden. Deshalb müssen zuerst die Kernbereiche „Berichterstattung“ und „Dokumentation der Abweichungen“ erfüllt sein.

In [PKSp02] wird zur Erklärung dieser Beziehungen der gesamte Testprozess sehr treffend mit einem Fitnessprogramm verglichen:

Die Kernbereiche entsprechen einzelnen Körperbereichen wie Arme, Beine etc. Die Ebenen bezeichnen verschiedene körperliche Zustände und Fitnessgrade, die anhand von Kontrollpunkten, wie mit der Ausführung von bestimmten Übungen, festgestellt werden können. Um nun in einer Disziplin wie Gewichtheben Verbesserungen zu erzielen, ist das kombinierte Zusammenspiel von mehreren Körperteilen nötig, ebenso wie bei der

Disziplin Testen viele Faktoren einfließen. So macht es keinen Sinn nur eine einseitige Optimierung der Armkraft zu trainieren, wenn die Beine dabei völlig vernachlässigt werden.

Genauso ist es unsinnig in einem unstrukturierten und unreifen Testprozess bereits eine umfassende Testautomatisierung anzustreben. Wenn noch grundlegende Teststrategien und –Methodiken nicht erfüllt sind, kann die Konzentration auf Testtools sogar kontraproduktiv sein. Oder anders ausgedrückt: „Automating chaos leads to faster chaos.“ (Graham et al, 1996). [PKSp02]

Mithilfe der Matrix kann nun der Leistungstand der einzelnen (Körper-)Bereiche erkannt und ein individuelles „Trainingsprogramm“ mit spezifischen Übungen und Verbesserungsmaßnahmen zusammengestellt werden.

Die verschiedenen Entwicklungsstufen werden zusätzlich noch in drei Kategorien (*categories*) eingeteilt, die die Reife des Testprozesses beschreiben. Die Stufen 1 bis 5 werden der Kategorie „beherrschbar“ zugeteilt. Hier geht es um die Verwaltung eines kontrollierbaren Testprozesses.

Die Kategorie „effizient“ umfasst die Stufen 6 bis 10 und legt den Schwerpunkt auf Abstimmung, Integration und Automatisierung.

Die Stufen 11 bis 13 bilden die Kategorie „optimierend“ mit ständiger Verbesserung und Anpassung des Testprozesses an veränderte Verhältnisse. [PKSp02]

Einsatz

Der Einsatz des Modells Testprozessoptimierung sollte, wie in Abbildung 2.11 gezeigt, damit beginnen Bewusstsein zu schaffen, dass es sinnvoll und nutzbringend ist den Prozess zu verbessern. Danach sollte bestimmt werden welche Ziele angestrebt und welche Vorgehensweise damit verbunden sind. Beispielsweise kann der Hauptfokus auf besserer Testabdeckung liegen oder auf mehr Informationen über die Qualität des Testobjektes.

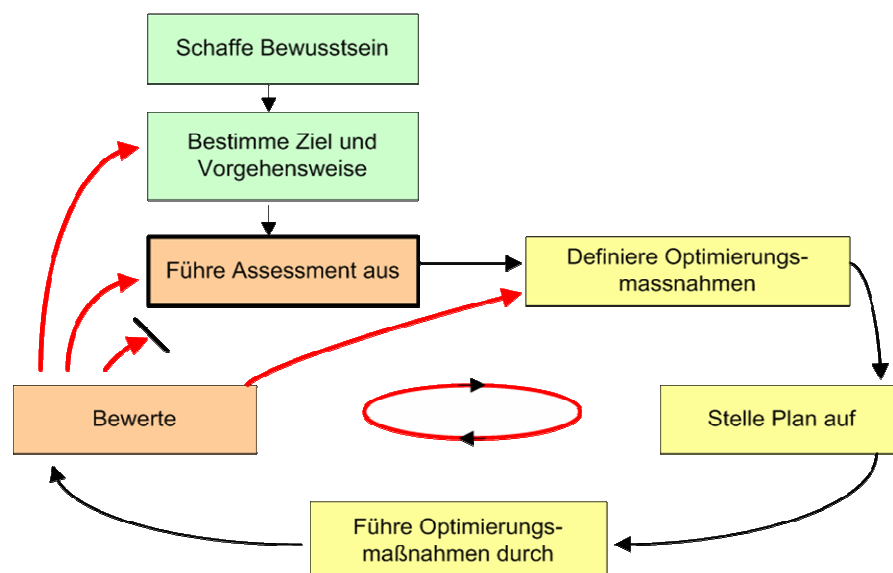


Abbildung 2.11: Der Änderungsprozess bei TPI® (nach [PKSp02])

Beim Assessment wird zunächst der Ist-Zustand der einzelnen Kernbereiche festgestellt. Das könnte z.B. wie im folgenden Ausschnitt in Abbildung 2.12 aussehen. Zur Vereinfachung werden hier nur einige Bereiche herausgegriffen.

Kernbereich / Stufe	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Teststrategie		A					B				C		D	
Einsatz des Phasenmodells		A			B									
Zeitpunkt der Beteiligung			A				B				C		D	
Test-Spezifikationstechniken		A		B										
Metriken						A			B			C		D

Abbildung 2.12: Beispiel des Ist-Zustands einiger Kernbereiche (aus [PKSp02])

Obwohl viel Zeit und Geld in das Testen investiert wird, ist das Ergebnis nicht zufriedenstellend. Von links nach rechts können nun in der TPI[®]-Matrix die zu verbessernden Kernbereiche ermittelt werden. Dabei ist ein schrittweises Vorgehen zu empfehlen, bei dem nicht alle Bereiche auf einmal angepackt werden.

Kernbereich / Stufe	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Teststrategie		A					B				C		D	
Einsatz des Phasenmodells		A			B									
Zeitpunkt der Beteiligung			A				B				C		D	
Test-Spezifikationstechniken		A		B										
Metriken						A			B			C		D

Abbildung 2.13: Beispiel eines Soll-Zustands einiger Kernbereiche (aus [PKSp02])

Im vorliegenden Beispiel sollte man deshalb erst wie in obiger Abbildung 2.13 gezeigt, bei der Teststrategie Ebene A anstreben, um anhand der Risiken bewusst zu entscheiden, was wie intensiv mit welchen Testspezifikationstechniken (Ebene B) getestet wird. Mit der Einführung von Metriken macht es Sinn noch zu warten, bis die anderen Maßnahmen Früchte getragen haben. Außerdem macht ein früher Zeitpunkt der Beteiligung nur Sinn, wenn diese Zeit mit der Vorbereitung von Tests gemäß dem Einsatz des Phasenmodells gut genutzt wird [SRWL06]. Das vorliegende Bild zeigt den gewünschten Sollzustand.

Nachdem ein genauer Plan zur Durchführung erstellt wurde, bei dem auch Zuständigkeiten und Kosten festgehalten werden, findet die Umsetzung statt.

Der Erfolg der Veränderungen wird dann bei einer Bewertung der Ausführungen ermittelt und der nächste Zyklus bei der Testprozessverbesserung mit neuen oder veränderten Zielen kann beginnen. [PKSp02]

Eine Umfrage von 2002 [Koom02] zeigt, dass TPI[®] weltweit eingesetzt wird. Die meisten Teilnehmer an der Umfrage konnten durch die Verwendung von TPI[®] eine Verbesserung der Testqualität feststellen. Außerdem wurden höhere Testabdeckung, bessere Kontrolle des Testprozesses und ein gutes Kosten-Nutzen-Verhältnis beim Einsatz des TPI[®]-Modells beobachtet. [Koom02]

2.3 Werkzeugunterstützung mit project>kit

Das webbasierte Projektportal project>kit der method park Software AG bietet Zugriff auf die Informationen, die in einem Entwicklungsprojekt benötigt werden. Das Projekt- und Prozessmanagementwerkzeug ermöglicht dabei eine flexible Gestaltung, Integration und Anpassung der Entwicklungsprozesse [Meth05]. Grafiken, Diagramme oder Vorlagen sind einfach als Hilfsmittel zu speichern und Inhalte leicht einzugeben. Die verteilte

Webapplikation ist somit gut geeignet für das Informations- und Wissensmanagement in Projekten.

Das System besteht aus mehreren aufeinander abgestimmten Komponenten [Mroz06]. Der „Infocenter“ ist eine Informationsplattform zur Kommunikation der Projektmitglieder, „Auf einen Blick“ und „Downloadcenter“ unterstützen den schnellen Zugriff auf Vorlagen und Projektdokumente und der „Issue Tracker“ dient zum Fehler- und Änderungsmanagement.

Die zentrale Komponente ist die „Process Workbench“, in der die Beschreibung und Verwaltung der Entwicklungsprozesse zu finden ist. Um Prozessmodelle zu erstellen, werden verschiedene Möglichkeiten für unterschiedliche Projekttypen angeboten. So können vordefinierte Prozessmodelle verwendet, angepasst (getailort) oder auch von einem anderen project>kit System importiert werden [Meth05].

Perform Gap-Analysis

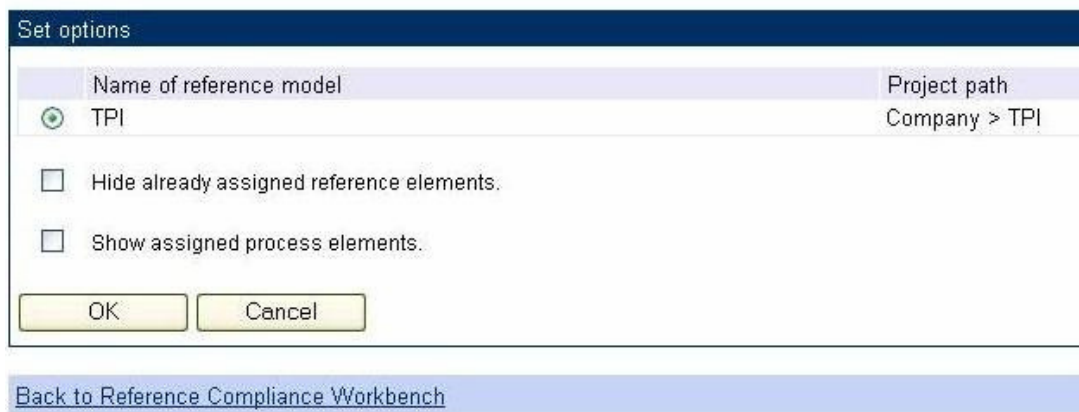


Abbildung 2.14: Die Reference Compliance Workbench von project>kit

Seit der Version 3.0 verfügt das Werkzeug auch über die hier in Abbildung 2.14 gezeigte so genannte „Reference Compliance Workbench“. Hier sind die folgenden Funktionen vorhanden:

- Abbildung von Prozessbeschreibungen auf beliebige Referenzmodelle wie CMMI, ISO9001 oder SPICE.
- Automatisierte Lücken- (Gap-) Analysen zur Konformität des jeweiligen Prozesses zum verwendeten Referenzmodell.
- Konsistenzprüfung der Prozessbeschreibung anhand definierbarer Bedingungen [Meth05].

Diese Eigenschaften sind bei der Vorbereitung und Durchführung von Assessments hilfreich.

Die Prozessmodelle sind flexibel definierbar, um die Vielzahl der Prozesse beschreiben zu können. So gibt es aktivitätsgetriebene oder artefaktgetriebene Modelle, die aus unterschiedlichen Prozesselementtypen bestehen, wie etwa Rollen, Phasen, Aktivitäten, Dokumente (Artefakte), Werkzeuge und Methoden. Diese Prozesselementtypen stehen in Beziehung zueinander, zum Beispiel ist eine bestimmte Rolle verantwortlich für die

Erstellung eines Dokumentes mithilfe einer bestimmten Methode in einer bestimmten Phase. Abbildung 2.15 zeigt mögliche Assoziationen, die hierbei auftreten können.

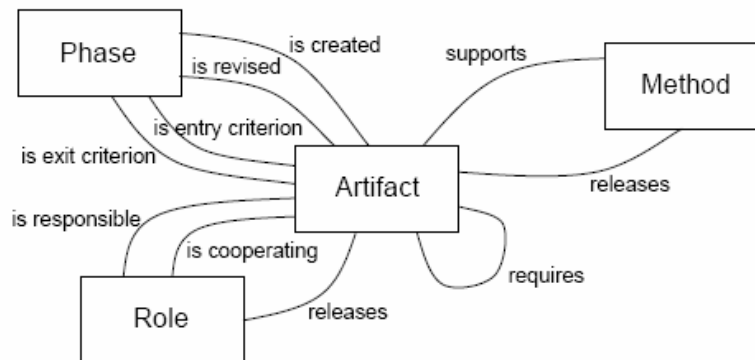


Abbildung 2.15: Assoziationen von Prozesselementtypen (aus [Meth06])

Die Beschreibung solcher Prozessstrukturen wird Metamodell genannt. Die Metamodelle werden in `project>kit` in der Datei „PkitProcesses.xml“ definiert und bieten ein sehr flexibles Konzept mit vielen Freiheitsgraden bei Prozessdefinitionen. Hier nur als Beispiel ein kleiner Auszug aus einem solchen Metamodell:

```

<pkit-process>
<!-- Beschreibung Metamodell -->
  <processtype>
    <element>
      <model><!-- Assoziationen --></model>
      <view><!-- Layout --></view>
    </element>
  </processtype>
</pkit-process>

```

Der Aufbau der xml-Datei erfolgt nach dem Model-View-Controller-Muster. Das heißt, dass im Model-Teil die Beziehungen (*associations*) der Prozesselementtypen beschrieben werden und im View-Teil die HTML-Darstellung enthalten ist [Meth06].

Eine umfangreiche Erklärung der Funktionsweise des Metamodells ist im `method park` Metamodeling Guide [Meth06] enthalten.

Im Rahmen dieser Arbeit wurde das Metamodell von `project>kit` für das erstellte Referenzmodell von TPI[®] nach SPICE angepasst. Im nächsten Kapitel wird die konkrete Umsetzung dazu genauer erläutert.

3 Umsetzung und Ergebnisse

Nachdem die grundlegenden Prinzipien der Modelle vorgestellt wurden, wird nun die konkrete Umsetzung der Aufgabenstellung gezeigt. Die Entstehung des Prozess-Assessment-Modells wird Schritt für Schritt erklärt und die Ergebnisse zum besseren Verständnis dokumentiert. Anschließend wird die praktische Umsetzung in project>kit erläutert.

3.1 Vergleich von TPI[®] und SPICE

Um ein Prozess-Assessment-Modell für TPI[®] nach SPICE zu erstellen, ist es zunächst nötig diese beiden Standards konzeptionell und inhaltlich zu vergleichen und die Unterschiede herauszustellen.

In der Literatur ist oft von SPICE die Rede, doch tatsächlich ist Teil 5 von ISO/IEC 15504 gemeint, also das Beispielreferenzmodell auf Basis von ISO/IEC 12207, einem Prozessmodell zur Software-Entwicklung [Wiki07a].

So wird auch bei der Studie von Q-Labs [Gros04] zum inhaltlichen Vergleich zwischen „ISO 15540 und TPI[®] im Automotive Umfeld“ das Beispielmodell zu ISO/IEC 12207 referenziert.

In dieser Studie wurden einzelne Prozesse von SPICE, Teil 5, herausgegriffen und untersucht wie der Inhalt durch TPI[®] abgedeckt wird. Dabei wurden die einzelnen Basispraktiken von SPICE den Kontrollpunkten von TPI[®] zugeordnet. Beispielsweise wurden für den Prozess Software-Erstellung die entsprechenden Anweisungen in verschiedensten Kernbereichen von TPI[®] gefunden, wie etwa Prüfen, Low-Level-Tests, Berichterstattung oder Fehlermanagement.

Für die untersuchten Prozesse von SPICE wurde keine, wenig oder volle Abdeckung durch TPI[®] ermittelt. In den anderen Fällen war der Inhalt von TPI[®] nicht in den SPICE-Prozessen enthalten.

