

3.3 Modellbasierter Software-Test – Generierung von Testfällen aus UML-Diagrammen

Prof. Dr. Mario Winter
 Telefon: +49-2261-8196-6285
 E-Mail: mario.winter@fh-koeln.de

Kooperationspartner:
 imbus AG, Möhrendorf, Dr. David Kreische

3.3.1 Einleitung

Unter dem Begriff der Qualitätssicherung (QS) versteht man die Summe aller Maßnahmen, die garantieren sollen, dass ein (entstehendes) Produkt die an es gestellten (Qualitäts-)Anforderungen erfüllt. Die Erkenntnis, dass Qualität nicht im Nachhinein in ein Produkt „hineingeprüft“ werden kann, hat zur Unterscheidung der konstruktiven Qualitätssicherung von der analytischen Qualitätssicherung geführt. Die konstruktive Qualitätssicherung sucht das Entstehen von Fehlern durch geeignete Maßnahmen während der Entwicklung zu verhindern. Sie basiert auf dem Einsatz von Methoden, Konstruktionsprinzipien, formalen Verfahren, Werkzeugen und Vorgehensmodellen. Allen Fortschritten auf diesem Gebiet trotzend häufen sich die Probleme aufgrund von Software-Fehlern. Dies liegt einerseits an der wachsenden Komplexität der modernen Softwaretechnologien und andererseits an der ständig wachsenden Durchdringung aller Gebiete der Technik durch Software – salopp gesagt macht, wer viel macht, auch viele Fehler. Die analytische Qualitätssicherung, auch als „Prüfen“ oder „Testen“ bezeichnet, versucht daher, entstandene Fehler möglichst früh zu entdecken und die erreichte Qualität zu bewerten. Im Mittelpunkt steht die Frage, ob der Prüfgegenstand (Teil, Zwischen- oder Endprodukt) bestimmte vorgegebene Qualitätskriterien erfüllt. Zu den Prüfgegenständen bei der Softwareentwicklung gehören z. B. die Anforderungsspezifikation, Entwurfsdokumente,

Diagramme, Quellcode, ausführbarer Code und die Testdokumente selbst. Bei der analytischen Qualitätssicherung unterscheidet man statische Prüfungen, welche den Prüfgegenstand nicht ausführen und somit auf alle Entwicklungsprodukte also auch z. B. Anforderungs- und Entwurfsspezifikationen anwendbar sind, und dynamische Prüfungen – oft einfach als „Software-Tests“ bezeichnet – welche „ausführbare“ Prüfgegenstände i. Allg. also Programmcode, erfordern und diese mit dem Ziel ausführen, Fehler zu finden. Dafür werden Testfälle benötigt, bei deren Ausführung für ausgewählte Eingaben das jeweilige Ergebnis der Ausführung mit dem erwarteten (d.h. vorher spezifizierten) Ergebnis verglichen wird. Das prinzipielle Problem der analytischen Qualitätssicherung besteht darin, dass erschöpfende dynamische Prüfungen wegen der Fülle der möglichen Eingaben und Programmzustände bei größeren Programmen nicht möglich sind. Testen als stichprobenartiges Verfahren kann somit nur die Anwesenheit von Fehlern aufzeigen, nicht aber deren Abwesenheit beweisen. Bei der Entwicklung komplexer Softwaresysteme hoher Qualität müssen somit konstruktive und analytische Tätigkeiten und die entsprechend eingesetzten Techniken Hand in Hand gehen. Das Projekt setzt auf dem konstruktiven Ansatz der modellgetriebenen Softwareentwicklung auf mit dem Ziel, die in den Modellen enthaltenen Informationen dahingehend auszuwerten, Testfälle für den Test der Software automatisch zu generieren. Beide Aspekte werden im Folgenden getrennt betrachtet.

3.3.2 Modellgetriebene Softwareentwicklung

Zu den etablierten konstruktiven Techniken gehört der Einsatz von Modellierungstechniken wie z.B. der Unified Modeling Language (UML) zur Spezifikation von Funktion, Struktur und Verhalten der Systeme. Einen ganz neuen Weg beschreitet dabei die modellgetriebene Softwareentwicklung (model driven architecture, MDA; model driven software development, MDSD). Einfach ausgedrückt wird dabei ein Modell fachlich so präzisiert, bis dass das Softwaresystem im Rahmen einer standardisierten Architektur generiert werden kann – ähnlich der computergestützten Fertigung, bei der ein dreidimensionales Modell eines Werkstücks in einem CAD-System erstellt, zusammen mit den notwendigen Bearbeitungshinweisen in ein Programm für die CNC-Werkzeugmaschine übersetzt, von dieser auf einem Rohling ausgeführt und so als reales Werkstück erstellt wird. Detaillierter betrachtet geht man in mehreren Schritten vor. Zunächst erstellt man ein beschreibendes, von jeglichen »IT-Aspekten« unabhängiges Modell des Anwendungsbereiches (computation independent model, CIM). Daraus leitet man manuell ein rein fachliches, technologieneutrales Modell für das Anwendungssystem ab (platform independent model, PIM), aus dem dann ein plattformspezifisches Modell (platform specific model, PSM) für eine bestimmte Architektur bzw. Realisierungstechnologie generiert wird. Aus diesem wird – ggf. nach weiteren Zwischenschritten – das Implementierungsmodell (implementation specific model, ISM) bzw. direkt der Quellcode des Anwendungssystems erzeugt. Den Zusammenhang verdeutlicht Abbildung 1.

