



NACHLESE

3RD WORLD CONFERENCE FOR SOFTWARE QUALITY, MÜNCHEN 2005

ÜBERBLICK UND DETAILS ZU EINIGEN AUSGEWÄLTEN VORTRÄGEN

ÜBERBLICK

Bei der vom 27. – 29.09.2005 in München abgehaltenen 3rd World Conference for Software Quality wurden über drei Tage hinweg in vier Tracks am Vormittag und am Nachmittag jeweils drei Vorträge geboten, in den Mittagspausen ergänzten in „Industrial Sessions“ die Ausstellerfirmen das Programm. Eingeleitet wurde die Konferenz am 26.09.2005 mit insgesamt 13 Halbtagestutorials. Nicht vergessen werden dürfen auch die Keynotes, die jeden Konferenztag eingeleitet oder abgeschlossen haben.

Die Vorträge deckten folgende Schwerpunkte ab:

- Projektmanagement,
- Engineering Process Philosophies
- Integration and Testing
- Quality Models
- Engineering Process Benchmarking
- Review Techniques
- Reserach Sessions

- Project Retrospectives
- Software Process Benchmarking
- Requirements Engineering
- Testing
- Estimation
- Embedded Software
- Measurement and Metrics
- Risk Management
- Methodology and Tool Rollout
- Selection and Tailoring
- Model Based Quality
- Human Management
- Agile Methods
- Configuration Management

Wie man anhand der Fülle an Themen und Vorträgen leicht erkennt, können in dieser Nachlese nur einige wenige Vorträge detaillierter betrachtet werden.

ENGINEERING PROCESS BENCHMARKING

PRACTICAL EXPERIENCE WITH THE NEW ISO/IEC 15504, MARKUS MÜLLER, KUGLER MAAG CIE

Die wesentlichen Änderungen vom Technical Report (TR) zum International Standard (IS) sind:

- Fokuswechsel von TR **Software Process Assessment** zu IS **Process Assessment**
- Strukturelle Änderung: 5 statt 9 Teile
- Alternative Prozessreferenzmodelle dürfen verwendet werden
- Die ISO/IEC 155054 enthält kein standardisiertes Referenzmodell mehr

Ein Referenzmodell kann z.B. die ISO/IEC 12207:2002/AM1 Amendment to ISO/IEC 12207:1995 Information Technology – Software Life Cycle Processes sein.

Wesentlicher konzeptioneller Punkt der Norm ist die Unterscheidung zwischen Referenzmodell und Assessment Modell.

Das **Prozess-Referenzmodell**

- muss die Forderungen der Norm ISO/IEC IS 15504-2 (Section. 6.2) erfüllen

- enthält allgemeine Prozessbeschreibungen, die in Assessment-Modellen detailliert werden
- muss sicherstellen, dass das Assessment-Modell mit dem in der ISO/IEC IS 15504-2 definierten Capability Level und Zielen übereinstimmt

Das **Assessmentmodell**

- basiert auf einem oder mehreren Prozessreferenzmodellen
- enthält die detaillierten Prozessbeschreibungen
- dient als Basis sowohl für die Sammlung der Erkenntnisse als auch für die Bewertung der Process Capability

Änderungen der Process Dimension in Kürze

- Die Process Dimensions wurden auf der ISO/IEC 12207:1995/Amd1:2002 basierend erweitert und geändert
- Die Engineeringprozesse wurden restrukturiert
- Acquisition und Supply wurden in zwei Gruppen aufgeteilt
- neue Prozessgruppen für Reuse and Product Quality wurden eingeführt



- der Change Request Management Process wurde aufgenommen
- Änderungen bei Base Practices und Work Products wurden vorgenommen

Änderungen der Capability Dimension

- Änderungen der Struktur
 - aus Management Practices werden Generic Practices
 - neu hinzugekommen sind Generic Resources und Generic Work Products
- Inhaltlich
 - Level 1 fokussiert explizit auf die Process Outcomes
 - Level 2 Änderungen bei den Generic Practices
 - Auf Level 3 wurden die Process Attributes neu organisiert