Unabhängig von dieser Arbeit von Q-Labs lässt sich folgendes feststellen:

Es gibt eine Reihe von Prozessen im Modell für ISO 12207 nach SPICE, die sich mit Testen befassen. Diese sind vor allem in den Prozessen für Engineering (mit ENG gekennzeichnet) und den unterstützenden Prozessbereichen (Abkürzung SUP für support) zu finden. Im Prozess „ENG.6 Softwareerstellung“ werden Modultests und statische Testmethoden wie z.B. Codereviews erwähnt. „ENG.7 Softwareintegration“ fordert Tests für integrierte Softwareeinheiten, das integrierte Softwareprodukt wird in „ENG.8 Softwaretest“ geprüft. Entsprechend sind in „ENG.9 Systemintegration“ und „ENG.10 Systemtest“ die Tests für Systemelemente und das integrierte System enthalten.

Bei den unterstützenden Prozessen finden sich „SUP.1 Qualitätssicherung“, „SUP.2 Verifikation“, „SUP.3 Validierung“ und „SUP.4 Gemeinsame Reviews“.

Insgesamt ist SPICE im Testbereich wenig detailliert und bietet auch keine konkreten Verbesserungsmöglichkeiten an. Der Modultest findet sich etwa nur kurz bei „ENG.6 Softwareerstellung“, die Automatisierung wird höchstens indirekt bei der Erstellung von Regressionsteststrategien erwähnt. Die Testbarkeit dagegen wird bei der Erstellung von

Software- und Systemanforderungen (ENG.2, ENG.4) und beim Softwaredesign in ENG.5 betont.

Andere Testanforderungen wie die Testumgebung oder der Testarbeitsplatz werden im untersten Level gar nicht beschrieben bzw. gefordert, sondern erst in den höheren Levels erwähnt.

Es wäre somit höchstens möglich, das vorhandene SPICE-Prozess-Assessment-Modell nach ISO 12207 zum Teil mit den Inhalten von TPI[®] zum Softwaretest zu ergänzen. Mit dieser Arbeit soll aber ein eigenes Prozess-Assessment-Modell nach SPICE explizit für Testprozessverbesserung erstellt werden und zwar für das detaillierte und praxisbewährte TPI[®]-Modell.

Zunächst ein allgemeiner Vergleich der beiden Modelle:

SPICE	TPI [®]
Allgemeines Rahmenwerk mit Anforderungen für die Erstellung eines beliebigen Prozess-Assessment-Modells.	Referenzmodell mit Fokus auf Testprozessverbesserung.
Internationaler Standard, ISO-Norm.	Berücksichtigt bestehende Standards nicht, aber bewährt und anerkannt in der Praxis.
Hauptsächlich in Europa verbreitet.	Weltweit verbreitet, Schwerpunkt Europa.
Wachsende Durchdringung verschiedener Industriedomänen (Automotive, Raumfahrt, Medizintechnik etc.)	Angewandt in verschiedenen Branchen (Automotive, Verwaltung, etc.).
Modell zur Ermittlung des Ist-Zustandes mit Stärken und Schwächen von verschiedensten Prozessen.	Modell zur Ermittlung des Ist-Zustandes mit Stärken und Schwächen von Testprozessen.
Ermöglicht Verbesserungsmaßnahmen.	Konkrete, detaillierte Vorschläge und Beispiele zu Verbesserungsmöglichkeiten.

Tabelle 3.1: Vergleich der Modelle SPICE und TPI[®]

Die zwei Modelle sind somit beide weit verbreitet und werden vielfältig eingesetzt, wobei TPI[®] formal keinen offiziellen Standard berücksichtigt. Die Erfahrungen und Inhalte von TPI[®] sollen deshalb im Schema des internationalen Standards SPICE dargestellt werden.

Im Folgenden werden nun die Anforderungen an ein SPICE-konformes Modell erläutert, mit den Konzepten von TPI[®] verglichen und diese dann entsprechend umgeformt, ergänzt oder angepasst.

3.2 Das Prozess-Referenzmodell

Ein Prozess-Referenzmodell muss nach ISO/IEC 15504 bestimmte Anforderungen erfüllen, damit es für ein Assessment geeignet ist. Im folgenden Bild werden noch mal die wichtigsten Elemente wiederholt.

Ein Prozess-Referenzmodell beschreibt für ein bestimmtes Anwendungsgebiet eine Menge von Prozessen. Hierzu muss der Verwendungszweck angegeben werden.

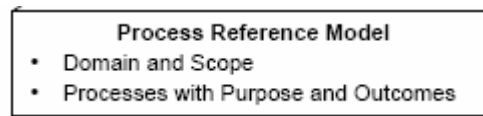


Abbildung 3.1: Normative Elemente des Prozess-Referenzmodells (aus [ISO05])

Anwendungsgebiet und Verwendungszweck

Für das Prozess-Referenzmodell von TPI[®] ist das Anwendungsgebiet der Test der Software-Entwicklung. Es sind alle dazugehörigen Aspekte dieser Domäne enthalten. Das Modell umfasst Low-Level-Tests, auch Modul- und Integrationstest genannt, genauso wie High-Level-Tests, also den System- bzw. Abnahmetest. Dabei berücksichtigt das TPI[®]-Modell, dass in der Praxis anfangs eher der Systemtest im Mittelpunkt steht und erst beim reiferen Testprozess die Low-Level-Tests und die Abstimmung zwischen den einzelnen Teststufen betrachtet wird [PKSp02].

Der Verwendungszweck des Prozess-Referenzmodells nach TPI[®] ist dessen Einsatz bei der Bewertung von Softwaretestprozessen im Rahmen von Assessments nach ISO/IEC 15504.

Prozesse mit Zweck und erwarteten Ergebnissen

Das TPI[®]-Modell enthält 20 Kernbereiche. Für das PRM wurden alle diese Kernbereiche als Prozesse übernommen.

Bei einigen Kernbereichen stellte sich die Frage, ob sie als eigener Prozess geeignet oder eventuell zu speziell formuliert sind und man sie deshalb besser mit anderen zusammenfassen sollte. So erscheinen die Kernbereiche „Testumgebung“, „Testarbeitsplatz“ oder auch „Testfunktionen und Ausbildungen“ nicht gut für einen SPICE-Prozess geeignet. Aber auch beim exemplarischen SPICE-Modell, dem Teil 5 der Norm nach ISO 12207, finden sich die Prozesse „RIN.4 Infrastructure“ und „RIN.2 Training“. Auch hier wurden diese Themenbereiche als wichtig erachtet und deshalb gesondert aufgeführt.

Die Überlegung, Bereiche wie „Engagement und Motivation“ und „Kommunikation“ eventuell zusammenzufassen wurde ebenso verworfen. Bei der Erstellung von TPI[®] wurden diese „weichen“ Themen bewusst getrennt, da sich bei den Erfahrungen in der Praxis gezeigt hat, dass Projekte oft an diesen Dingen scheitern und deshalb nicht zu unterschätzen sind. Für beide Bereiche gibt es eine Vielzahl an unterschiedlichen Verbesserungsmöglichkeiten und Maßnahmen und eine getrennte Betrachtung ist sehr sinnvoll. Dies gilt auch für den „Zeitpunkt der Beteiligung“, der zwar im Nachhinein nicht mehr zu ändern ist, aber sehr viel über die Reife und Qualität des Testprozesses aussagt. [PKSp02]

Außerdem bereitet es keine Probleme für alle Kernbereiche Rollen, Verantwortlichkeiten, Arbeitsprodukte (wie etwa Protokolle oder Projektpläne) zu bestimmen.

Der 21. Kernbereich „Integrationstests“ wurde wie beim Modell TPI[®] Automotive [Soge04] ergänzt, um die Integration von Komponenten und deren Zusammenspiel mehr zu betonen. Gerade bei großen Softwaresystemen spielt das Zusammenführen von Teilsystemen von oftmals unterschiedlichen Herstellern eine wichtige Rolle. Besonders die Bedeutung der Schnittstellen und deren Test wird dabei meist zu wenig beachtet. Der

TPI[®]-Bereich „Low-Level-Tests“ beschreibt dagegen den Test der individuellen Komponenten.

Eine Gruppierung der einzelnen Prozesse wurde unterlassen. In [PKSp02] werden die Kernbereiche entsprechend der Methodik TMap auf dessen vier Pfeiler aufgeteilt. Die Kernbereiche sind entsprechend den Pfeilern Phasenmodell, Techniken, Infrastruktur und Organisation zugeteilt. „Prüfen“ und „Low-Level-Tests“ wird allen zugeordnet.

Da das PRM für TPI[®] aber ein allgemeines Modell für Testprozessverbesserung sein soll und eine Gruppierung auch nicht bei allen Prozessen eindeutig möglich ist, wurde diese Einteilung nicht übernommen. Eine andere sinnvolle oder hilfreiche Gruppierung konnte ebenfalls nicht gefunden werden.

Die Aufzählung der Bereiche in der Reihenfolge wie sie in der Matrix beschrieben wird ist übersichtlich und allen TPI[®]-Anwendern geläufig und wurde deshalb nicht geändert.

Die Prozesse haben alle das Kürzel „TPI“ und die entsprechende Nummerierung als ID enthalten, also z.B. „TPI.1 Teststrategie“.

Neben dem Prozessnamen und einer eindeutigen ID sind noch Prozesszweck und Prozessergebnisse zu definieren. Diese wurden aus den Beschreibungen des TPI[®]-Modells abgeleitet und formuliert. Der Prozesszweck informiert in ein bis zwei Sätzen worum es in diesem Prozess geht. Die Prozessergebnisse (*outcomes*) sind ebenfalls aus den Beschreibungstexten entstanden.

Die Kontrollpunkte des TPI[®] waren die Vorlage für Basispraktiken, die dem Assessor zur Beurteilung dienen, ob ein Reifegrad erreicht wurde oder nicht.

Anschließend wurden die Basispraktiken den Ergebnissen zugeordnet. Dies sind keine 1:1 Beziehungen, d.h. einem Ergebnis können mehrere Basispraktiken und umgekehrt auch einer Basispraktik mehrere Prozessergebnisse zugeordnet werden (vgl. Beispielprozesse in Anhang A). Außerdem wird, im Gegensatz zum Beispielmodell aus SPICE, Teil 5, in den Basispraktiken nicht immer nur der Text aus den Ergebnissen umformuliert und wiederholt, sondern die Aussage kann sich etwas unterscheiden. Das kommt daher, dass die Ergebnisse aus der Beschreibung und die Praktiken aus den Kontrollpunkten abgeleitet sind und somit mehr Inhalt umfassen.

Beispielsweise wird in „TPI.1 Teststrategie“ bei den Prozessergebnissen die Strategiebestimmung aufgrund von Risikoabschätzungen und eine Optimierung bei den Testaktivitäten als Ziel angegeben. Die Basispraktiken enthalten dann konkrete Arbeitsanweisungen wie „differenziere die Testintensität abhängig von den erkannten Risiken“ oder auch „benutze Testspezifikationstechniken“. In eckigen Klammern wird dann angegeben welche gewünschten Ergebnisse des Prozesses mithilfe der Basispraktiken erreicht werden können, z.B. „[Outcomes 1, 2]“.

Eine genaue Beschreibung von Verbesserungsmaßnahmen ist nach ISO/IEC 15504 nicht nötig, allerdings in der Praxis sehr hilfreich. Deshalb wurden die Optimierungsvorschläge von TPI[®] mit in das neue Modell übernommen. Hier stehen dann Ratschläge wie etwa bei TPI.1, dass man alle Beteiligten bei der Strategiebestimmung mit einbeziehen und das nötige Bewusstsein für das nötige Vorgehen schaffen sollte.

Außerdem wurden die Abhängigkeiten zu anderen Prozessen aus dem TPI[®]-Modell übertragen, um das Assessment zu unterstützen. Der Prozess Teststrategie hängt eng mit den Testspezifikationstechniken und „Engagement und Motivation“ zusammen.

Eine weitere Hilfe bei der Bewertung von Prozessen sind die Arbeitsprodukte. Das können Ziele, Pläne, Listen, Berichte, Strategien, Design oder auch Software sein. Entsprechend der Prozessinhalte wurden Arbeitsprodukte definiert, die bei der Durchführung des

jeweiligen Prozesses entstehen. Da als eingehende Arbeitsprodukte beim Test eigentlich immer das zu testende Produkt und die Anforderungen stehen, wurde darauf verzichtet die „Input Work Products“ gesondert aufzuführen. Die im Prozess geänderten Arbeitsprodukte werden eh in den „Output Work Products“ aufgelistet. Dort werden auch die entscheidenden neu erzeugten Arbeitsprodukte genannt.

Beim gewählten Beispiel der Teststrategie werden durch die Ausführung des Prozesses die Arbeitsprodukte Validationsstrategie, Risikostatusbericht und der Softwaretestplan beeinflusst.

Die folgende Tabelle 3.2 gibt einen Überblick zur Umsetzung der PRM-Anforderungen im TPI[®]-Modell nach SPICE:

TPI[®]	SPICE
Kernbereich (<i>key area</i>)	Prozess (<i>process</i>)
Beschreibung (<i>description</i>)	Prozesszweck (<i>process purpose</i>) Ergebnisse (<i>outcomes</i>) Arbeitsprodukte (<i>work products</i>)
Kontrollpunkte (<i>checkpoints</i>)	Basis Praktiken (<i>base practices</i>) Generische Praktiken (<i>generic practices</i>)
Abhängigkeiten (<i>dependencies</i>)	Abhängigkeiten (<i>dependencies</i>)
Optimierungsvorschläge (<i>improvement suggestions</i>)	Verbesserungsvorschläge (<i>improvement steps</i>)

Tabelle 3.2: Abbildung der TPI[®]-Elemente auf ein SPICE-PRM

Einige Beispielprozesse des TPI[®] Prozess-Referenzmodells sind im Anhang A dargestellt. Das gesamte Modell ist auf der beiliegenden CD enthalten.

3.3 Der Bewertungsrahmen

Der normative Teil 2 von ISO/IEC 15504 fordert einen einheitlichen Bewertungsrahmen für Prozess-Assessment-Modelle. Hier noch einmal die wichtigsten Bestandteile:

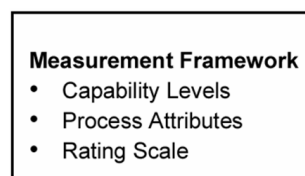


Abbildung 3.2: Normative Elemente des Bewertungsrahmens (aus [ISO05])

Reifegradstufen bei SPICE

Wie in 2.1.3 bereits aufgeführt, definiert die ISO/IEC 15504 sechs Reifegradstufen von 0 (unvollständig) bis 6 (optimierend). Bei TPI[®] werden drei verschiedene so genannte Kategorien für die Bewertung des Testprozesses beschrieben. Um diese Einteilungen besser vergleichen zu können, werden sie hier an diese Stelle mit ihren Bedeutungen und Auswirkungen näher erklärt. Da es bei TPI[®] auch den Begriff Stufe gibt, allerdings in einer

etwas anderen Bedeutung als die Reifegradstufe bei SPICE (diese entsprechen eher den TPI[®]-Kategorien) wird hier zur besseren Unterscheidung der Begriff SPICE-Level verwendet.

SPICE-Level 0 bedeutet, dass der entsprechende Prozess überhaupt nicht existiert oder aber seinen Zweck nicht erfüllt, er ist somit „unvollständig“. Der Projekterfolg hängt von wenigen „Helden“ ab und ist nicht vorhersagbar.

Bei Level 1 wird der Prozess „durchgeführt“, er ist vorhanden, muss aber nicht perfekt sein.

Der Level 2 wird vergeben, wenn der Prozess „gemanagt“ ist. Wichtig hierbei sind Planung, Kontrolle und entsprechendes Konfigurationsmanagement.

Level 3 wird „etabliert“ bezeichnet. Es bestehen organisationsweit definierte Prozesse, die für die jeweiligen Projekte angepasst (getailort) werden können. Dies bildet die Basis für die Durchführung und Verbesserung der Projektaufgaben. Es kommen explizite Rollenbeschreibungen, Infrastrukturanforderungen und eine Analyse der Prozessausführung dazu.