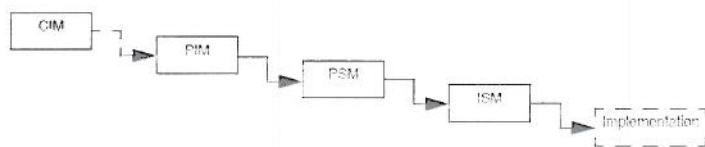


Abb. 1: Modelltransformationen der MDA

3.3.3 Modellbasierter Software-test

Bei der automatischen Generierung von Testfällen aus den während der Entwicklung erstellten Model-

len ist zu beachten, dass die Modellierungsinformation in der Regel nicht ausreichend ist, vollständige Tests zu generieren, da ansonsten ja auch die komplette Software generiert und – einen fehlerfreien Ge-

nerator vorausgesetzt – nicht mehr getestet werden müsste. In der Regel ist somit eine manuelle Nachbereitung der generierten Testfälle notwendig. Das Prinzip zeigt Abbildung 2.

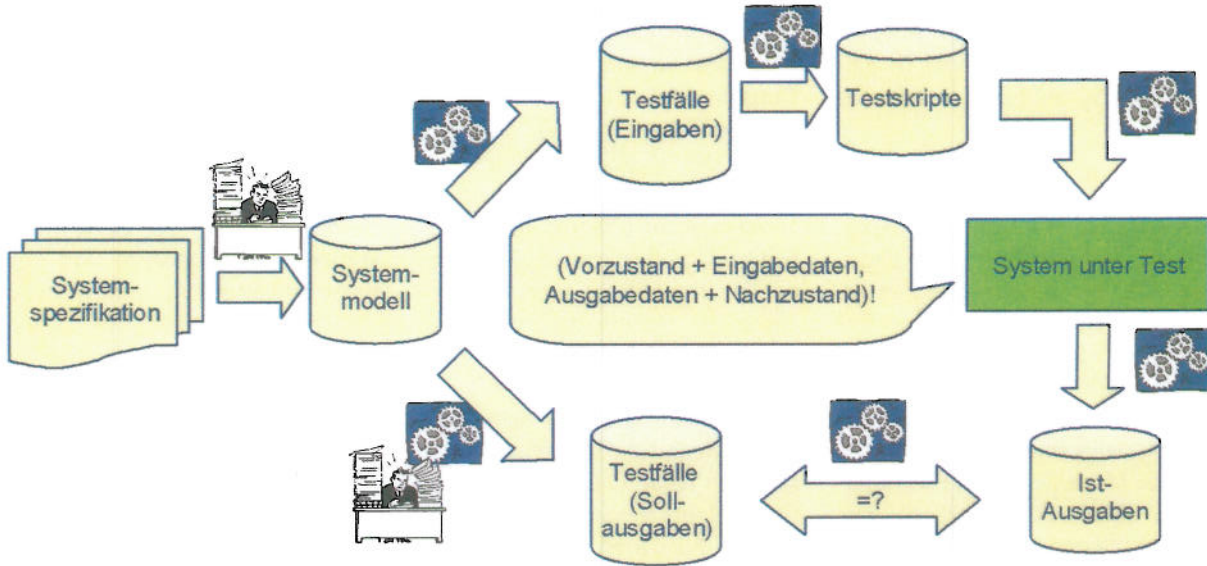
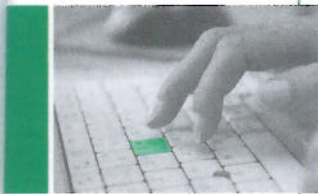


Abb. 2: Prinzip modellbasierter Software-Tests (MBT)



Produkte & Lösungen



Akademie



Beratung



Softwaretest Services

Entwicklung braucht Wissen

Die Zunahme modellbasiert entwickelter, komplexer Softwaresysteme bedeutet für den Softwaretest künftig eine Ergänzung der klassischen Testmethoden durch modellbasiertes Testen. Bei der Entwicklung von Testwerkzeugen muss jetzt über die Möglichkeit einer direkten Unterstützung von Modelltransformationen oder zumindest eine derartige Schnittstelle nachgedacht werden.

Die imbus AG hat aus diesem Grund das Erstellen einer Diplomarbeit zum Thema bei Prof. Dr. Mario Winter unterstützt. Die hierin gewonnenen Erkenntnisse werden bei imbus zu einem Forschungsprojekt weitergeführt werden.

Entwicklung braucht Wissen. Der Diplomand ist mittlerweile Mitarbeiter der imbus Rheinland GmbH. Und imbus ist weiter auf der Suche nach hervorragend ausgebildetem und hochqualifiziertem Personal.

Aktuell sucht imbus z.B. in Köln Testmanager (m/w):

- ❑ mit solider Kenntnis der gängigen Testmanagement- und Testautomatisierungs-Werkzeuge zur Software-Qualitätssicherung
- ❑ gute Kenntnis der etablierten (Test-)Prozessmodelle sowie der einschlägigen Normen und Standards
- ❑ Erfahrung im operativen Testen (konzipieren, planen, steuern, kontrollieren)
- ❑ erste Erfahrung im Management von Testprojekten auf Leitungsebene
- ❑ idealerweise ISTQB® Certified Tester Advanced Level, Test Manager

Die vollständige Stellenausschreibung für die Position des Testmanagers sowie weitere aktuelle Stellenangebote der imbus AG finden Sie im Internet unter www.imbus.de. Dort finden Sie auch ein Kontaktformular bei Fragen zur Diplomarbeit.

www.imbus.de