Änderungen auf Level 1

- Die Process Outcomes werden stärker betont
- Outcomes und Base Practices sind nicht immer homogen
- Falls Assessoren die Process Outcomes statt der Base Practices zur Bewertung heranziehen, kann sich das Rating des Process Capability Attributes auf Level 1 verändern

Änderungen auf Level 2

- Wesentlich präzisere Beschreibung der Base Practices, vor allem durch die Aufzählungen in jeder Practice
- Ergänzung um zusätzliche Outcomes für „Prozeßschnittstellen“ und zu Ressourcen und Information zur Unterstützung des Prozesses
- Verbesserungen hinsichtlich Configuration Management und Lenkung der Work Products
- Nach wie vor Schwierigkeiten, Details und die neuen Erläuterungen mit den Generic Practices zu mappen
- Insgesamt ist der IS eine wesentliche Erleichterung für die Assessoren beim Rating

Änderungen auf Level 3

- PA3.1 Process Definition fokussiert nun wesentlich klarer auf die Prozessdefinition als zuvor
- PA3.2 Process Deployment (neu) hat nun das Deployment des Standardprozesses als Defined Process zum Thema

Der Vortragende empfiehlt, bereits jetzt, beide Versionen TR und IS (ISO/IEC IS 15504 Part 2-4) zu berücksichtigen und das Assessment Model (ISO/IEC IS 15504-5) so rasch wie möglich nach der nach Publizierung anzuwenden.

APPLYING SPICE IN THE FINANCE SECTOR, DR. CHRISTIAN VÖLCKER, SYNSPACE SA, JOHANN LEIBIG, HVB SYSTEMS GMBH

Kostenreduktion zwingt die Banken ihre IT-Prozesse weiter zu optimieren. Die HVB nutzt dazu SPICE-Assessments. Der Vortrag beschreibt das Vorgehen bei HVB Systems.

Das Top Management wählte den durchschnittlichen SPICE Capability Level als eine der Schlüsselmetriken für die Balanced Scorecard (BSC). Für jedes Jahr wird ein minimaler durchschnittlicher Capability Level als zu messendes Ziel für die BSC definiert, der bei Erreichen einen Bonus für die bewertete Abteilung ergibt.

In 7 Monaten wurde fast 70 SPICE Assessments durchgeführt. Das Ziel der Assessments war nicht nur, BSC-Daten zu gewinnen, sondern auch tiefere und detaillierte Erkenntnisse über die Performance und Angemessenheit des Standardprozesses zu gewinnen und Improvement-potentiale für den Capability Level 3 zu ermitteln.

Die Assessmentmethode wurde angepasst, um die Gesamt- und Interviewdauer zu verkürzen. Weiters wurden nur die für die Organisation relevanten Prozesse betrachtet.

Der Assessmentprozess wurde durch Pre-Assessment-Fragelisten und den automatischen Import dieser Antworten in eine Datenbank, Verwendung von Videoprojektoren während der Interviews und elektronischen Zugriff auf die relevante Dokumentation um 30% reduziert (Basis Erfahrungswerte SYNSPACE).

Der Assessment-Report jedes einzelnen Assessments enthielt das Prozessprofil und die Capability Level, eine Liste der Stärken und Verbesserungsvorschlägen sowohl für das Projekt als auch den Standardprozess.

Durch den starken Managementsupport war das Interesse an den Verbesserungsvorschlägen enorm. Workshops auf Abteilungs- und Organisationsebene wurden abgehalten und insgesamt um die 1000 Verbesserungsvorschläge in ca. 50 allgemeine Empfehlungen für die Prozessanwender und ca. 40 für die Verantwortlichen der Standardprozesse zusammengefasst.

Das Management der HVB Systems hat einige entscheidende Vorteile der Assessments für Kunden, Mitarbeiter und das Unternehmen erkannt und wird den SPICE-Ansatz weiter verfolgen.