Ein Prozess auf Level 4 ist „vorhersagbar“. Entscheidend ist hier ein quantitatives Verständnis für Messungen bzw. Statistiken und damit entsprechende Kontrolle des Prozesses. Das ermöglicht Steuerung und Korrekturmaßnahmen bei der Durchführung von Projekten.

Level 5 ist schließlich für „optimierende“ Prozesse vorgesehen. Prozesse werden kontinuierlich verbessert und Innovationen zur weiteren Optimierung untersucht. Außerdem bestehen langfristige Strategien um die Effektivität zu erhöhen.

[PKSp02], [RoDo05]

Kategorien bei TPI[®]

Im TPI[®]-Modell sind die Stufen von 0 bis 13 in drei Kategorien unterteilt bzw. zusammengefasst.

Die Stufen 1 bis 5 werden als „beherrschbar“ beschrieben. Der Fokus liegt dabei hauptsächlich bei der Verwaltung des Testprozesses. Der Prozess ist kontrolliert, beim Testen wird spezifiziert und dokumentiert.

Die Ebenen der Stufen 6 bis 10 unterstützen einen Testprozess, der möglichst „effizient“ ist. Es geht um Automatisierung, bessere Abstimmung und Integration. Zudem ist eine Verankerung der Arbeitsweisen in der Organisation wichtig.

Die Stufen 11 bis 13 sind „optimierend“. Eine ständige Verbesserung ist angestrebt, der Testprozess wird an veränderte Verhältnisse und Neuentwicklungen angepasst. [PKSp02]

Zuordnung von SPICE-Levels und TPI[®]-Kategorien

Zunächst erscheint die Einteilung in verschiedene Kategorien bzw. Reifegradstufen bei den beiden Modellen recht unterschiedlich. Bei einigen Kernbereichen von TPI[®] zeichnen sich höhere Ebenen dadurch aus, dass sie eine größere Reichweite haben, also dass Teststrategien erst für einzelne Tests, dann für alle Tests einer Teststufe und dann für alle Teststufen angewandt werden.

Bei genauerer Betrachtung weisen die Entwicklungsebenen der verschiedenen Modelle allerdings doch sehr viele Ähnlichkeiten auf, da auch bei SPICE in den unteren Levels eine eher unsystematischer, wenig kontrollierter Einsatz von Methodiken vorherrscht und bei höheren Reifegradstufen eine organisationsweite, dokumentierte Vorgehensweise für alle Bereiche vorgegeben wird.

Die Stufe 0 eines unvollständigen Prozesses ist bei TPI[®] genauso wie bei SPICE enthalten, sie wird nur nicht explizit als Kategorie erwähnt.

Die Kategorie „beherrschbar“ bei TPI[®] umfasst im Prinzip die Levels 1 und 2 bei SPICE. Es geht um durchgeführte Prozesse, die geplant und beherrscht werden. Die bei TPI[®] zugeordneten Ebenen kann man auf die zwei SPICE-Levels aufteilen, da auch hier unterschiedliche Entwicklungsniveaus enthalten sind. Das ist gut daran zu erkennen, dass sowohl Ebene A als auch Ebene B in den TPI[®]-Stufen eins bis fünf enthalten sind.

Ebenso sind die TPI[®]-Kategorie „effizient“ und die SPICE-Levels 3 und 4 inhaltlich sehr ähnlich. Hier wird eine einheitliche Vorgehensweise in der gesamten Organisation entscheidend und Metriken werden als wichtig erachtet.

Die obersten Stufen enthalten schließlich bei beiden Modellen übereinstimmend Optimierung und kontinuierliche Verbesserung.

Die folgende Matrix in Abbildung 3.3 verdeutlicht die erste grobe Zuordnung der jeweiligen Stufen und Kategorien bei TPI[®] und SPICE. Die genaue Einteilung und ihre Hintergründe werden in Kapitel 3.4 erklärt.

TPI -Kategorien		beherrschbar					effizient					optimierend		
Kernbereich / Stufe	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A					B				C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A		B							
7 Metriken						A			B			C		D
8 Testautomatisierung					A			B			C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A				B						C		
12 Testfunktionen und Ausbildung				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A		B								C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
SPICE-Level	0	1		2		3		4		5				

Abbildung 3.3: Zuordnung TPI[®]-Kategorien und SPICE-Levels

Prozessattribute

Ob ein bestimmter Reifegrad erreicht ist, wird bei SPICE anhand von neun Prozessattributen beurteilt. Sie sind den Reifegradstufen zugeordnet und beschreiben eine bestimmte inhaltliche Anforderung und den konkreten Aspekt der jeweiligen Reifegradstufe. Damit ist es möglich die Leistungsfähigkeit eines Prozesses zu messen.

Eine weitere Hilfe bei der Bewertung von Prozessen liefern die so genannten Indikatoren der Prozessattribute. Sie bezeichnen wichtige Aktivitäten (als Generische Praktiken bezeichnet), Ressourcen oder Arbeitsprodukte.

Hier sind nochmals alle Reifegradstufen und Prozessattribute im Überblick zu sehen.

Level 0 enthält keine Prozessattribute. Diese Stufe wird immer erreicht. Die Prozessattribute der anderen Reifegradstufen sind Tabelle 3.3 zu entnehmen.

	Capability Level	Prozess Attribute
0	Incomplete	-
1	Performed	PA1.1 Process performance
2	Managed	PA2.1 Performance management PA2.2 Work Product management
3	Established	PA3.1 Process definition PA3.2 Process deployment
4	Predictable	PA4.1 Process measurement PA4.2 Process control
5	Optimizing	PA5.1 Process innovation PA5.2 Process optimization

Tabelle 3.3: Prozessattribute und Reifegradstufen bei SPICE (nach [ISO05])

Solche Prozessattribute sind im TPI[®]-Modell nicht vorgesehen. Nachdem aber Indikatoren wie z.B. Praktiken bestehen und Arbeitsprodukte definiert wurden, ist es leicht sie ins Prozess-Assessment-Modell für TPI[®] nach SPICE zu übertragen und dort anzuwenden. Sie stellen eine genauere Untergliederung und gute Hilfe für das Assessment dar. Details zur Bewertung werden im nächsten Kapitel erläutert.

Bewertungsskala

Die Bewertungsskala bei SPICE ist vierstufig mit den Stufen N, P, L und F für „not“, „partially“, „largely“ und „fully achieved“. Um bei einem Prozessattribut einen bestimmten Reifegrad zu erreichen muss mindestens ein „largely“ erreicht werden.

Bei TPI[®] kann man eine Ebene eines Kernbereiches nur komplett oder gar nicht erreichen. Zwischenstufen für eine teilweise oder überwiegende Erfüllung sind nicht definiert. Eine Zuordnung ist jedoch leicht möglich wie in der folgenden Tabelle 3.4 zu sehen.

Nachdem die Bewertung anhand von mehreren Kontrollpunkten bzw. Basispraktiken erfolgt und auch immer „Zwischenstufen“ auftreten, ist es kein Problem die Skala von SPICE für TPI[®] anzuwenden. Im Gegenteil, es werden sogar genauere und aussagekräftigere Ergebnisse liefert.

SPICE	TPI[®]
N (nicht erreicht)	Nicht erreicht
P (teilweise erreicht)	
L (überwiegend erreicht)	Erreicht
F (komplett erreicht)	

Tabelle 3.4: Zuordnung der Bewertungsskalen von TPI[®] und SPICE

3.4 Das Prozess-Assessment-Modell

Vorüberlegungen

Auf den ersten Blick sind TPI[®] und SPICE beide Modelle, die helfen die Qualität von Software-Prozessen zu bewerten, Verbesserungsmaßnahmen zu ermitteln und damit eine höhere Reifegradstufe zu erreichen. Ein höherer Level bedeutet dabei jeweils die Optimierung von Qualität, Kosten und Zeitaufwand.

ISO 12207, worauf der fünfte Teil der SPICE-Norm basiert, enthält im Prozess-Referenzmodell nur Prozessbeschreibungen für SPICE-Level 1. Es werden keine Vorgaben für höhere Reifegradstufen gemacht. Bei TPI[®] sind Beschreibungen für unterschiedlich reife Entwicklungsstufen, die so genannten Ebenen A bis D, enthalten. Die Aufgabe besteht nun darin diese TPI[®]-Ebenen auf die SPICE-Levels abzubilden und das Prozess-Referenzmodell und den Bewertungsrahmen im Prozess-Assessment-Modell zusammenzuführen.

Im vorherigen Kapitel zum Bewertungsrahmen wurde bereits eine grobe Zuordnung mithilfe der TPI[®]-Kategorien „beherrschbar“, „effizient“ und „optimierend“ getroffen. Hierbei ist allerdings zu beachten, dass diese Bewertung bei TPI[®] den Softwaretest im Ganzen beschreibt, bei SPICE aber nur Wertungen für jeden Prozess einzeln vergeben werden.

Weiterhin fällt auf, dass nicht für jeden Kernbereich alle Ebenen von A bis D existieren. Beim Testarbeitsplatz gibt es zum Beispiel nur Ebene A, bei anderen sind nur zwei (Statische Testtechniken) oder drei Ebenen (Testumgebung) beschrieben. Die Ebenen der verschiedenen Kernbereiche sind sehr unterschiedlich in der Matrix verteilt, je nachdem wann der Einsatz eines Bereiches im Gesamtzusammenhang sinnvoll ist. So ergibt es sich, dass „Metriken“ oder „Prüfen“ erst bei höherer Testprozessreife definiert werden. Durch die Abhängigkeiten zwischen den Ebenen der Kernbereiche entsteht daher eine Matrix mit „Lücken“, also mit leeren Zellen.

Eine Überlegung wäre, das TPI[®]-Modell zu einem Stufenmodell wie bei CMMI umzuformen. In jeder Reifegradstufe werden unterschiedliche Bereiche dazu genommen und nicht von Anfang an alle betrachtet. Bei näherer Betrachtung erweist sich diese Idee jedoch als problematisch. Zum einen ist SPICE ein kontinuierliches Modell, bei dem jeder Prozess für alle Stufen definiert ist. Zum anderen trifft das Stufenmodell nicht wirklich auf TPI[®] zu. Beim Stufenmodell tritt ein Prozess nur in jeweils genau einer Reifegradstufe auf und ist ihr zugeordnet. Bei TPI[®] dagegen gibt es für einen Bereich mehrere Ebenen, die auf unterschiedliche Stufen verteilt sind.

Ein weiterer Ansatz wäre einfach eine genaue Zuordnung der Ebenen A bis D zu den SPICE-Levels 1 bis 4 vorzunehmen und den fünften Level zu ergänzen. Damit würde man allerdings die Abhängigkeiten in der TPI[®]-Matrix völlig ignorieren. Das Herzstück dieses Modells, nämlich die wichtigen Erkenntnisse aus der Praxis bezüglich der Optimierungsreihenfolge der verschiedenen Bereiche würden somit über Bord geworfen (Zitat „Automating chaos leads to faster chaos.“ siehe Kapitel 2.2.3).

Aus diesem Grund wurden für die Umsetzung des PAM für TPI[®] nach SPICE sowohl die TPI[®]-Matrix und die Einteilung auf die drei Kategorien, als auch die Ebenen der einzelnen Kernbereiche berücksichtigt.

Zuordnung der Stufen von TPI[®] und SPICE

Ebene A bei TPI[®] beinhaltet die grundlegenden Anforderungen an einen Kernbereich. Dies entspricht der Stufe 1 bei SPICE. In der Matrix treten einige Bereiche allerdings erst relativ spät in Erscheinung. Aus diesem Grund wurden hier zusätzliche Ebenen für diese Kernbereiche bzw. Prozesse eingeführt.

Bei „Metriken“ wurde eine Ebene „Basismetriken“ für SPICE-Level 1 eingefügt, die erfüllt ist, wenn der Prozess überhaupt existiert. Hier müssen noch keine bestimmten Metriken wie Input- oder Outputmetriken vorhanden sein. Es genügt, wenn erst einmal einzelne Metriken gesammelt und analysiert werden, um hier bereits das Bewusstsein dafür zu schaffen und die Grundlagen für höhere Stufen zu legen.

Bei „Prüfen“ wurde ebenfalls eine Basisebene „Informelles Prüfen“ ergänzt. Hier geht es darum, dass Checklisten und Reviews durchgeführt und dokumentiert werden, ohne dass hier bereits detaillierte Anforderungen erfüllt werden müssen.

Um die Stellung der Ebene A eines Kernbereiches in der Matrix einzubeziehen, das heißt das Auftreten bereits in der untersten Entwicklungsstufe der TPI[®]-Matrix oder erst in höheren Stufen, wurden den Prozessen noch Prioritäten (*priority*) von 1 bis 4 zugeordnet. Priorität 1 bedeutet hierbei, dass der Level 1 dieses Prozesses möglichst frühzeitig beim Verbesserungsprozess angestrebt werden sollte, während ein Prozess mit Priorität 4 erst später optimiert werden kann.

Die weiteren Ebenen von TPI[®] wurden den SPICE-Levels bzw. den Prozessattributen nach ihrer Stellung in der Matrix und nach ihrem Inhalt zugeordnet (s.

Tabelle 3.5). Dabei wurden die Inhalte einer Ebene nicht getrennt und auf verschiedene Levels aufgeteilt, da die Kontrollpunkte und Maßnahmen zusammengehören und nur gemeinsam Sinn machen. Es wurde versucht so viel wie möglich aus der bewährten TPI[®]-Matrix zu belassen, an einigen Stellen wurden aber trotzdem Ebenen als Kompromiss in etwas höhere oder niedrigere Stufen verschoben, wenn der Inhalt dort thematisch besser zum SPICE-Prozessattribut gepasst hat. Es soll schließlich ein ISO/IEC 15540-konformes Modell entstehen, so dass der SPICE-Inhalt letztlich entscheidend ist. Falls eine inhaltliche Zuordnung schwierig war, wie etwa bei „Zeitpunkt der Beteiligung“, wurde die Einordnung der Ebenen in der TPI[®]-Matrix als Anhaltspunkt genommen, um abzuleiten bei welcher Entwicklungsstufe welche Anforderungen und Maßnahmen angestrebt werden sollten.

Im Vordergrund stand immer der Praxisbezug. Beim Beispiel-Assessment wurde der Fokus darauf gelegt, wie sinnvoll und hilfreich die Zuordnung bei einer Prozessbewertung ist. und notfalls noch Korrekturen vorgenommen. Hier sind einige Beispiele zur Erläuterung:

TPI.15 Berichterstattung:

Ebene A bei TPI[®] beschreibt das Aufdecken von Abweichungen und wird deshalb SPICE-Level 1 zugeordnet. Ebene B enthält Fortschritt und Berichterstattung und entspricht somit inhaltlich SPICE-Prozessattribut 2.2 zum Management der Arbeitsprodukte, wo ebenfalls Dokumentation im Mittelpunkt steht. Ebene C behandelt Risiken und Empfehlungen anhand von Metriken. Auch wenn diese Ebene in der TPI[®]-Matrix früher, also auf niedrigerer Stufe eingeteilt ist, wird es inhaltlich SPICE-Prozessattribut „PA 4.1 Prozessmessung“ zugeordnet. Ebene D fordert schließlich, dass Empfehlungen einen Software Process Improvement Charakter haben. Prozessattribut 5.1 behandelt ebenfalls Prozessinnovation und Prozessverbesserungsmöglichkeiten und ist damit inhaltlich vergleichbar.

TPI level	PA 2.1	PA 2.2	PA 3.1	PA 3.2	PA 4.1	PA 4.2	PA 5.1	PA 5.2
1.B Combined strategy for high-level tests			x					
1.C Combined strategy for high-level tests plus low-level tests or evaluation						x		
1.D Combined strategy for all test and evaluation levels							x	
2.B Planning, preparation, specification, execution, completion	x							
3.B Start of the test basis			x					
3.C Start of the requirements definition						x		
3.D Project initiation								x
4.B Statistically substantiated estimating and planning						x		
5.B Formal techniques	x							
6.B Checklists			x					
7.A Project metrics (product)		x						
7.B Project metrics (process)				x				
7.C System metrics						x		
7.D Organization metrics (>1system)								x
8.B Managed test automation				x				
8.C Optimal test automation							x	
9.B Testing in the most suitable environment				x				
9.C 'Environment on call'								x
11.B Testing integrated in project organization	x							
11.C Test engineering							x	
12.B (Formal) methodical, technical, and functional support, management of test process, testware, infrastructure			x					
12.D Formal internal quality assurance							x	
13.B Organization generic				x				
13.C Organization optimizing, R&D activities							x	
14.B Project communication (defects, change control)	x							
14.C Communication within the organization about the quality of test processes							x	
15.B Progress, activities, defects with priorities		x						
15.C Risks and recommendations, substantiated with metrics					x			
15.D Recommendations have a Software Process Improvement character							x	
16.B Extensive defect management with flexible reporting facilities	x							
16.C Project defect management				x				
17.B External management of test basis and test object		x						
17.C Reusable testware				x				
17.D Traceability system requirements to test cases						x		
18.B Planning, execution, monitoring, and adjusting	x							
18.C Monitoring and adjustment within the organization							x	
19.A Evaluation techniques			x					
19.B Evaluation strategy					x			
20.B White box techniques			x					
20.C Low-level test strategy					x			
21.B Test strategy for integration		x						
21.C Standardized approach for integration				x				

Tabelle 3.5: Abbildung der TPI®-Ebenen auf SPICE-Levels

TPI.16 Dokumentation der Abweichungen

TPI[®]-Ebene A enthält Informationen zu interner Dokumentation der Abweichung und wird auf SPICE-Level 1 abgebildet. Ebene B beinhaltet die Dokumentation und Berichterstattung mit der Definition von Verantwortlichkeiten analog „PA 2.1 Management der Prozessdurchführung“ bei SPICE. Und Ebene C fordert Dokumentation der Abweichung im gesamten Projekt und wird Prozessattribut „PA 3.2 Prozessanwendung“ zugeordnet, da es bei beiden Beschreibungen um definierte Standardprozesse und Befugnisse zur Ausführung geht.

Eine noch genauere Zuordnung mit Kontrollpunkten aus TPI[®] und Generischen Praktiken aus SPICE ist in Auszügen in Anhang B zu sehen. Die komplette Exceltabelle befindet sich auf der beiliegenden CD, Anhang E.

Die Gewichtung beim Assessment

Nachdem die Zuordnungstabelle zu TPI[®]-Kategorien und SPICE-Levels erklärt wurde, muss noch gezeigt werden wie diese TPI[®]-Inhalte bei einem SPICE-Assessment zum Tragen kommen.

Zunächst wird die Tabelle als Generische Ressource dem Modell hinzugefügt. Generische Ressourcen in SPICE können Werkzeuge, Methoden oder auch Ergebnisdatenbanken sein. Diese gehen in die Bewertung eines Prozesses mit ein, genauso wie Generische Arbeitsprodukte und die Generischen Praktiken, die anderen beiden Indikatoren bei der Ermittlung des Reifegrades.

Für Level 1 existieren ja bereits Basispraktiken und Arbeitsprodukte für jeden einzelnen Prozess aus den untersten TPI[®]-Ebenen. Das trägt der Tatsache Rechnung, dass die Prozesse der meisten Unternehmen bei Reifegrad 1 eingestuft werden. Genauso ist bei TPI[®] Ebene A bei Testprozessen vorherrschend (siehe [Koom02]), was wiederum die richtige Zuordnung von Ebene A zu SPICE-Level 1 unterstreicht.

Eine Möglichkeit der Gewichtung wäre mit einer zusätzlichen Generischen Praktik die Zuordnungstabelle und damit die TPI[®] -Inhalte in die Bewertung miteinzubeziehen. Diese Praktik könnte etwa heißen: „Consider the SPICE TPI mapping table to achieve a full level according to TPI[®]“. Zum einen würde das allerdings die bisher vorhandene 1:1 - Zuordnung von Generischen Praktiken zu „achievements“ (entspricht in etwa den „process outcomes“ im Prozess-Referenzmodell) in der Beschreibung der Prozessattribute durchbrechen, da für die neue Generische Praktik kein solches „achievement“ existiert. Diese sind nach dem normativen Teil 2 von ISO/IEC aber fest vorgeschrieben.

Zum anderen würde der Inhalt von TPI[®] dann nur zu einem geringen Teil in das Gesamtergebnis einfließen, da die anderen Generischen Praktiken SPICE-spezifisch sind und nicht von TPI[®] abgeleitet. In [HDHM06] steht hierzu als Hinweis für Assessoren:

„Obwohl die Norm lediglich eine Bewertung der Prozessattribute vorsieht, werden in der Assessment-Praxis meistens auch die Basispraktiken und Generische Praktiken mittels der N/P/L/F-Skala bewertet, da dies eine objektive und nachvollziehbare Bewertung sehr gut unterstützt. [...] Diese Methode kann jedoch immer nur als Anhaltspunkt dienen, da die Basispraktiken in der Regel nicht gleichgewichtig sind und auch in der individuellen Situation eine unterschiedliche Gewichtung haben können. Gleiches gilt für Generische Praktiken.“

Aus diesem Grund wurde jeder Generischen Praktik ein Zusatz hinzugefügt, dass die Generische Ressource der Zuordnungstabelle zu beachten ist. Damit beeinflusst TPI[®] nicht nur das Ergebnis von einer einzigen Generischen Praktik. Somit hat der Assessor eine Hilfe und genaue Angaben für die Assessment-Durchführung, die Bewertung und auch die

Empfehlung von Verbesserungsmaßnahmen bei den verschiedenen Stufen der einzelnen Prozesse zur Hand. Und die letztendliche genaue Bewertung des gesamten Prozessattributes ist schließlich, wie oben zitiert, immer noch der Interpretation und Einschätzung des Assessors überlassen.

3.5 Integration in project>kit

Erstellen des Metamodells

Im vorherigen Kapitel wurde bereits eine kurze Einführung in das Metamodell von project>kit gegeben und ein Beispiel für die möglichen Zusammenhänge zwischen Prozesselementtypen aufgeführt.

Beim Prozess-Assessment-Modell für TPI[®] nach SPICE genügt eine einfachere Struktur, wie Abbildung 3.4 zeigt. Es werden Artefakte (*artifacts*) benötigt, die Aktivitäten (*activities*) zugeordnet sind. Eine Aktivität (das kann eine Basispraktik sein oder ein anderer Indikator) liefert oder verändert dabei ein Arbeitsprodukt, also ein Artefakt („changes/creates“). Außerdem muss es umgekehrt möglich sein einem Artefakt eine Aktivität zuzuordnen („is output work product“).

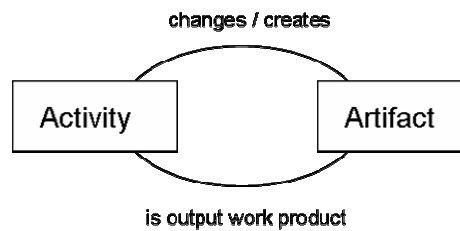


Abbildung 3.4: Assoziationen der Prozesselementtypen des Metamodells

Für diese Arbeit wurde das Metamodell für die Belange des erstellten Prozess-Assessment-Modells angepasst. Der komplette XML-Code befindet sich im Anhang D. Hier soll nur kurz zum besseren Verständnis auf die Besonderheiten hingewiesen werden.

Am Anfang wird ein eindeutiger Bezeichner, in diesem Fall „tpispice“, vergeben und das Modell als Referenzmodell markiert. Das ermöglicht das spätere Abbilden eines eigenen Prozesses auf die Inhalte des Referenzmodells:

```
<processtype ident="tpispice" referencemodel="true">
```

Anschließend folgt die Beschreibung des ersten Prozesselementtyps „activity“. Hervorzuheben sind hierbei die verschiedenen Subtypen wie „basepractice“ oder „level“, die als Aktivität verwendet werden. Die anderen Attribute erzeugen eine hierarchische Baumstruktur bei der Darstellung (*hierarchy*, *indexStyle*), ermöglichen die Definition von Tailoring-Bedingungen (*modalities*) und die Unterstützung von Abkürzungen (*shortnames*). Hier der entsprechende Ausschnitt aus der XML-Datei:

```
<element ident="activity" hierarchy="true" modalities="true"
subtypes="process basepractice activity level indicator"
shortname="true" indexStyle="tree">
```


Als nächstes werden im „model“ die Assoziationen definiert, in diesem Fall die Assoziationen der Aktivitäten zu den Artefakten.

Das Attribut „resources“ ermöglicht die Bereitstellung von Ressourcen im Downloadcenter von project>kit:

```
<model>
  <associations>
    <association target="artifact" types="output genoutput" />
  </associations>
  <resources/>
</model>
```

Nachdem auf Reifegradstufe 1 Arbeitsprodukte und auf den höheren Stufen Generische Arbeitsprodukte als Artefakte existieren, wurden zwei verschiedene Assoziationstypen definiert: „output“ für die Beziehung von Dokumenten zu Prozesselementen (*processes*) und „genoutput“ für Dokumente und Indikatoren (*indicators*) der höheren SPICE-Levels.

Im „view“ wird schließlich die Browserdarstellung der einzelnen Subtypen genauer festgelegt. „Subelements“ bewirkt die Ausgabe aller Kindelemente und „description“ die Anzeige einer Beschreibung. Hier ein Code-Ausschnitt:

```
<view>
  <subtype ident="activity">
    <description/>
    <subelements/>
  </subtype>
  ...
</view>
```

Die folgende Zeile verweist auf einen Text, der in der Datei LocalPkit.properties konfigurierbar ist:

```
<message key="artifact.output.tpispice"/>
```

Die Definition des zweiten Prozesselementtyps „activity“ weist eine ähnliche Struktur analog zu dem Element „artifact“ auf.

Das Metamodell enthält also zum einen eine allgemeine Definition von Prozesselementtypen und deren Assoziationen. Andererseits sind auch sehr spezielle Beschreibungen zur Browserdarstellung enthalten, die dem Anwender eine einfache Suche und einen schnellen Überblick ermöglichen soll [Mroz06].

Eingabe und Export des Inhalts

Nach der Ergänzung der XML-Datei PKitProcess.xml mit dem erstellten Metamodell und der Installation von project>kit (Details siehe Anhang C) konnte der Inhalt des Prozess-Assessment-Modells mit Prozessbeschreibungen, Verbesserungsvorschlägen usw. in project>kit eingegeben werden. Die Formatierung funktioniert mit der aus der Online-Enzyklopädie Wikipedia bekannten Wiki-Notation, so dass Überschriften, Aufzählungen und auch Verweise auf andere Elemente erzeugt werden können. (s. Ausschnitt in Abbildung 3.5). Außerdem wurde die erstellte Zuordnungstabelle eingefügt.

Diese Prozessbeschreibung kann exportiert und die resultierende Zip-Datei in andere project>kit-Installationen importiert werden. Eine Alternative ist der HTML-Export für einen Offline-Zugriff auf die erzeugten Inhalte, bei dem die gewohnte Oberfläche mit allen Hilfsmitteln erhalten bleibt.

Im Anhang dieser Arbeit befindet sich eine CD mit der HTML-Version des erzeugten Modells für TPI[®] nach SPICE.

Improvement Steps

Define the methodology in a handbook or test plan or refer to the literature. This will partly consist of putting together existing material.

Assure that there is or will be (as a result of training) sufficient test expertise, to make sure that the methodology described can in fact be followed.

Dependencies

- [TPI.2 Life-Cycle model, planning, specification, and execution](#)
- [TPI.5 Test specification techniques, formal techniques](#)
- [TPI.16 Defect management, internal defect management](#)
- [TPI.17 Testware management, internal testware management](#)
- [TPI.18 Test process management, planning, execution, monitoring, and adjusting](#)

[\[Change\]](#)

This process consists of the following subprocesses:

- [TPI.13.BP1: Formulate a methodology for each project. \[Outcomes 1, 2, 3\]](#)
- [TPI.13.BP2: Cover at least the basic activities. \[Outcomes 1, 2, 3\]](#)
- [TPI.13.BP3: Follow the methodology. \[Outcomes 2, 3\]](#)

Output work products:

- [08-12: Project plan](#) [Outcomes 2, 3]
- [08-21: Software test plan](#) [Outcomes 2, 3]

Abbildung 3.5: Ausschnitt der Browserdarstellung eines Prozesses in project>kit

4 Praktische Anwendung

In dieser Arbeit soll es nicht bei einem theoretisch erstellten Konzept zur Unterstützung des Testprozesses in der Software-Entwicklung bleiben, sondern das erstellte Modell auch in der Praxis bei einem Testprojekt angewandt werden. Assessment-Durchführung, Ergebnisse und die sich daraus ergebenden Maßnahmen werden in diesem Kapitel beschrieben.

4.1 Das SPICE-Assessment

Der Erfolg von Reifegradmodellen hängt sehr stark von der Qualität des durchgeführten Assessments ab. Deshalb werden in der ISO/IEC 15504 im normativen zweiten Teil die Anforderungen an ein Assessment beschrieben. Dies soll eine Vorgehensweise und damit Ergebnisse garantieren, die so weit wie möglich objektiv, wiederholbar, konsistent und auch repräsentativ für die bewerteten Prozesse sind. [HDHM06]

Teil 3 von SPICE enthält eine Interpretation der Anforderungen, um die Durchführung eines Assessments zu konkretisieren und zu unterstützen. Hier werden verschiedene Assessment-Arten unterschieden:

- Selbstassessments durch Personen der eigenen Organisation
- unabhängige oder externe Assessments durch Organisationsfremde.

Dabei werden mehrere Rollen genannt. Der Sponsor gibt das Assessment in Auftrag, der Competent Assessor ist der erfahrene Leiter und wird unterstützt vom Assessor.

Auch die Ausbildung der Assessoren wird in Teil 3 von ISO/IEC 15504 geregelt. Eine mögliche Vorgehensweise für die Schulung von Assessoren nach den Normanforderungen ist das iNTACS-Schema (International Assessor Certification Scheme), das von Experten aus verschiedensten Industriezweigen entwickelt wurde [HDHM06].

Das im Rahmen dieser Arbeit durchgeführte Assessment dient lediglich zur beispielhaften Anwendung des entwickelten Modells und strebt keine offizielle Zertifizierung der Organisation an. Nachdem die Bewertung der Prozesse hier nicht von einem ausgebildeten und erfahrenen Assessor durchgeführt wurden, ist das Ergebnis keine offizielle SPICE-Bewertung. Es soll vor allem als Beispiel und Pilotanwendung dienen.

4.2 Durchführung

Um nun das erstellte PAM an einem Projekt in der Praxis auszuprobieren und auch verbessern zu können, wurde ein Testprojekt für das Probe-Assessment gewählt.

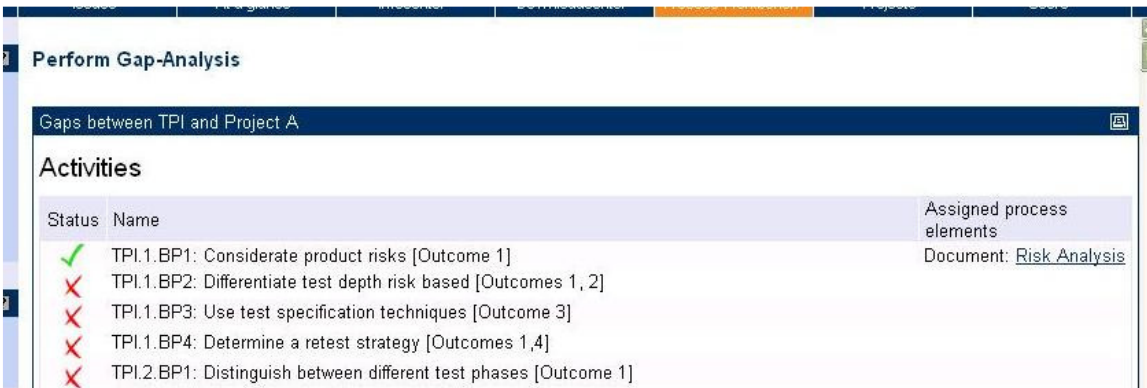
Dabei handelt es sich um ein Projekt bei einer deutschen Behörde, für die eine Berechnungssoftware mit grafischer Benutzeroberfläche erstellt wird. An diesem Produkt arbeiten etwa 30 Personen, davon 9 im Testbereich. Die Projektdauer beträgt ca. zwei Jahre.