STRATEGISCHE ANSÄTZE ZUR PROZESS- UND QUALITÄTSVERBESSERUNG IN ASIEN

A STRATEGY FOR THE IMPROVEMENT OF SOFTWARE QUALITY AND PRODUCTIVITY IN JAPAN, KEIGO YAMADA, JAPANESE MINISTRY OF ECONOMY, TRADE AND INDUSTRY (METI)

Das japanische Wirtschafts-, Handels- und Industrieministerium hat gemeinsam mit der Industrie und Wissenschaft des Landes eine Initiative, das Software Engineering Center (SEC), zur Verbesserung der Software Development Capability ins Leben gerufen. Für das Jahr 2005 stehen dem SEC 20 Mio US\$, 26 Mitarbeiter als Core Staff und weitere 130 Mitarbeiter zur Verfügung. Das SEC kooperiert mit dem Fraunhofer Institut. Ziel des SEC ist es, eine Plattform für die Software Development Capability zu etablieren. Das SEC hat drei Projekte voranzubringen:

- Software Engineering for Enterprise Systems (SEES)
- Software Engineering for Embedded Systems (SEET) und
- Advanced SW-Development Project

Die Schwerpunkte im SEES-Bereich liegen auf

- Requirements Engineering
- Quantitative Data Analysis
- Estimation Methods
- Software Lifecycle Processes

Die Schwerpunkte im SEET-Bereich liegen auf

- Quality Improvement Technique
- Project Management Technique
- Process Technique

Das Advanced SW-Development Project hat die Entwicklung einer neuen IT-Plattform zum Ziel, durch die Busse, Lastkraftwagen und Privatfahrzeuge vernetzt mit Informationen versorgt werden können.

In einzelnen Bereichen (z.B. Estimation) liegen bereits White Paper vor, eine Übersetzung ins Englische ist geplant.

QUALITY IMPROVEMENT PRACTICE OF OFFSHORE DEVELOPMENT IN CHINA, WU ANHUA, NEC

Der Status Quo der Offshore-Entwicklung in China

Mit großer Dynamik und Unterstützung durch die chinesische Regierung engagieren sich viel chinesische Firmen in der Offshore Software-Entwicklung. Wegen der ähnlichen Kultur, langer historischer sowie geografischer Verbindungen zwischen China und Japan und langer Kooperationen mit japanischen Firmen, sind ca. 60% der chinesischen Offshore-Entwicklung von Japan initiiert.

Die meisten Offshore-Firmen sind klein und hauptsächlich mit Codieren und Testen beauftragt. China hat vier Capability-Level für das Offshore Development definiert:

- Stufe 1: Übernahme von Codier- und Testprojekten, die von Japan ausschließlich zur Kostensenkung vergeben werden.
- Stufe 2: Software Know-How-Transfer von japanischen Firmen, Einführung in das japanische Projektmanagementsystem und Entwicklung von Software einem Standardprozess folgend
- Stufe 3: Verstehen der japanischen Kundenanforderungen; Durchführung großer Softwareprojekte mit Unterstützung japanischer Firmen
- Stufe 4: Erwerb der Fähigkeiten, große Softwareprojekte selbständig durchzuführen (Endziel)

Aktuell sind folgende Themen zu bewältigen

- Einführung von den internationalen Standards entsprechenden Entwicklungsprozessen
- Teilnahme am globalen Markt, um die Entwicklungsfähigkeit, insbesondere die Designfähigkeit zu stärken
- Verbesserung des Projektmanagements, um die Qualität und damit den Erfolg der Entwicklung sicherzustellen
- Verbesserung der Fremdsprachenkenntnisse (Japanisch), um mit den Kunden gut kommunizieren zu können.