Der Fokus des Assessments lag in erster Linie auf der Erprobung des Modells für TPI[®] nach SPICE und an zweiter Stelle auf der Bewertung der Testprozesse.

Als Richtgröße für ein Assessment geben Entwickler und Benutzer des TPI[®]-Modells etwa zwei Wochen an. TPI[®] nach SPICE ist noch umfangreicher und detaillierter, und so wurde der Aufwand hier angepasst und entsprechend beschränkt. In einem Prozess über mehrere Wochen, in dem die Erfahrungen zum Teil Änderungen und Anpassungen des Modells zur Folge hatten, wurde ein informelles Assessment durchgeführt. Dabei wurden Projektdokumente ausgewertet und einzelne Mitarbeiter, Testmanager sowie Tester, befragt.

Das Assessment wurde im Februar 2007 über alle Testprozesse durchgeführt und eine Bestandsaufnahme der Prozessreife gemacht.

Das Werkzeug project>kit bietet dazu auch die Möglichkeit den eigenen Software-Entwicklungsprozess auf ein Referenzmodell abzubilden, wie hier in Abbildung 4.1 zu sehen:



Status	Name	Assigned process elements
✓	TPI.1.BP1: Considerate product risks [Outcome 1]	Document: Risk Analysis
✗	TPI.1.BP2: Differentiate test depth risk based [Outcomes 1, 2]	
✗	TPI.1.BP3: Use test specification techniques [Outcome 3]	
✗	TPI.1.BP4: Determine a retest strategy [Outcomes 1,4]	
✗	TPI.2.BP1: Distinguish between different test phases [Outcome 1]	

Abbildung 4.1: Referenzmodell-Mapping in project>kit

4.3 Assessment-Ergebnisse

Die Ergebnisse des Assessments in Bezug auf Handhabung und Anwendung des PAM für TPI[®] nach SPICE werden im nächsten Kapitel diskutiert. An dieser Stelle geht es um die Bewertungen der untersuchten Testprozesse des Softwareprojekts.

Ein wichtiger Punkt bei der Durchführung von Assessments ist die Zusicherung von Vertraulichkeit, um negative Auswirkungen durch die Bekanntmachung von projektinternen Problemen oder Vorgehensweisen zu vermeiden [HDHM06]. Aus diesem Grund wird hier die Projektbezeichnung anonymisiert und nur das Gesamtergebnis mit den wichtigsten Verbesserungsmaßnahmen genannt ohne auf Details der einzelnen Prozessbewertungen einzugehen.

Zunächst ist der momentane Ist-Zustand festzustellen, nach dem dann Verbesserungsmaßnahmen und weiteres Vorgehen formuliert werden können. Denn: „Ohne zu wissen wo man steht, nutzt einem ein Plan nichts.“ (Humphrey, 1989). [Nöss04] Es wurden alle 21 Prozessbereiche des Prozess-Assessment-Modells untersucht. Dabei wurde fünfmal die Stufe 0 festgestellt, dreizehnmal Stufe 1 und dreimal Stufe 2. Eine höhere Bewertung wurde für keinen der Prozessbereiche ermittelt.

Neben project>kit wurde auch eine Exceltabelle bei der Bewertung (siehe beigefügte CD) verwendet. Darin ist es möglich Bemerkungen einzugeben und Berechnungsformeln zur Ermittlung des erreichten Prozessreifegrades einzusetzen. Das nimmt dem Assessor Arbeit

ab, gibt einen guten Überblick und ermöglicht eine erste objektive Aussage über das Ergebnis.

Dabei ist zu beachten, dass die Bewertungen der Basispraktiken und Generischen Praktiken zum Teil nur einen Anhaltspunkt für das endgültige Ergebnis liefern. Die eigentliche Bewertung ist immer auch abhängig von der speziellen Projektsituation und der Einschätzung und Erfahrung des Assessors.

Die folgende Tabelle 4.1 gibt einen Überblick der Assessment-Ergebnisse. Die horizontale Achse zeigt die 21 Testbereiche und auf der vertikalen Achse den jeweils erreichten Reifegrad.

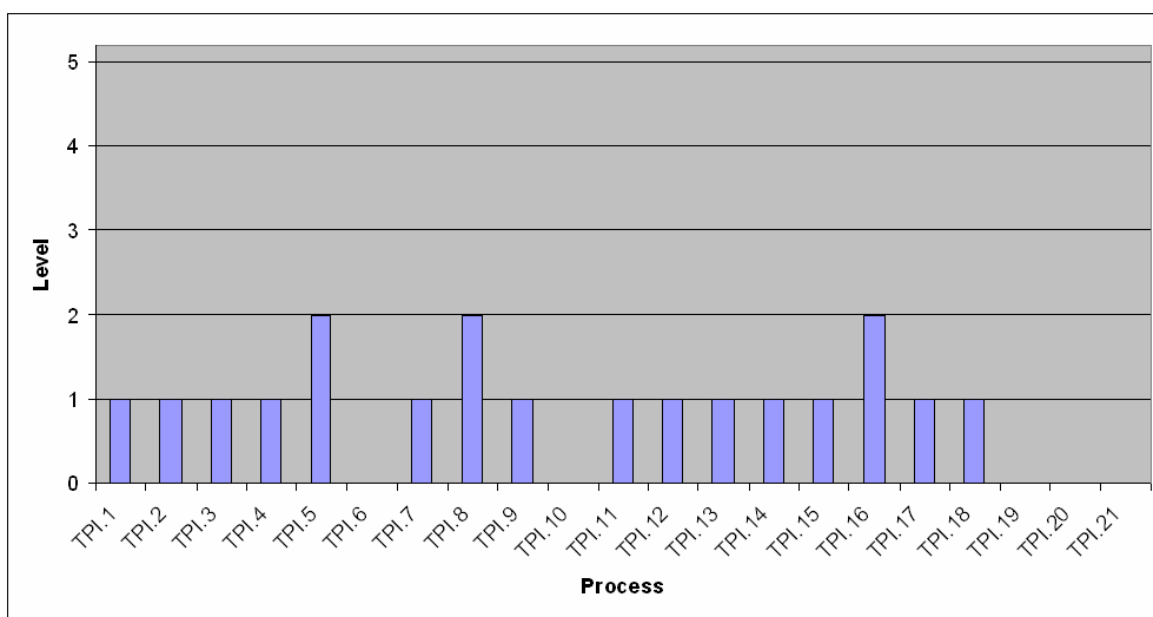


Tabelle 4.1: Übersicht der Assessment-Ergebnisse

Insgesamt zeigt sich, dass einige Bereiche“ auf der einen Seite schon etwas weiter entwickelt sind. Das ist bei den Testspezifikationstechniken (TPI.5) und dem Fehlermanagement (TPI.16) der Fall. Außerdem ist der Bereich „TPI.8 Testautomatisierung“ weiter fortgeschritten. Auf der anderen Seite fehlen aber noch grundlegende Vorgehensweisen bei „TPI.6 Statische Testtechniken“, „TPI.19 Prüfen“, „TPI.20 Modultest“ und „TPI 21 Integrationstest“. Verbesserungsbedarf besteht auch beim Testarbeitsplatz (TPI.10), da eine große Distanz zwischen Test und anderen Abteilungen besteht.

4.4 Maßnahmen

Aufgrund des Assessments ergab sich eine Menge an Verbesserungsvorschlägen. Eine Erkenntnis war, dass die Hauptprobleme eher bei organisatorischen und nicht bei technischen Mängeln zu suchen sind. Einige Änderungen scheinen dabei ohne größeren Aufwand möglich. Hier ist eher entscheidend, sich der Problematik bewusst zu werden und gegenzusteuern.

So wäre es sehr förderlich für das Projekt, wenn die räumliche Trennung zwischen Testern und restlichen Projektmitarbeitern aufgehoben würde. Damit wären auch die Bedingungen

für eine bessere Kommunikation geschaffen. Die bereits durchgeführten Besprechungen sollten konsequent regelmäßig stattfinden, um eine Abstimmung zwischen den Teststufen und ständigen Informationsfluss sicherzustellen.

Ein weiterer Vorschlag betrifft eine intensivere Ausbildung der Fachtester in Testmethodiken und damit auch die Anwendung von risikobasierten Teststrategien.

Im Projekt werden verschiedene Phasen im Entwicklungszyklus eingehalten. Leider wurde der Test etwas spät hinzugezogen und hat deshalb jetzt noch mit Änderungen der Anforderungen zu kämpfen. Eine frühere Beteiligung der Testabteilung und auch im späteren Projektverlauf eine saubere Trennung von einzelnen Phasen (wie Planung, Spezifikation, Durchführung) könnten hier schnell zur Prozessverbesserung beitragen.

Die Modultests und ebenso Integrationstests sollten verpflichtend und systematisch eingeführt werden, da hier die Möglichkeit besteht bereits frühzeitig Fehler in der Software zu entdecken.

Eine weitere Maßnahme wäre die Einführung von Checklisten, um die Vollständigkeit und Konsistenz von Dokumenten zu überprüfen. Außerdem könnten statische Prüfmethode konsequenter umgesetzt werden bei „Codereviews“ oder qualitätssichernden Maßnahmen bei der Definition von Anforderungen.

Ein generelles Manko ist, dass durchaus Erfahrungswerte aus anderen Projekten vorhanden sind, diese aber nicht genutzt werden, sondern meist eine neue Vorgehensweise für neue Projekte aufgesetzt wird. Vieles ist definiert, aber nicht explizit aufgeschrieben. Es ist empfehlenswert dedizierte Personen für das Qualitäts- und Konfigurationsmanagement zu benennen, auch wenn diese Rollen bereits teilweise gelebt werden.

Die vorhandenen Prozesse und ihre Dokumente sollten konsequent gepflegt und auch neue Versionen stets gereviewt werden.

4.5 Ausblick

Mit der Anregung von Verbesserungsmaßnahmen ist es natürlich nicht getan, um eine Veränderung zu bewirken. Es muss ein konkreter Aktionsplan entwickelt werden, bei dem klare Zuständigkeiten bestimmt werden. Um die Verbesserungen zu ermöglichen ist die Unterstützung des Managements nötig, das auch Zeit und Geld für die Umsetzung zur Verfügung stellen muss.

Außerdem ist es wichtig die Maßnahmen zu priorisieren und eine Auswahl der Verbesserungsmöglichkeiten zu treffen, da es keinen Sinn macht alles auf einmal ändern zu wollen. Die Aktionen müssen überwacht und gemanagt werden und die Ergebnisse sollten regelmäßig überprüft und gegebenenfalls die Maßnahmen angepasst werden.

Eine Verbesserung einer Organisation passiert nicht von heute auf morgen. Oft ist eine grundlegende Veränderung der Denk- und Arbeitsweise der Mitarbeiter nötig, die entsprechend Zeit braucht.

5 Diskussion

In diesem Kapitel werden nun die unkommentierten Ergebnisse und Erfahrungen aus dem vorherigen Abschnitt interpretiert und kritisch bewertet. Dabei sollen die Vor- und Nachteile des Lösungsansatzes herausgestellt und die Thematik mit Arbeiten zu ähnlichen Problemstellungen verglichen werden.

5.1 Interpretation und Bewertung der Ergebnisse

Allgemeine Vorgehensweise

Mit der vorliegenden Arbeit ist es gelungen ein Prozess-Assessment-Modell für TPI[®] nach SPICE zu entwickeln.

Dabei wurde das Modell TPI[®] gemäß den Anforderungen von ISO/IEC 15504 zu einem konformen Testprozessverbesserungsmodell umgewandelt und ergänzt, um diese Norm zu erfüllen. Die folgende Vorgehensweise, um aus einem vorliegenden Modell ein SPICE-konformes Modell zu entwickeln, hat sich dabei bewährt:

- Inhaltlichen und konzeptionellen Vergleich des Modells mit ISO/IEC 15504 (Fokus, Verbreitung, Vorgehen, Aufbau, Detaillierungsgrad, etc.) durchführen
- Anwendungsgebiet und Verwendungszweck des zu erstellenden Modells bestimmen
- Prozess-Referenzmodell definieren: Prozesse festlegen, eventuell gruppieren, jeweils ID, Prozesszweck mit Ergebnissen, Basispraktiken und Arbeitsprodukten bestimmen und aus den Vorgaben ableiten bzw. notfalls ergänzen
- Einheitlichen Bewertungsrahmen nach SPICE festlegen
- Prozess-Assessment-Modell mit der Zusammenführung von Prozess-Referenzmodell und Bewertungsrahmen erstellen
- Wenn im verwendeten Modell bereits verschiedene Entwicklungsstufen enthalten sind: Detaillierte Zuordnung dieser Entwicklungsstufen zu den Indikatoren der höheren SPICE-Reifegradstufen (Mappingtabelle)
- Wenn nötig: Gewichtung der Inhalte des verwendeten Modells beim Assessment-Ergebnis bestimmen (z.B. mit Ergänzungen bei Generischen Ressourcen und Generischen Praktiken)
- Praxistest bei Probe-Assessment in einem realen Projekt durchführen, eventuell Korrekturen vornehmen

Gemäß den TPI[®]-Inhalten wurden somit für die einzelnen Testbereiche konkrete Zwecke, Ergebnisse und Arbeitsprodukte definiert, die als Indikatoren zur Bewertung bei Assessments nötig sind.

Nachdem das Testmodell von sogeti weniger Kategorien vorsieht als SPICE Reifegradstufen und Prozessattribute hat, wurden die TPI[®]-Kategorien auf die Fähigkeitsstufen der ISO-Norm abgebildet und in einer Matrix entsprechend zugeordnet. Damit ist es bei Assessments auch bei höheren Fähigkeitsstufen möglich auf Generische Praktiken speziell für Testprozesse zuzugreifen und entsprechende Verbesserungsmaßnahmen zu treffen.

Das entstandene Modell für TPI[®] nach SPICE entspricht einem internationalen Standard und ist somit im Aufbau allgemein anerkannt und ermöglicht objektive, vergleichbare Ergebnisse. Das PAM ist unabhängig von Programmiersprachen, Test- oder Entwicklungsmethodiken und deshalb breit und vielfältig anwendbar. Trotzdem sind konkrete Detailanweisungen und Verbesserungsvorschläge für die Praxis enthalten, die sowohl wenig entwickelte als auch relativ „ausgereifte“ Testprozesse unterstützen [PKSp02]. Außerdem kann das PAM für alle oder auch nur für ausgewählte, einzelne Testprozesse eingesetzt werden, um den Ist-Zustand, Optimierungsschritte und weitere Fortschritte zu bestimmen.

Das Prozess-Assessment-Modell für TPI[®] nach SPICE hat einerseits gezeigt, dass sich der TPI[®]-Inhalt SPICE-konform präsentieren lässt. Andererseits unterstreicht es die Eignung und Stellung des Standards ISO/IEC 15504 als allgemeingültiges Rahmenwerk für Assessments mit Modellen aus unterschiedlichen Bereichen.

Das Prozess-Assessment-Modell

Im Gegensatz zum TPI[®]-Modell von sogeti enthält das Prozess-Assessment-Modell einige Ergänzungen und Erweiterungen, die den Einsatz bei Assessments erleichtern.

Es wurden zu erreichende Ergebnisse formuliert und auch Arbeitsprodukte spezifiziert.

Für die Kernbereiche bei TPI[®] war die Anzahl der Stufen sehr unterschiedlich und konnte von einer bis zu vier Stufen enthalten. Im PAM wurde die Zahl der Entwicklungsstufen pro Bereich erweitert und ist nun einheitlich. Zum Teil wurde noch eine niedrigste Stufe eingeführt, die die Lücke schließt, falls ein Testbereich erst zu einem fortgeschritteneren Entwicklungsstadium im TPI[®]-Modell erscheint. Somit können auch schon erste Ansätze zur Umsetzung eines Teilbereiches analysiert und entsprechend honoriert werden.