Vor dem Start der Verbesserung (2002) hatte das Unternehmen 8 Jahre Erfahrung mit Offshore-Entwicklung, eine Anzahl von Basis-Entwicklungsstandards eingeführt und das ISO 9001-Zertifikat erworben. Die meisten Projekte starteten mit Detailed Design, bei steigender Tendenz bereits ab dem Funktionsdesign, d.h. die Firma hatte die Stufe 2 erreicht. Zur Stärkung der Entwicklungsfähigkeit und um die Stufen 2 und 3 zu erreichen, wurden folgende Themen angegangen:

- Verbesserung der Entwicklungsqualität, um homogene Ergebnisse zu erreichen
- Etablierung eines umfassenden Projektmanagementsystems
- Ausbildung von qualifizierten Offshore-Entwicklern (insbesondere japanische Sprachkenntnisse)

Ziel der Verbesserungsaktivitäten

Als Ziel wurde definiert, in eineinhalb bis zwei Jahren den Offshore Development Capability-Level 4 zu erreichen und Chinas Nummer 1 bei Qualität und Produktivität zu werden. Diese



Produktentwicklungsfähigkeit für alle Life Cycle Phasen (Design bis Test) soll durch Standardisierung des Entwicklungsprozesses und Vereinheitlichung des Entwicklungsmanagementsystems gelingen. Weitere Ziele sind:

- **Qualität:** Defektrate nach Auslieferung (defects/KLOC), wie die Offshore-Marktführer (ca. 0.022 defects/KLOC).
- **Produktivität:** code line per person hour of development (Loc/HH). Da viele Projekte mission-critical Produkte herstellen, werden die Ziele vom Produktivitätsziel der militärischen mission-critical Software der USA übernommen
- **Japanischkenntnisse:** die Anzahl der Entwickler mit Sprachkenntnissen des Level 2 und höher (Japanese Language Proficiency Test) auf höchstes Niveau in China zu bringen

Verbesserungsmaßnahmen

Die Verbesserungsmaßnahmen wurden in zwei Kategorien unterteilt

- **allgemeine** zur Erhöhung der Entwicklungsfähigkeit im gesamten Unternehmen und
- **individuelle** zur Lösung spezieller Probleme in jedem Projekt.

Die allgemeinen Maßnahmen umfassten die Perfektion des Entwicklungsprozessstandards, Festlegung des einheitlichen Projektmanagementsystems und Verbesserung der Kommunikationsfähigkeit.

Die Optimierung des Entwicklungsprozessstandards zur Verbesserung und Vereinheitlichung der Qualität beinhaltete

- Verbesserung und Verfeinerung des vorhandenen Prozesses (auch Tailoring) und Bereitstellung von Checklisten, Vorlagen und guten Beispielen von Spezifikationen und Berichten für die einfachere Nutzung
- Genauere Vorhersage und Analyse von Fehlern und bessere Qualitätslenkung des Prozesses auf Basis des überarbeiteten Standards
- Einbeziehung von Entwicklern in die Prozesseinführung, die anschließend als Trainer fungieren

Die Etablierung des einheitlichen Projektmanagementsystems wird erreicht durch

- Definition der Elemente der Baselines und der Methoden für die Datenanalyse
- Einführung von Qualitäts- und Fortschrittsverfolgungs-, Prozessmanagement- und Fehlermanagementtools, wodurch sich die Managementfähigkeit deutlich erhöhte
- Diskussionsrunden zum Erfahrungsaustausch

Die Verbesserung der Trainings zur Qualifikationssteigerung der Entwickler umfasste

- Verbessertes Trainingssystem zum Erlernen der japanischen Sprache und Einsatz der Sprache im Tagesgeschäft
- Entsendung der Entwickler zum Kunden

Die individuellen Maßnahmen für die Projekte umfassten

- Schulung spezieller Techniken wie auch zum Projekthintergrund
- Anpassung des Entwicklungsprozesses
- Festlegen quantitativer Ziele für Qualität und Produktivität

Umsetzung der Verbesserungsmaßnahmen

Zur Umsetzung der Verbesserungsmaßnahmen wurden Zwischenziele gesetzt und der PDCA-Zyklus in den Projekten angewendet, die ihre Ziele nicht erreichen konnten.

Ergebnisse der Verbesserungsmaßnahmen

Folgende quantitative Ergebnisse wurden nach ca. zwei Jahren erreicht:

- 5,9-fache Steigerung der Qualität (Defektrate nach Auslieferung)
- 2,4 fache Steigerung der Produktivität
- 3,1 fache Steigerung der Japanischkenntnisse

Der Status wird halbjährlich gemessen und vom Top Level Management überprüft.

Diese qualitativen Ergebnisse ermöglichen es nun, unabhängig hoch qualitative Projekte durchzuführen und die Software zeitgerecht zu liefern:

- Homogene Qualität und stabile Qualitätskennzahlen, wie Designqualität, Testqualität, etc, die im ersten Halbjahr 2004 in 95% der Projekte eine Schwankungsbreite von unter $\pm 2\sigma$ erreicht haben.
- Qualität und Produktivität haben die Zufriedenheit des Kunden erreicht
- Projekte starten nun auch mit dem Funktionsdesign oder noch früheren Phasen
- Der Einsatz des PDCA-Zyklus unterstützt den weiteren Verbesserungsprozess.

Auswirkungen auf das Unternehmen

Durch dieses Verbesserungsprogramm hat das Unternehmen die Level 4 Fähigkeit erlangt. Um die Verbesserung kontinuierlich weiter voranzutreiben wurde das Ziel gesetzt, den CMMI Level 5 zu erreichen. Mit März 2005 hat das gesamte Unternehmen das CMMI Level 5 Appraisal bestanden.



PROJEKTERFAHRUNGEN

TEST METHOD DEPLOYMENT, DIPL.-INF. HOLGER HANISCH, DATEV EG, NÜRNBERG

Der Autor stellt den Consulting-Ansatz seines Unternehmens zur Einführung von Testmethoden in Projekten dar.

Als Hauptgründe für die geringe Verbreitung von Testmethoden sieht er drei Gründe:

- Zu wenig Training: der Unterschied zwischen einer Methode (abstraktes Wissen) und der täglichen Praxis erfordert bessere Unterstützung beim Erlernen der Techniken und dem Wissenstransfer
- Nicht mein Stil: der häufig wissenschaftliche Publikationsstil erfordert die Übersetzung in praktische Schritte
- Nicht die richtige Zielgruppe: Der Graben zwischen technologischen Visionären und der Masse der Praktiker erfordert, dass eine neue Methode auch für die speziellen Geschäftsziele der Praktiker relevant und gerechtfertigt sein muss.

Im Testbereich wird das Vorgehen in folgenden Bereichen den jeweiligen Bedürfnissen des Kunden angepasst

- Umfang der technischen Unterstützung (ohne Tools, nur zusätzliche Tools, vollständige Automatisierung)
- Umfang der Analyse der Anforderungen für die Testfallvorbereitung (normal, linguistisch)
- Umfang der Testfall-Dokumentation (umfangreich für Anfänger, Kurzversion für Experten)
- Grad der Testautomatisierung (modularisierte Skripte bis zu Skriptsteuerung in Datentabellen)
- Anzahl der verfügbaren Testprozeduren (feste Anzahl für Anfänger, spezielle Kombination für Experten)

Der Besuch nur von Seminaren reicht für effektives Lernen nicht aus. Der Autor schlägt folgende Schritte vor

- Vorstellung der Methoden
- Training mit Anleitung
- Verfestigung und Reflektion
- Übergabe an den Kunden

Der Schritt „Verfestigung und Reflektion“ ist der wichtigste, da der Kunde die vorgestellten Methoden nicht nur mechanisch nutzen, sondern auch die Vor- und Nachteile im jeweiligen Projekt abwägen soll.