Neben kleineren inhaltlichen Ergänzungen wurden auch noch wie bei Automotive TPI[®], dem an die Belange der Automobilbranche angepassten Verbesserungsmodell, ein 21. Testbereich zum Integrationstest eingeführt.

Nachdem die Prozesse der meisten Unternehmen erfahrungsgemäß in die niedrigen SPICE-Levels eingestuft werden, ist es auch gerechtfertigt, dass die Darstellung der Inhalte von Level 1 im Modell detaillierter erfolgt, als bei den höheren Stufen.

Ein Manko des bisherigen TPI[®] war, dass eine Stufe nur ganz erreicht oder gar nicht erreicht werden konnte. Im Gespräch mit erfahrenen TPI[®]-Assessoren wurde das klar bemängelt. Hier bietet das PAM den Vorteil, dass man auch ein „teilweise erreicht“ oder „fast vollständig erreicht“ vergeben kann. Das macht das Ergebnis gerechter und transparenter und spiegelt die Realität besser wider als die „digitale“ Bewertungsweise.

Neben diesen Verbesserungen ergab sich auch ein Verbesserungsvorschlag für die Norm ISO/IEC 15504 selbst. So wurden die Optimierungsvorschläge und Abhängigkeiten von sogetis TPI[®] übernommen und im Prozess-Assessment-Modell verwendet. Das ist im SPICE-Standard nicht vorgeschrieben, wäre aber eine sinnvolle und hilfreiche Ergänzung.

Neben diesen gut gelungenen Erweiterungen ist die Abbildung der einzelnen TPI[®]-Entwicklungsstufen auf die SPICE-Levels sicherlich zum Teil noch diskutierbar bzw. muss sich erst im Praxiseinsatz bewähren. Mit dieser Arbeit wurde dafür ein erster Vorschlag erarbeitet. Das gilt auch für die Auswahl der Arbeitsprodukte, die noch ergänzt werden kann.

Praktische Umsetzung in project>kit

Das entwickelte Modell wurde im Prozess- und Projektmanagementtool project>kit in einer Browserdarstellung umgesetzt. Dafür wurde das flexible Metamodell von project>kit entsprechend angepasst und die Inhalte eingepflegt. Die Darstellung ermöglicht durch Verknüpfung der Prozessbeschreibungen und Arbeitsprodukte Beziehungen und Abhängigkeiten darzustellen und dem Anwender die Arbeit zu erleichtern. Bei Assessments können so dem Referenzmodell konkrete Dokumente oder Phasen des Projekts zugeordnet werden und die Bewertung unterstützen.

Eine weitere Hilfe bietet die Darstellung der Modellinhalte in einer Exceltabelle. Darin können beim Assessment Ergebnisse und Bemerkungen eingetragen werden. Aus den Einzelbewertungen werden dann die Gesamtbewertungen für die erzielten Reifegrade der Prozesse berechnet und übersichtlich dargestellt.

Eine wichtige Erkenntnis war, dass die Werkzeugunterstützung zwar keine bessere Qualität erzeugt, aber auf alle Fälle wichtig ist und einen Anreiz schafft, das Modell zu verwenden, da es die Handhabung vereinfacht. Unverzichtbar sind allerdings auch ein begleitendes Coaching bei der Einführung von Referenzmodellen und das Aufzeigen der resultierenden Vorteile und Verbesserungen. [Meie06]

Um die Akzeptanz bei den Anwendern weiter zu erhöhen, sind einige Ideen zur Verbesserung in project>kit entstanden. Es würde Assessments unterstützen, wenn mehr Beispieldokumente und Checklisten zum Referenzmodell und auch zur Durchführung des Assessments vorliegen würden. Das reicht von konkreten Fragebögen und Zeitplänen bis hin zu Vorlagen für Protokolle, Gesprächsnotizen oder Abschlusspräsentationen. Dadurch ließe sich die Hemmschwelle beim erstmaligen Einsatz des PAM deutlich herabsetzen. Denn entscheidend für den Erfolg eines Modells ist letztlich der Einsatz und Nutzen im Arbeitsalltag.

Das Probe-Assessment

Ein entscheidender Faktor bei der Erstellung des Modells war der direkte Einsatz anhand eines Beispielprojekts. Damit war es möglich Details (etwa bei den Arbeitsprodukten) zu korrigieren und ein plausibles, in der Praxis gut anwendbares PAM zu konstruieren. So wurde beim Assessment deutlich, dass es Sinn macht noch einen Prozess zum Integrationstest hinzuzufügen, um dessen Bedeutung für den Softwaretest zu verstärken. Außerdem wurde klar, dass organisatorische und menschliche Einflüsse und Probleme nicht zu unterschätzen sind. Deshalb wurden Prozesse zu Kommunikation, Motivation oder Ausbildung nicht zusammengefasst, sondern als eigene Bereiche belassen, um ihren Einfluss auf das Ergebnis der Bewertung nicht zu schmälern.

Das gewählte Projekt erwies sich als gut geeignet und repräsentativ für andere Software-Entwicklungsprojekte, da die Prozesse der meisten Unternehmen auf den unteren Fähigkeitsstufen stehen [Koom02]. Das rechtfertigt und unterstreicht auch die explizite Darstellung der Inhalte für die erste Reifegradstufe im Gegensatz zur generischen und allgemeineren Beschreibung der höheren Levels.

Das durchgeführte Assessment hat bestätigt, dass organisatorische Probleme in einem Projekt schwerer wiegen als technische Unzulänglichkeiten. Die Projekterfahrung passt hier sehr gut zum erreichten Ergebnis und wurde durch die festgestellten Fähigkeitsstufen gut getroffen. Diese Erkenntnis liefert schließlich den besten Beweis dafür, dass sich das Modell hier als Bewertungsschema eignet. Die daraus resultierenden Verbesserungsmaßnahmen wurden als hilfreich und nötig empfunden.

Im Rahmen dieser Arbeit war nur ein kleines Probe-Assesment möglich, das nicht von einem geschulten und erfahrenen Assessor durchgeführt wurde. Um aussagekräftigere Ergebnisse zu erzielen sind mehr Pilotprojekte auch aus verschiedenen Branchen nötig. Dieses Modell ist aber ein erster Vorschlag, der bereits seine prinzipielle Einsatzfähigkeit gezeigt hat. Weitere Verfeinerungen und Änderungen sind wie bei jeder ersten Version in zukünftigen Überarbeitungen möglich und nötig.

So muss sich auch der Einfluss der TPI[®]-Inhalte auf die Bewertung der höheren Stufen noch bewähren. Bisher werden nur einige der Generischen Praktiken von SPICE durch TPI[®] ergänzt und beeinflussen somit das Endergebnis in unterschiedlichem Maße. Die abschließende Bewertung liegt allerdings immer im Ermessen des Assessors, so dass die Einzelbewertungen der Generischen Praktiken eh nur eine Richtung vorgeben.

Fazit

Im Vergleich zum „ursprünglichen“ TPI[®]-Modell ist das Prozess-Assesment-Modell umfangreicher und damit auch etwas komplexer und schwieriger in der Handhabung. Es werden viele Aspekte beim Assesment herangezogen und untersucht, die bei TPI[®] keine Beachtung finden. Es erscheint somit zunächst zweifelhaft, ob ein bisheriger TPI[®]-Assessor zum PAM für TPI[®] nach SPICE wechseln wird.

Allerdings ist das „neue“ Modell konform zu einem verbreiteten Standard, so dass eine erzielte Bewertung mehr Gewicht und Aussagekraft erhält. Und vielleicht ist es ein Weg, um SPICE-Assessoren zum PAM für TPI[®] zu führen und damit die Bedeutung des Testprozesses zu stärken.

5.2 Vergleich mit verwandten Arbeiten

Für die Aufgabe ein Prozess-Assesment-Modell für TPI[®] nach SPICE zu entwickeln gibt es bisher keine anderen Lösungsansätze. Aus diesem Grund existiert keine direkte Vergleichsliteratur zu diesem Thema. Allerdings wurden zu ähnlichen Problemstellungen schon Vorschläge erarbeitet, deren wichtigste hier etwas genauer vorgestellt und bewertet werden sollen.

„TPI[®] Automotive – Test Process Improvement“

Zunächst wäre hier TPI[®] Automotive zu nennen [Soge94]. Dieses Testprozessverbesserungsmodell wurde in den Jahren 2003 und 2004 entwickelt, initiiert von den deutschen Automobilherstellern. Das vorhandene TPI[®]-Modell, in den neunziger Jahren von der Firma sogeti erstellt, wurde dazu erweitert und an die Besonderheiten der Automobilbranche angepasst.

Das bewährte TPI[®]-Schema wurde im Prinzip beibehalten. Eine herausragende Änderung war der 21. Kernbereich zum Integrationstest, der hinzugefügt wurde. Der Integrationstest ist in der Automobilbranche von besonderer Wichtigkeit, um die einzelnen Teilsysteme von verschiedenen Herstellern zu einem Gesamtsystem zusammenzubringen.

Außerdem wurden viele Beispiele mit Besonderheiten aus dem Automobilbereich bei den einzelnen Bereichen und Stufen hinzugefügt.

Der Kernbereich „Testarbeitsplatz“ wurde umbenannt zu „Büro- und Laborausstattung“. Bei den Testspezifikationstechniken wurde eine Ebene C mit mathematischen Methoden ergänzt und der Bereich der Reichweite der Methodik enthält eine weitere Ebene mit Schnittstellenbeschreibungen für die Zulieferer. Der Kernbereich Prüfen hat eine neue Ebene A zu nichtformalen Prüfungen enthalten, die bereits in Stufe 3 der Matrix erscheint und nicht wie die bisherige Ebene A erst bei Stufe 6.

Beim Testwaremanagement wurde dagegen die Ebene D zur Rückverfolgbarkeit der Systemanforderungen vorgezogen und mit der Ebene B zusammengefasst. Alles in allem ist ein branchenspezifisches TPI[®] entstanden, das aber keine Änderungen des Bewertungsschemas und der Vorgehensweise aufweist.

„Automotive-SPICE und IEC61508“

In [MeSc06] werden zwei Standards gegenübergestellt: IEC 61508, ein internationaler Standard für „Functional Safety Management“ (Systemsicherheit) und Automotive-SPICE. Letzteres ist der modifizierte Teil 5 aus ISO/IEC 15504, basierend auf ISO 12207 „Software Life Cycle Processes“. Die Herstellerinitiative Software (HIS) deutscher Automobilhersteller hat hierzu die für die eigene Branche relevanten Prozesse herausgegriffen, erweitert und entsprechend angepasst. Die Prozessgruppe zu Ressourcen und Infrastruktur wurde beispielsweise weggelassen und dafür Prozesse zu technischen, administrativen oder projektbezogenen Anforderungen hinzugefügt. Es wurden auch Basispraktiken oder Arbeitsprodukte ergänzt. Im Gegensatz zu dem exemplarischen Prozess-Assessment-Modell aus ISO/IEC 15504 wurden nur erzeugte Arbeitsprodukte definiert und die eingehenden Arbeitsprodukte weggelassen.

Die beiden Standards wurden hier zunächst konzeptionell verglichen. Insofern weist diese Arbeit ähnliches Vorgehen auf, wie bei der Erstellung des PAM von TPI[®] nach SPICE. IEC 61508 gibt jedoch kein Assessment-Verfahren an und schreibt keine Prozessverbesserungskriterien vor. Außerdem wurde keine Sicherheitsnorm konform zum SPICE-Standard erstellt, sondern die IEC 61508 inhaltlich dem Prozess-Assessment-Modell nach ISO 12207 gegenübergestellt, um die Inhalte beider Modelle implementieren zu können. Es wurde sozusagen Automotive-SPICE um „Functional Safety“ ergänzt.

„ISO/IEC 15504 (SPICE) und Automotive TPI[®]“

Ähnliches wurde auch bei der Gegenüberstellung von ISO 15504 und Automotive TPI[®] [Gros04] gemacht. Der Titel lässt zunächst vermuten, dass es sich um die gleiche Problemstellung wie bei der vorliegenden Abschlussarbeit handelt, nämlich die Erstellung eines Modells für Testprozessverbesserung für das TPI[®]-Modell nach den Richtlinien, die in ISO/IEC 15504, Teil 2, beschrieben werden. Tatsächlich wird aber eine inhaltliche Gegenüberstellung von Automotive TPI[®] und Teil 5 von SPICE, also dem Referenzmodell nach ISO 12207, erstellt. Dabei wurde untersucht wie SPICE (Teil 5) durch den Testprozess von TPI[®] abgedeckt wird. Zunächst wurden beide Modelle grundsätzlich verglichen. Anschließend wurden Prozesse identifiziert, die durch TPI[®] gar nicht, teilweise oder vollständig abgedeckt werden. In vielen Bereichen ist TPI[®] umfangreicher und detaillierter als der Standard zum Software-Entwicklungsprozess nach ISO 12207 und kann diesen somit gut ergänzen. Näheres wurde bereits in Kapitel 3 erläutert.

Als offene Punkte wurden ein Vergleich aller Prozesse einschließlich der enthaltenen Arbeitsprodukte genannt und eine Abbildung der TPI[®]-Stufen auf die Fähigkeitsgrade von SPICE. Letzteres wurde in dieser Abschlussarbeit erstellt.

„ISO/IEC 15504-5– Part 5: An exemplar Process Assessment Model“

Am meisten Übereinstimmung weist die vorliegende Arbeit mit dem fünften Teil der SPICE-Norm auf. Hier wurde ebenfalls ein Standard zur Software-Entwicklung verwendet, um nach den normativen Regeln von ISO/IEC 15504 ein Prozess-Assessment-Modell zu entwickeln. ISO 12207 beschreibt einen Rahmen für Prozesse im Lebenszyklus von Software [Wiki07a], der 1995 veröffentlicht wurde.

Dieser Standard enthält wichtige Prozesse, die bei der Entwicklung von Softwaresystemen von der Ideenfindung über die Entwicklung bis hin zu Installation und Wartung nötig sind. Diese Prozesse sind in Prozessgruppen eingeteilt und wurden bei der Definition des PAM übernommen. Die Basispraktiken und Arbeitsprodukte mussten dann noch als zusätzliche Information ergänzt werden, da sie als Indikatoren zur Prozess Erfüllung bei Assessment nötig sind. [Wiki07a]

Im Gegensatz zu TPI[®] sind bei ISO 12207 aber keine konkreten Verbesserungsvorschläge enthalten. Außerdem enthält dieser ISO-Standard keine unterschiedlichen Entwicklungsstufen. Somit war eine Abbildung auf die SPICE-Reifegradstufen nicht nötig, was bei der Erstellung des Modells für TPI[®] nach SPICE jedoch eine wichtige Rolle gespielt hat.

Zusammenfassend lässt sich feststellen, dass zwar einige Prozess-Assessment-Modelle nach ISO/IEC 15504 erstellt wurden. Aber speziell für Testprozessverbesserungsmodelle ist die vorliegende Arbeit ein Vorreiter, der vielleicht in Zukunft noch weitere Entwicklungen nach sich zieht.

6 Schlussbetrachtungen

Das letzte Kapitel enthält eine Zusammenfassung der erzielten Ergebnisse. Außerdem werden im Ausblick Ideen für interessante mögliche Fortführungen dieser Arbeit beschrieben und am Schluss ein Fazit gezogen.

6.1 Zusammenfassung

Im Rahmen dieses Projekts wurde erstmalig ein Lösungsansatz eines Prozess-Assessment-Modells für TPI[®] nach SPICE erstellt und in einem Beispielprojekt pilotiert. Damit ist es möglich mit dem bekannten und standardisierten SPICE-Bewertungsschema Assessments speziell für den Testprozess in Unternehmen durchzuführen.

Dazu wurden zunächst die Unterschiede und Gemeinsamkeiten der beiden Modelle TPI[®] und SPICE herausgearbeitet.

Anschließend wurde dem normativen Teil 2 des Standards SPICE (ISO/IEC 15504) entnommen, welche Anforderungen zur Erreichung der SPICE-Konformität eines Referenzmodells erfüllt sein müssen. Nach diesen Regeln wurde dann TPI[®] nach SPICE umgesetzt. Dabei waren einige Ergänzungen und Anpassungen nötig. Die Kernbereiche und Ebenen von TPI[®] wurden auf die Prozesse und Stufen des SPICE-Modells abgebildet. Außerdem wurden fehlende Strukturen und Inhalte ergänzt.