Der Testprozess wird wie folgt strukturiert:

- Auswahl des passenden Testlevel (Dokument, Module/Komponente, eine/mehrere Anwendungen)
- Aufteilung der Anwendung in einer Liste oder einem Baum mit Testgebieten (z.B. Use Cases), Priorisierung dieser Testgebiete
- Auswahl von vorab festgelegten Testarten (Funktionstest, Performance Test, ...) auf Basis der Qualitätsattribute
- Erstellen der Testanforderungen, abgeleitet aus den identifizierten Risiken je Testgebiet und Testart
- Prüfen auf Abhängigkeit der Testanforderungen und Darstellung der Abhängigkeiten in Entscheidungstabellen
- Implementierung von Testfällen zur Abdeckung der Testgebiete

Dieses Vorgehen unterstützt die starke und übersichtliche Strukturierung des Tests (Managementaspekte) bis hin zur konkreten Herleitung von Testfällen (Umsetzungsaspekte).

Zur Herleitung der Testfälle wird vom häufigsten Szenario ausgegangen, dann die Daten und anschließend die Pfade variiert, bis alle Testanforderungen abgedeckt sind.

PRACTICAL EXPERIENCES IN INTRODUCING EFFICIENT MANAGEMENT OF REQUIREMENTS AND A METHODOLOGICAL TEST APPROACH IN PARALLEL, EDGAR BRODDE, RDS CONSULTING, DÜSSELDORF

Der Vortragende schildert seine Erfahrungen bei der Rheinischen Bahngesellschaft AG in Düsseldorf.

Ausgangssituation war ein inhomogenes Entwicklungsmodell, unterschiedliche Methoden, Tools und Dokumentation sowie parallele Arbeit in Entwicklungs- und Wartungsprojekten. Dies alles resultierte in permanenter Unterbrechung der Arbeit, nie endenden Projekte, keiner Zeit für nichts und schon gar nicht für Dokumentation

Um ein für alle Teams geltendes Entwicklungsmodell zu etablieren, wurde ein Projekt mit den folgenden Phasen ins Leben gerufen:

- Kick-Off
- Detaillierte Analyse
- Umsetzung
- Coaching und Optimierung

Im Kick-Off wurde in Interviews ein erster Überblick über die Problemlage gewonnen und in einer Kurzanalyse das vermutete Verbesserungspotential bestätigt.



In der Detailanalyse wurden die einzelnen Teams und der Kunde intensiv befragt. Als zentrale Problemkreise kristallisierten sich folgende Themen heraus:

- Requirements Engineering
- Testmethoden
- Projektmanagement

Um die Wissensdefizite zu beseitigen, startete die Umsetzungsphase mit Workshops zu Software Engineering (V-Modell, XP, RUP), Requirements Engineering und Testmethoden. Trotz intensiver Diskussionen und dem Infragestellen der meisten der vorgestellten Methoden war man sich einig, dass die Probleme gelöst werden müssen.

Im nächsten Schritt diskutierten Gruppen von Freiwilligen in moderierten Workshops die erkannten Problemkreise und erarbeiteten Lösungen, die in einem Pilotprojekt getestet wurden. Die endgültige Version des Prozessmodells wurde allen Entwicklern vorgestellt.

Während der Coaching- und Optimierungsphase übernahmen die jüngeren Mitarbeiter die Techniken und diskutierten die Optimierung der Umsetzung, so dass nur wenig Coaching bei der Formulierung von Anforderungen und dem Entwurf von Testfällen notwendig war. Die langjährigen Entwickler argumentierten gegen die Neuerungen. Das Coaching dieser Mitarbeiter macht einen anderen Ansatz notwendig – sie müssen überzeugt werden. Das Projekt ist noch nicht beendet.

Das Projekt dauerte doppelt so lange als geplant, da die sozialen Probleme das Projekt blockierten:

- Das Verhalten der langjährigen Mitarbeiter zu ändern, war die schwierigste Aufgabe
- Die Kommunikation im Unternehmen war schnell und effizient – die Programmierer und

ihre Manager waren unzufrieden, aber der Kunde war glücklich, denn sein Geschäft funktionierte! So mussten die Änderungen möglichst nahtlos erfolgen.

- Beinahe niemand hatte zuvor gelernt, präzise Dokumentation effizient zu erstellen. Und beinahe niemand hatte die Erfahrung gemacht, dass schriftliche und gut strukturierte Dokumentation besser ist, als ein paar handschriftliche Notizen. Es bestand die Angst, dass Diskussionen mit dem Kunden in Konflikten enden würden, wenn man die Anforderungen vorlegt, die der Kunde ein paar Tage zuvor gestellt hatte.

Diesen Problemen wurde mit folgenden Maßnahmen begegnet:

- Die offiziellen Prozesse müssen von jedem Entwickler in jedem Projekt eingehalten werden
- Ausnahmen werden nur in speziellen Situationen akzeptiert, die Gründe müssen dokumentiert und mit dem verantwortlichen Manager diskutiert werden
- Der Umfang des Coaching wurde erhöht, insbesondere für die älteren Programmierer. Unterstützend halfen dabei die konsequente Nutzung von Vorlagen und Prozeduren.
- Die anderen Abteilungen wurden von den Veränderungen informiert, der Vorteil für alle herausgestrichen und die Kunden ermuntert, die Prozesse zu unterstützen.

Nach mehr als einem Jahr ist das Prozessmodell weitgehend akzeptiert. Alle Projekte laufen ziemlich ähnlich ab und die meisten Mitarbeiter haben erkannt, dass verwaltete Anforderungen und ein systematisches Testverfahren Zeit und Nerven sparen.

INTEGRATION AND TESTING

STRUCTURAL AND FUNCTIONAL PREDICATE COVERAGE TESTING, MARIO WINTER, MICHAEL AVERSTEGGE, UNIVERSITY OF APPLIED SCIENCES, COLOGNE

Die Autoren stellen die aus der Literatur bekannten Pfadabdeckungstestmethoden vor:

- Zweigüberdeckung
- Einfache Bedingungsüberdeckung
- Zweigüberdeckung mit einfacher Bedingungsüberdeckung
- Mehrfache Bedingungsüberdeckung
- Minimal Mehrfache Bedingungsüberdeckung.

Anschließend erläutern Sie die Spezifikationstechnik „Design by Contract“ (precondition,

postcondition, invariant) und zeigen, dass deren Bedingungen sehr komplex sein können. Daher macht es Sinn, die White-Box-Testtechniken anzuwenden.

In weiterer Folge wird das Verfahren „Functional Predicate Testing“ beschrieben. Im Funktionstest werden öffentliche Funktionen der Schnittstellen von Klassen, Subsystemen oder Systemen getestet. Als sinnvolle Reihenfolge für die Tests der Operationen geben die Autoren an:

- Konstruktoren
- Einfache Observer wie `getMember(): value`
- Einfache Modifier wie `setMember(value)`
- Komplexe Observer
- Komplexe Modifier



- Destruktoren (falls implementiert)

Um eine ausreichende Testabdeckung zu erreichen, wird die Minimal Mehrfache Bedingungsüberdeckung vorgeschlagen. Es sind solange Testfälle für die Operationen zu erstellen, bis alle erwarteten Ergebnisse der postconditions abgedeckt sind.

Die Autoren haben das Werkzeug WeSUF (Werkzeug für strukturelle und funktionale Bedingungsüberdeckungs-Tests von Java-Programmen mit Object Constraint Language-Spezifikation) zum Erstellen der Testfälle entwickelt. Das Werkzeug extrahiert und analysiert die predicates von contracts, die in OCL spezifiziert sind und errechnet ein minimales Set von Testfällen mit minimaler Wertekombination, um die gewünschte Abdeckung zu erzielen. Durch Codeinstrumentierung mit einem weiteren Tool und dem Sammeln und Speichern der erreichten Testabdeckung in einer Datenbank können die erzielten Ergebnisse vom WeSUF analysiert werden. Der Tester hat dann die Möglichkeit, nicht abgedeckte Pfade zu erkennen und die Tests entsprechend zu ergänzen. Reports in XML- und HTML-Format runden die Funktionalität des Tools ab.