Bei der Ausarbeitung sehr nützlich war die Erprobung des Modells an einem konkreten Beispielprojekt, was dazu beitrug das erstellte Konzept schrittweise zu verbessern und in der Praxis zu überprüfen.

Im Projekt- und Prozessmanagementwerkzeug project>kit wurde das interne Metamodell angepasst, um es für das Prozess-Assessment-Modell für TPI[®] nutzen und die Inhalte einfügen zu können. Die Browserdarstellung der Prozesse mit der Möglichkeit der Verknüpfung der einzelnen Elemente erwies sich als große Hilfe bei der Assessment-Durchführung.

Bei der Erstellung des Modells sind zudem Ideen für Verbesserungsvorschläge sowohl für SPICE, als auch für TPI[®] und project>kit entstanden.

In der vorliegenden Arbeit wurde eine Vorgehensweise zur Erstellung eines PAM nach SPICE ermittelt und gezeigt, dass sich das Referenzmodell TPI[®] danach in ein SPICE-konformes Modell umwandeln lässt.

Und schließlich wird damit auch die Stellung von SPICE (ISO/IEC 15504) als allgemeiner internationaler Standard für Prozess-Assessments unterstrichen.

6.2 Ausblick

Der nächste Schritt wäre nun dieses Modell anhand einer größeren Anzahl von Testprojekten aus verschiedenen Bereichen zu validieren. Besonders interessant wären Assessments der höheren Reifegradstufen, welche allerdings eher selten vorgefunden werden. Die Erfahrungen könnten dann dazu benutzt werden die Umsetzung von TPI[®] nach SPICE zu verfeinern und eventuell zu ergänzen, beispielsweise mit noch fehlenden Arbeitsprodukten oder Anmerkungen.

Bei häufigerem Praxiseinsatz wäre es auch interessant die Akzeptanz bei Assessoren und bewerteten Unternehmen zu untersuchen. Mit Umfragen könnte statistisch ausgewertet werden in welchem Maße damit objektive Bewertungen und effektive Verbesserungen bei Testprozessen erzielt werden.

Daneben entstanden auch Ideen wie man eine effiziente Durchführung von Assessments unterstützen könnte. Ein Satz an Vorlagen für Fragebögen, Zeitpläne, Gesprächsprotokolle, Assessorennotizen, außerdem Checklisten oder Beispielpräsentationen speziell für Testprozesse wären dabei sehr nützlich.

Eine bessere Werkzeugunterstützung wäre auch denkbar bei der Erstellung von Assessment-Berichten, der Auswertung der Ergebnisse und deren Darstellung in Grafiken und Bildern.

Eine weitere Aufgabe für die Zukunft wäre somit, den Bedarf an solchen Hilfen zu ermitteln und sie zu erstellen.

Und schließlich ist noch die Definition weiterer SPICE-konformer Modelle denkbar, die den Standard nach ISO/IEC 15504 erfüllen.

6.3 Fazit

Bleibt am Ende die Feststellung, dass das entwickelte Prozess-Assessment-Modell von TPI[®] nach SPICE lediglich eine Richtschnur bei der Qualitätssicherung und Durchführung von Testprozessen darstellt. Die Verwendung des Modells ist dabei lediglich ein Hilfsmittel und kein Selbstzweck. Das Ziel sollte immer eine effektive Verbesserung und damit ein Gewinn für das Unternehmen sein.

Assessment-Modelle können dazu beitragen Software-Prozesse und damit auch die Softwarequalität zu verbessern. Entscheidend dabei ist die Akzeptanz bei allen Beteiligten - sowohl Anwendern, als auch dem Management. Wenn der Mitarbeiter erst vom Nutzen überzeugt ist und eine strukturierte Vorgehensweise zu schätzen gelernt hat, dann ist der anfängliche Widerstand gebrochen und die Prozesse werden nicht nur von oben verordnet, sondern wirklich gelebt.

Die Bedeutung und spezifische Anwendung von Referenzmodellen im Testbereich zur Verbesserung der Softwarequalität wird in Zukunft sicherlich weiter zunehmen.

Ich hoffe, dass diese Arbeit ein wenig dazu beiträgt die Rolle des Testens bei der Software-Entwicklung zu stärken und hilft die Meinung zu entkräften, dass Qualität von quälen kommen muss (s. Seite 3).

Anhang A: Beispielprozesse für TPI® nach SPICE

TPI.1 Test strategy

Process ID	TPI.1
Process Name	Test strategy
Process Purpose	The purpose of the Test strategy process is to define which requirements and (product) risks are covered by what tests in order to find the most important defects as early and cheaply as possible.
Process Outcomes	<p>As a result of successful implementation of this process:</p> <ol style="list-style-type: none"> 1) the test strategy is determined based on a risk assessment, because this is important for optimizing the test effort (that is, test coverage); 2) an analysis is made of what, where, and how much is to be tested to find the optimal balance between the desired quality and the amount of time/money required; 3) an optimization has taken place to distribute resources among the test activities. <p>(Strategy for single high-level test - priority 1)</p>
Base Practices	<p>TPI.1.BP1: Consider product risks. A motivated consideration of the product risks takes place, for which knowledge of the system, its use and its operational management is required. [Outcome 1]</p> <p>TPI.1.BP2: Differentiate test depth based on risk. There is a differentiation in test depth, depending on the risks and, if present, the acceptance criteria: not all subsystems are tested equally thoroughly and not every quality characteristic is tested (equally thoroughly). [Outcomes 1, 2, 3]</p> <p>TPI.1.BP3: Use test specification techniques. One or more test specification techniques are used, suited to the required depth of a test. [Outcome 3].</p> <p>TPI.1.BP4: Determine a retest strategy. For retests also, a (simple) strategy determination takes place in which a motivated choice of variations between 'test solutions only' and 'full retest' is made. [Outcomes 1, 2]</p>

Improvement Steps	<p>Involve the various interested parties such as end user, systems manager, and project manager in determining the test strategy.</p> <p>Create awareness by indicating the risks of the current working method, or indicate how testing can be done more cheaply and/or faster.</p> <p>Determine and maintain a regression test which is used to test new releases of the system for regression. Such a test is often composed of already existing test cases and is suitable for automation.</p> <p>Eventually perform the complete determination of the strategy. Here is a short description of the steps that must be taken to reach a test strategy:</p> <ul style="list-style-type: none"> * Determine quality characteristics * Determine relative importance of quality characteristics * Break down into subsystems * Determine relative importance of system parts * Determine the measuring techniques to be used
Dependencies	<ul style="list-style-type: none"> * TPI.5 Test specification techniques, informal techniques * TPI.11 Commitment and motivation, assignment of budget and time

Output work products
<p>08-21: Software test plan [Outcomes 1, 2, 3] 08-23: Validation test plan [Outcomes 1, 2, 3] 08-15: Regression test plan [Outcomes 2, 3] 15-9: Risk status report [Outcome 1] 17-14: Test case specification [Outcomes 2, 3] 19-10: Verification strategy [Outcome 3] 19-11: Validation strategy [Outcome 3]</p>

TPI.2 Life-cycle model

Process ID	TPI.2
Process Name	Life-cycle model
Process Purpose	The purpose of the Life-cycle model process is to distinguish a number of phases consisting of activities; for each activity its aim, the products to be delivered, and how the activity has to be performed are described, what makes the test process manageable.
Process Outcomes	As a result of successful implementation of this process: 1) the phases planning, specification and execution are distinguished and performed subsequently; 2) a test plan is defined in the planning phase defining how, by whom, with what and when the test activities are to be executed; 3) in the specification phase the test cases are defined and the test execution is prepared. In addition the realization of the infrastructure is provided; 4) in the execution phase the specified tests are performed to gain insight into the quality of the test object. (Planning, specification, execution - priority 1)
Base Practices	TPI.2.BP1: Distinguish between different test phases. For the test (at least) the following phases are distinguished: planning, specification, and execution. These are subsequently performed, possibly per subsystem. A certain overlap between the phases is allowed. [Outcomes 1, 2, 3, 4]
Improvement Steps	Focus attention on the test process staying on the critical path of the project for as short a time as possible. Usually testing comes onto the critical path when the developer has delivered the software for test execution. Beforehand, the test process should have dealt with as many matters as possible that do not require the presence of the software to be tested. This makes the lead time of the total project as short as possible and/or there is a maximum amount of time available to perform all planned tests.
Dependencies	* TPI.11 Commitment and motivation, assignment of budget and time.

Output work products

03-07: Test data [Outcomes 1, 4] 08-21: Software test plan [Outcomes 1, 2] 08-23: Validation test plan [Outcomes 1, 2] 13-16: Change request [Outcomes 1, 4] 14-04: Test log [Outcomes 1, 4] 15-10: Test incident report [Outcomes 1, 4] 15-11: Defect report [Outcomes 1, 4] 17-14: Test case specification [Outcomes 1, 3] 17-07: Infrastructure requirements [Outcomes 1, 2]

Anhang B: Zuordnung SPICE-Levels - TPI®-Ebenen

TPI.1 Test strategy

SPICE process attributes (GPs)	TPI levels (checkpoints)
PA 3.1 Process definition attribute	1.B Combined strategy for high-level tests
GP 3.1.1 Define the standard process that will support the deployment of the defined process.	The result of the coordination is a coordinated strategy, which is put in writing. During the total test process, this strategy is controlled.
GP 3.1.1 Define the standard process that will support the deployment of the defined process.	Each high-level test determines its own test strategy, based on the coordinating strategy.
GP 3.1.2 Determine the sequence and interaction between processes so that they work as an integrated system of processes.	Coordination takes place between the different high-level tests, often the system, acceptance, and production acceptance tests, in the field of test strategy (risks, quality characteristics, area of consideration of the test, and planning).
GP 3.1.5 Determine suitable methods to monitor the effectiveness and suitability of the standard process.	Deviations from the coordinating strategy are reported, after which a substantiated adjustment to the coordinating strategy is made, based on the risks.

PA 4.2 Process control attribute	1.C Combined strategy for high-level tests plus low-level tests or evaluation
GP 4.2.1 Determine analysis and control techniques, appropriate to control the process performance.	Coordination takes place between the high-level tests and the low-level tests or the evaluation levels in the area of test strategy (risks, quality characteristics, area of consideration of the test/evaluation, and planning).
GP 4.2.2. Define parameters suitable to control the process performance.	Each high-level test determines, on the basis of the coordination, its own test strategy.
GP 4.2.2. Define parameters suitable to control the process performance.	(if applicable) Each low-level test determines, on the basis of the coordination, its own test strategy.
GP 4.2.2. Define parameters suitable to control the process performance.	(if applicable) Each evaluation level determines, on the basis of the coordination, its own evaluation strategy.
GP 4.2.3 Analyse process and product measurement results to identify variations in process performance.	The result of the coordination is a coordinated strategy, which is documented. During the total (evaluation and) test process this strategy is controlled.
GP 4.2.4 Identify and implement corrective actions to address assignable causes.	Deviations from the coordinating strategy are reported, after which, on the basis of the risks, a substantiated adjustment of the coordinating strategy is made.

PA 5.1 Process innovation attribute	1.D Combined strategy for all test and evaluation levels
GP 5.2.1 Assess the impact of each proposed change against the objectives of the defined and standard process.	Coordination takes place between the high level tests, the low-level tests and the evaluation levels in the area of test strategy (risks, quality characteristics, area of consideration of the test/evaluation, and planning).
GP 5.2.1 Assess the impact of each proposed change against the objectives of the defined and standard process.	The result of the coordination is a coordinating strategy, which is documented. During the total evaluation and test process this strategy is controlled.
GP 5.2.2 Manage the implementation of agreed changes to selected areas of the defined and standard process according to the implementation strategy.	Each high-level test determines its own test strategy on the basis of the coordination, such as described for level A.
GP 5.2.2 Manage the implementation of agreed changes to selected areas of the defined and standard process according to the implementation strategy.	Each low-level test determines its own test strategy on the basis of the coordination.
GP 5.2.2 Manage the implementation of agreed changes to selected areas of the defined and standard process according to the implementation strategy.	Each evaluation level determines its own evaluation strategy on the basis of the coordination.
GP 5.2.3 Evaluate the effectiveness of process change on the basis of actual performance against process performance and capability objectives and business goals.	Deviations from the coordinating strategy are reported, after which a substantiated adjustment of the coordinating strategy is made on the basis of the risks

TPI.2 Life-cycle model

SPICE process attributes (GPs)	TPI levels (checkpoints)
PA 2.1 Performance management attribute	2.B Planning, preparation, specification, execution and completion
GP 2.1.2 Plan and monitor the performance of the process to fulfill the identified objectives.	For high-level test the following phases are distinguished: planning, preparation, specification, execution, and completion. The phases are executed consecutively, possibly per subsystem. A certain overlap between the phases is allowed

Anhang C: Installationsanleitung für project>kit

Um in project>kit das TPI[®] SPICE Referenzmodell zu installieren, sind folgende Schritte auszuführen:

- Unter C:\Programme\methodpark\projectkit\tomcat\webapps\pkit\WEB-INF\classes die Datei *LocalPkit.properties* um folgende Zeilen erweitern:

```
process.metamodel.tpispice = TPI SPICE Reference Model

process.activity.artifact.output.tpispice = Output work
  products:
process.activity.artifact.genoutput.tpispice = General work
  products:
process.artifact.activity.output.tpispice = This work product
  is used in the following TPI processes:
process.artifact.activity.genoutput.tpispice = This work
  product is used in the following process attributes:

process.element.type.singular.activity.level = Level
process.element.type.plural.inword.activity.level = levels
process.element.type.singular.activity.indicator = Indicator
process.element.type.plural.activity.indicator = indicators
process.element.type.plural.inword.activity.indicator =
  indicators
process.element.type.children.activity.level = This level
  consists of the following {0}:
process.element.type.children.activity.indicator = This
  indicator consists of the following {0}:

process.xls.tpispice.activity.artifact.output = 0
process.xls.tpispice.activity.artifact.genoutput = G
process.xls.tpispice.artifact.activity.output = 0

process.xls.legend.tpispice.activity.artifact.output = Output
  work product of process
process.xls.legend.tpispice.activity.artifact.genoutput =
  General work product of indicator
process.xls.legend.tpispice.artifact.activity.output = Work
  product
```

- Unter C:\Programme\methodpark\projectkit\tomcat\webapps\pkit\WEB-INF\conf die Datei *PKitProcesses.xml* um das Metamodell für TPI[®]>SPICE (s. Anhang D) erweitern.
- *UpdateModel.bat* ausführen.
- Project>kit Server starten.
- Einloggen.
- Unter „Projects“ – „Reference Models“ – „New Subprojects“ ein neues Projekt anlegen.
- In der „Process Workbench“ bei „Import process description“ die Zip-Datei mit der Prozessbeschreibung (ist nicht auf CD enthalten) importieren.