Der Arbeitskreis TOOP "Testen objektorientierter Programme" der Gesellschaft für Informatik stellt auf seiner Homepage <http://toop.gi-ev.de> eine Onlineauswertung der Fehlerhäufigkeiten in objektorientierten Systemen zur Verfügung.

A COMPONENT-BASED REUSE TECHNIQUE OF SOFTWARE TEST CASES, ZHENYU LIU, SHANGHEI SOFTWARE CENTER

Dieser Vortrag stellt das Schema der Attribute von Testfällen in BNF-Notation vor. Das Schema wurde um Attribute zur Unterstützung der Wiederverwendung von Testfällen erweitert. Weiters werden der Reuse-Ansatz in einer Testfall-Bibliothek und das Suchen von Testfällen beschrieben. Die Informationen eignen sich sehr gut, sich rasch mit den Anforderungen an Testfallverwaltungstools und die Funktionalität hinsichtlich der Wiederverwendung von Testfällen vertraut zu machen.

TUTORIAL MEASURING THE QUALITY OF TESTING USING DDP (DEFECT DETECTION PERCENTAGE), DOROTHY GRAHAM, GROVE CONSULTANTS (UK)

Die Metrik DDP ist hauptsächlich eine a posteriori nutzbare Metrik. Zu Vorhersagen der zu erwartenden Produktionsfehler ist sie geeignet, zur Ermittlung des Testende-Kriteriums nicht.

Die Metrik besteht vor allem durch ihre einfache Berechnung:

$DDP = \frac{\text{in einer Teststufe gefundene Fehler}}{\text{Gesamtanzahl der Fehler ab einer Teststufe}}$

Beispiel:

System-Test (ST)	Abnahme-Test (AT)	Einsatz	ST DDP nach AT	ST DDP nach Einsatz	AT DDP nach Einsatz	DDP gesamt nach Einsatz
100	50	100	67%	40%	33%	60%

$DDP \text{ ST nach AT} = 100/150 = 67\%$

$DDP \text{ AT nach Einsatz} = 50/150 = 33\%$

Defect Detection Prediction kann in folgenden Fällen eingesetzt werden:

- Überwachung der Effektivität des Tests im Hinblick auf das Finden von Fehlern für unterschiedliche Teststufen (z.B. Systemtest, Abnahmetest) in unterschiedlichen Releases. Voraussetzung ist, dass Daten zu Test- und Produktionsfehlern vorhanden sind.
- Vorhersage von Fehlern in der Produktion („Reverse DDP“). Voraussetzung ist die Kenntnis der typischen DDP für ein Produkt und die Teststufen.
 - Beispiel: DDP 75%, 30 Fehler gefunden; dann folgt daraus, dass in der Produktion voraussichtlich 10 Fehler gefunden werden
 - Testen wird ab einer DDP von mehr als 80% wirklich effektiv (Quelle: Alistair Gerrard).
- Benchmarking

Zur Ermittlung der DDP ist vor allem wichtig, dass Fehlersammlung und Analyse konsistent erfolgen. Weder unterschiedliche Größe, Dauer, oder Technologie noch Lifecycle-Ansätze in Projekten haben Einfluss auf die DDP-Berechnung.

Sinnvoll einsetzbar ist DDP allerdings erst, wenn ausreichend Daten über mehrere Projekte und einen längeren Zeitraum hinweg verfügbar sind. DDP sollte nicht eingesetzt werden, um Personen oder zu kleine Gruppe und Zeiträume zu überwachen. Sie ist auch nicht geeignet, um Vorhersagen für das Testende zu treffen. Ebenfalls nicht eingesetzt werden sollte DDP, wenn

- zu wenige Fehler verfügbar sind,
- die Fehlerverwaltung unreif, sehr subjektiv, unzuverlässig ist oder gar nicht existiert
- es unmöglich ist, Fehlerdaten aus dem Einsatz zu erhalten

Trotz aller genannten Einschränkungen kann DDP Verbesserungen im Testprozess oder die Auswirkungen von Termindruck darstellen.