Anhang D: Metamodell für project>kit

```

<processtype ident="tpispice" referencemodel="true">
  <element ident="activity" hierarchy="true" modalities="true"
    subtypes="process basepractice activity level indicator"
    shortname="true" indexStyle="tree">
    <model>
      <associations>
        <association target="artifact" types="output genoutput" />
      </associations>
      <resources />
    </model>
    <view>
      <subtype ident="activity">
        <description />
        <subelements />
      </subtype>
      <subtype ident="level">
        <description />
        <subelements />
      </subtype>
      <subtype ident="process">
        <table border="1" cellspacing="3" cellpadding="10"
          width="100%">
          <tr valign="top">
            <td width="100%">
              <description />
              <subelements />
            </td>
          </tr>
          <tr valign="top">
            <td width="100%" colspan="2">
              <heading>
                <message key="artifact.output.tpispice" />
              </heading>
              <assoc target="artifact" type="output" style="list"
                />
            </td>
          </tr>
        </table>
      </subtype>
      <subtype ident="basepractice">
        <table border="1" cellspacing="3" cellpadding="10"
          width="100%">
          <tr>
            <td colspan="2">
              <description />
            </td>
          </tr>
        </table>
      </subtype>
      <subtype ident="indicator">
        <table border="1" cellspacing="3" cellpadding="10"
          width="100%">
          <tr valign="top">

```

```

        <td width="100%">
            <description />
            <subelements />
        </td>
    </tr>
    <tr valign="top">
        <td width="100%" colspan="2">
            <heading>
                <message key="artifact.genoutput.tpispice" />
            </heading>
            <assoc target="artifact" type="genoutput"
                style="list" />
        </td>
    </tr>
</table>
</subtype>
</view>
</element>
<element ident="artifact" modalities="true" shortname="true"
    hierarchy="true" indexStyle="tree">
    <model>
        <associations>
            <association target="activity" types="output genoutput" />
        </associations>
        <resources />
    </model>
    <view>
        <table border="0" cellspacing="3" cellpadding="10"
            width="100%">
            <tr>
                <td colspan="2">
                    <description />
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <subelements />
                </td>
            </tr>
            <tr valign="top">
                <td width="100%">
                    <heading>
                        <message key="activity.output.tpispice" />
                    </heading>
                    <assoc target="activity" type="output" style="list" />
                    <heading>
                        <message key="activity.genoutput.tpispice" />
                    </heading>
                    <assoc target="activity" type="genoutput" style="list"
                        />
                </td>
            </tr>
        </table>
    </view>
</element>
</processtype>

```

Anhang E: Beiliegende CD

Die CD enthält:

- HTML-Version des erstellten Prozess-Assessment-Modells für TPI[®] nach SPICE (Die Prozesse nach TPI[®] und die Indikatoren für sämtliche SPICE-Levels sind unter „Activities“ eingeordnet. Unter „Documents“ befinden sich die Arbeitsprodukte.).
- Exceltabelle zur Unterstützung bei Assessments
- Zuordnungstabelle mit SPICE-Prozessattributen (Generische Praktiken) und TPI[®] - Ebenen (Kontrollpunkte)

Abkürzungsverzeichnis

BP	Base Practice
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
EFQM	European Foundation for Quality Management
ESPRIT	European Strategic Program for Research in Information Technology
GP	Generic Practice
GR	Generic Resource
GWP	Generic Work Product
HIS	Herstellerinitiative Software
HTML	Hypertext Markup Language
IEC	International Electrotechnical Commission
INTACS	International Assessors Certification Scheme
ISO	International Organisation for Standardization
iSQI	International Software Quality Institute
MMAST	Maturity Model for Automated Software Testing
PA	Process Attribute
PAM	Process Assessment Model
PCI	Process Capability Indicator
PPI	Process Performance Indicator
PRM	Process Reference Model
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SEI	Software Engineering Institute (Carnegie Mellon University)
SPI	Software Process Improvement
SPICE	Software Process Improvement Capability dEtermination
TAP	Testing Assessment Programme
TCMM	Testing Capability Maturity Model
TIM	Test Improvement Model
TMap	Test Management Approach
TMM	Testing Maturity Model
TOM	Test Organisation Maturity Model
TPI	Test Process Improvement
TQM	Total Quality Management
TR	Technical Report
TSM	Testability Support Model
WP	Work Product
XML	Extended Markup Language

Abbildungsverzeichnis

Abbildung 2.1: Das Dreieck zu Kosten, Zeit und Qualität (s. [Kneu03]).....	10
Abbildung 2.2: Die Reifegradstufen bei CMMI (nach [Kneu03]).....	13
Abbildung 2.3: Übersicht der Reifegradmodelle (u.a. nach [Balz98]).....	14
Abbildung 2.4: ISO/IEC 15504 im Überblick (aus [ISO05]).....	15
Abbildung 2.5: Fähigkeitsstufen und Prozessattribute bei SPICE (nach [ISO05]).....	16
Abbildung 2.6: Die zentrale Rolle von Prozess-Assessments (nach [ISO05], [Balz98])....	17
Abbildung 2.7: Bestandteile eines SPICE-Assessments (nach [ISO05]).....	17
Abbildung 2.8: Fehlerkosten in verschiedenen Projektphasen (nach [Balz98])	19
Abbildung 2.9: Aufbau des TPI-Modells (nach [PKSp02]).....	22
Abbildung 2.10: Die TPI [®] -Matrix (aus [PKSp02]).....	23
Abbildung 2.11: Der Änderungsprozess bei TPI [®] (nach [PKSp02])	24
Abbildung 2.12: Beispiel des Ist-Zustands einiger Kernbereiche (aus [PKSp02])	25
Abbildung 2.13: Beispiel eines Soll-Zustands einiger Kernbereiche (aus [PKSp02]).....	25
Abbildung 2.14: Die Reference Compliance Workbench von project>kit	26
Abbildung 2.15: Assoziationen von Prozesselementtypen (aus [Meth06])	27
Abbildung 3.1: Normative Elemente des Prozess-Referenzmodells (aus [ISO05]).....	30
Abbildung 3.2: Normative Elemente des Bewertungsrahmens (aus [ISO05]).....	32
Abbildung 3.3: Zuordnung TPI [®] -Kategorien und SPICE-Levels	34
Abbildung 3.4: Assoziationen der Prozesselementtypen des Metamodells	40
Abbildung 3.5: Ausschnitt der Browserdarstellung eines Prozesses in project>kit.....	42
Abbildung 4.1: Referenzmodell-Mapping in project>kit.....	44

Tabellenverzeichnis

Tabelle 2.1: Reife und unreife Organisationen (nach [RoSF05]).....	11
Tabelle 2.2: Die Reifegradstufen im TMM (nach [Veen02]).....	21
Tabelle 3.1: Vergleich der Modelle SPICE und TPI [®]	29
Tabelle 3.2: Abbildung der TPI [®] -Elemente auf ein SPICE-PRM.....	32
Tabelle 3.3: Prozessattribute und Reifegradstufen bei SPICE (nach [ISO05]).....	35
Tabelle 3.4: Zuordnung der Bewertungsskalen von TPI [®] und SPICE	35
Tabelle 3.5: Abbildung der TPI [®] -Ebenen auf SPICE-Levels	38
Tabelle 4.1: Übersicht der Assessment-Ergebnisse.....	45

Literaturverzeichnis

- [Balz98] Balzert, H.: *Lehrbuch der Software-Technik – Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Spektrum Akademischer Verlag, Heidelberg, 1998.
- [Brab06] Braband, J.: *Kompetenzmanagement nach People CMM in der Eisenbahnindustrie*. SIGNAL + DRAHT (98), S.6-8, 3/2006, http://www.eurailpress.com/archiv/showpdf.php?datei=/erparchiv/sud2006/06_08_Braband.indd.pdf, letzter Abruf Juni 2007.
- [BuSC96] Burnstein, I.; Suwannasart, T.; Carlson, C.: *Developing a Testing Maturity Model: Part I*. Illinois Institute of Technology, 1996, <http://www.improveqs.nl/pdf/Crosstalk%20TMM%20part%201.pdf>, letzter Abruf Juni 2007.
- [CHKK04] Chodura, H.; Hofmann, P.; Kalusche, B.; Knoblach, J.; Spohr, J.; Weber, T.: *Standardisierung im Automotive-Umfeld*. Elektronik Automotive 4/2004, S. 48-52, 2004, http://www.automotive-his.de/download/auto4-04_S.48.pdf, letzter Abruf Juni 2007.
- [CVWC01] Cass, A.; Völcker, C.; Winzer, L.; Carranza, J.; Dorling, A.: *SPICE for SPACE: A Process Assessment and Improvement Method for Space Software Development*. esa bulletin 107, S. 112-119, 2001, http://www.esa.int/esapub/bulletin/bullet107/bul107_14.pdf, letzter Abruf Juni 2007.
- [Dorn04] Dorn, R.: *TPI Automotive – Aus der Sicht der Herstellerinitiative Software*. HIS AK Softwaretest, Präsentation. <http://www.tpiautomotive.de/Docs/HIS%20Vortrag%20TPI%20Automotive.pdf>, letzter Abruf Juni 2007
- [ErSU96] Ericson, T.; Subotic, A; Ursing, S.: *TIM – A Test Improvement Model*. 1996, <http://www.lucas.lth.se/events/doc2003/0113A.pdf>, letzter Abruf Juni 2007.
- [Gerr07] Gerrard consulting: *Test Organisation Maturity Questionnaire V2.0*. www.gerrardconsulting.com, letzter Abruf Juni 2007.
- [Gros04] Grosch, U.: *ISO 15504 (SPICE) und Automotive TPI*. Q-Labs, 2004, <http://www.TPIautomotive.de/Docs/SPICE%20-%20TPI%20Automotive%20-%20Q-Labs.pdf>, letzter Abruf Juni 2007.
- [HDHM06] Hörmann, K.; Dittmann, L.; Hindel, B.; Müller, M.: *SPICE in der Praxis – Interpretationshilfe für Anwender und Assessoren*. dpunkt.verlag, Heidelberg, 2006.
- [Huyn02] Huynh, D.: *Software Testing Maturity Model (SW-TMM)*. <http://www.cs.umd.edu/~atif/Teaching/Fall2002/StudentSlides/Duy.pdf>, letzter Abruf Juni 2007.

- [Isel05] Iseler, M.: *Konzept und Implementierung einer Web-Anwendung zur Verwaltung und Auswertung von ISO/IEC-15504-kompatiblen Prozessbewertungen*. Friedrich-Alexander Universität Erlangen-Nürnberg, 2005.
- [ISO05] ISO/IEC 15504-1: *Information technology – Process assessment – Part 1: Concepts and vocabulary* Ausgabe 2005..
ISO/IEC 15504-2: *Information technology – Process assessment – Part 2: Guidance on performing an assessment*. Ausgabe 2005.
ISO/IEC 15504-3: *Information technology – Process assessment – Part 3: Performing an assessment*. Ausgabe 2005.
ISO/IEC 15504-4: *Information technology – Process assessment – Part 4: Guidance on use for process improvement and process capability determination*. Ausgabe 2005.
ISO/IEC 15504-5: *Information technology – Process assessment – Part 5: An exemplar Process Assessment Model*. Ausgabe 2005.
- [Kneu03] Kneuper, R.: *CMMI – Verbesserung von Softwareprozessen mit Capability Maturity Integration*. dpunkt.verlag, Heidelberg, 2003.
- [KoBa05] Koomen, T.; Baarda, R.: *TMap Test Topics*. UTN Publishers, Den Bosch, 2005.
- [Koom02] Koomen, T.: *Report: Worldwide Survey on Test Process Improvement*. Sogeti Nederland, 2002, http://www.gend.nl/images/Report_TPI_survey_2004_v1%5B1%5D.0_tcm6-30299.pdf, letzter Abruf Juni 2007.
- [KoPo99] Koomen, T.; Pol, M.: *Test Process Improvement – A practical step-by-step guide to structured testing*. Addison-Wesley, Harlow, 1999.
- [Meie06] Meier, E.: *Angemessene Entwicklungsprozesse*. http://www.sigs.de/publications/os/2006/BPM/meier_OS_BPM_06.pdf, 2006, letzter Abruf Juni 2007.
- [MeSc06] Metz, P.; Schedl, G.: *Automotive-SPICE und IEC 61508 – Synergie oder Widerspruch?*. VDE VDI GI Tagung „Automotive – Safety & Security 2006“, 2006. <http://www.automotive2006.de/programm/Metz.pdf>, letzter Abruf Juni 2007.
- [Meth05] method park Software AG: *project>kit Produktbeschreibung V 3.0*. Technische Dokumentation 2005.
- [Meth06] method park Software AG: *project>kit Metamodeling Guide V 3.0*. Technische Dokumentation 2006.
- [Mroz06] Mrozek, M.: *Integration von V-Modell XT in ein Projekt- und Prozessmanagementsystem*. Georg-Simon-Ohm Fachhochschule Nürnberg, 2006.
- [NoBr04] Notenboom, E.; Brands, A.: *TPI Automotive das Modell*. sogeti Deutschland GmbH, 2004, <http://www.TPIautomotive.de>, letzter Abruf Juni 2007.

- [Nöss04] Nössler, O.: *ISO 15288, 12207 und CMMI. – Brauchen wir Qualitätsmodelle bei der Prozessverbesserung?*. wibas IT Maturity Services GmbH, http://www.wibas.de/download/BITKOM_v1.pdf, 2004, letzter Abruf Juni 2007.
- [PKSp02] Pol, M.; Koomen, T.; Spillner, A.: *Management und Optimierung des Testprozesses – Ein praktischer Leitfaden für erfolgreiches Testen von Software mit TPI und TMap*. dpunkt.verlag, Heidelberg, 2002.
- [Prec99] Prechelt, L.: *Ausgewählte Kapitel der Softwaretechnik – Skriptum zur Vorlesung*, Karlsruhe, <http://page.mi.fu-berlin.de/~prechelt/swt2/node41.html>, letzter Abruf Juni 2007.
- [RaFB04] Raak, C.; Foegen, M.; Battenfeld, J.: *CMMI, SPICE, ISO 15504 – Keine Gegensätze sondern ergänzende Standards*. wibas IT Maturity Services GmbH, <http://www.wibas.de/download/CMMIundSPICE.pdf>, 2004, letzter Abruf Juni 2007.
- [RoDo05] Rout, T.; Dorling, A.: *ISO/IEC 15504 (SPICE) – A Status Report*. Klagenfurt, 2005, <http://www.qual-it-consulting.it/risorse/SPICE%202005%2015504%20Status%20Report.pdf>, letzter Abruf Juni 2007.
- [RoSF05] Rösel, A.; Sazama, F.; Foldenauer, J.: *Erfahrungen und Empfehlungen in der Prozessverbesserung*. Q-Labs, 2005, http://www.bitkom.org/files/documents/Top_2_Roesel_Erfahrungen_und_Empfehlungen_in_der_Prozessverbesserung_18.11.2005.pdf, letzter Abruf Juni 2007.
- [Soge04] sogeti Deutschland GmbH, *TPI Automotive – Test Process Improvement, Erratum for version 1.0*. <http://www.TPIautomotive.de>, letzter Abruf Juni 2007.
- [SpLi03] Spillner, A.; Linz, T.: *Basiswissen Softwaretest – Aus- und Weiterbildung zum Certified Tester*. dpunkt.verlag, Heidelberg, 2003.
- [SRWL06] Spillner, A.; Roßner, T.; Winter, M.; Linz, T.: *Praxiswissen Softwaretest – Testmanagement – Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB-Standard*. dpunkt.verlag, Heidelberg, 2006.
- [Swin00] Swinkels, R.: *A comparison of TMM and other Test Process Improvement Models*. Frits Philips Institute, 2000, <http://is.tm.tue.nl/research/v2m2/wp1/12-4-1-FPdef.pdf>, letzter Abruf Juni 2007.
- [Tarn06] Tarnowski, M.: *Ausweg aus der Sandwich-Falle – Bessere Software-Produkte mit SPICE und CMMI*. QZ, Carl Hanser Verlag, München, S. 20-23, 2006, http://www.q-labs.de/data/presse/q-labs_artikel_ausweg_aus_der_sandwich_falle_qz2_2006.pdf, letzter Abruf Juni 2007.
- [Veen02] van Veenendaal, E.: *The Testing Practitioner*. UTN Publishers, Den Bosch, 2002.

- [WeRa06] Wentzel, P; Rastofer, U.: *Requirements Engineering – Leitfaden zum neuen SPICE*. Automotive 9/2006, S.48-53.
- [Wiki07a] Wikipedia: *ISO 12207*. http://de.wikipedia.org/wiki/ISO_12207, letzter Abruf Juni 2007.
- [Wiki07b] Wikipedia: *ISO 9000*. http://de.wikipedia.org/wiki/ISO_9000, letzter Abruf Juni 2007.
- [Wiki07c] Wikipedia: *CMMI*. <http://de.wikipedia.org/wiki/CMMI>, letzter Abruf Juni 2007.
- [Wiki07d] Wikipedia: *TQM*. <http://de.wikipedia.org/wiki/TQM>, letzter Abruf Juni 2007.

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Sabine Link

Erlangen, Juni 2007