

Testmanagement meets Model-Based Testing – modellzentrierter Test unterstützt Testmanagement

Gerhard Baier, AFRA GmbH

Mirco Richter, AFRA GmbH

Einleitung

Die Softwareentwicklung spielt inzwischen eine wichtige Rolle in vielen verschiedenen Industriezweigen – angefangen bei der Automobilindustrie über die Medizintechnik bis hin zur Automatisierungstechnik und zu vielen anderen Sparten. Die wachsende Komplexität der Systeme und folglich die ständig steigende Anzahl von Anforderungen an die Software führen zu höheren Ansprüchen an den Entwicklungs- und den Testprozess.

Um eine durchgängige Kontrolle über alle Entwicklungsphasen hinweg zu erhalten sind ein integrierter Entwicklungsprozess und der Einsatz einer dazu passenden Werkzeugkette unerlässlich. Dafür kommt dem Anforderungsmanagement eine herausfordernde Aufgabe zu. Dieses sollte sich möglichst nahtlos in die Spezifikations-, Entwicklungs-, Test- und Wartungsphase integrieren, um die Verfolgbarkeit der Anforderungen zu garantieren.

Die modellzentrierte Entwicklung und der modellzentrierte Test sind eine konsequente Verbesserungen der klassischen Softwareentwicklung und ein adäquater Weg, die obige Herausforderung anzunehmen. Insbesondere ist es möglich die Modellierung so in die Prozesse zu integrieren, dass diese deutlich von der grafischen Darstellung in der Unified Modeling Language (UML) profitieren können.

Wir konzentrieren uns in diesem Artikel auf die modellzentrierte Testfallgenerierung mit Anforderungsverfolgung und Unterstützung des Testmanagements (TM).

mzT unterstützt TM

Die modellbasierte Entwicklung ist in der Industrie teilweise etabliert und hat ihre Stärken bereits zeigen können. Aktuell zieht der Ansatz des modellzentrierten Tests (mzT) ein immer stärkeres Interesse auf sich. Wie kann nun mzT geeignet in bestehende Prozesse und die Werkzeugkette integriert werden?

Zentraler Angelpunkt ist die Beschreibung des Systems durch ein Modell, welches die Sicht des Testers widerspiegelt. Da sich der Testentwickler auf die Modellierung des Systems konzentriert und nicht mehr auf die schriftliche Definition einzelner Testfälle, wird ein deutlicher Gewinn an Übersicht über das zu testende System erzielt. Diese kann beliebig verbessert werden, in dem weitere Mittel zur Strukturierung wie z. B. Pakete und Klassen, Wiederverwendbarkeit oder die Bildung von Hierarchien verwendet werden. Auf dieser Basis wird die Kommunikation zwischen Entwicklern und Testern deutlich vereinfacht und zielgerichteter. Besonders gilt dies, wenn der Entwicklungsprozess ebenfalls modellbasiert durchgeführt wird.

Die Anforderungen werden im Modell mit den entsprechenden Elementen verknüpft, so dass diese anschließend in die Testgenerierung und -ausführung übernommen werden können. Die Verfügbarkeit der Anforderungen im UML Modellierungstool kann durch eine Importschnittstelle zum Anforderungsmanagement unterstützt werden. Beim Export der Testfälle in das Testmanagementwerkzeug werden die Anforderungsschlüssel mit übertragen, damit die Zuordnung der Testfälle und der einzel-

nen Testschritte zu den Anforderungen gesichert ist. Ein Tracing durch verschiedene Anforderungsebenen und der dazugehörigen Testfälle ist möglich.

Die Generierung der Testfälle und der zugehörigen Testdaten erfolgt nach Vorgabe der Überdeckungskriterien automatisch durch die Anwendung evolutionärer Algorithmen. Hierbei kann das Ergebnis der Generierung durch eine vorherige Auswahl von Anforderungen kontrolliert werden. D.h. es werden die Testfälle gezielt zur Abdeckung bestimmter Anforderungen generiert. Der Export der Testfälle in ein Testmanagement- und Testdurchführungswerkzeug erfolgt automatisch. Ebenso können die Testfälle alternativ oder zusätzlich als Dokument inklusive der Anforderungen ausgegeben werden.

Die geschilderte Vorgehensweise von der Modellierung bis zur Testdurchführung werden wir anhand eines Projektes mit Enterprise Architect und HP QualityCenter zeigen.

Bei Modelländerungen wird die Information über die Art und Position der Änderung ebenso weitergegeben. Dazu werden nur Testfälle geändert oder neue erzeugt, wenn dies aufgrund einer Änderung im Modell erforderlich ist. Somit steht dem Testmanager die detaillierte Information zur Verfügung, welche Anforderung an welchem Testfall getestet wird und welche Testfälle aufgrund von Modelländerungen geändert oder neu erstellt wurden. Andere Testfälle sind als identisch oder nicht mehr gültig markiert.

Diese Information bildet dann eine komfortable Basis für die Testplanung und Testdurchführung.

Die Ergebnisse der Testdurchführung landen im Testmanagementwerkzeug. Eine Visualisierung der gefundenen Fehler im Modell kann bei der Analyse der Fehler hel-

fen. Dazu werden die Informationen aus dem TM-Werkzeug exportiert und im Modell für die Visualisierung zur Verfügung gestellt. Eine grafische Darstellung des betroffenen Testfalls, der Fehlerstelle und der betroffenen Anforderungen ist dadurch möglich.

Diese Information ist nun universell nutzbar. Der Testentwickler kann beurteilen, ob ein Fehler im Modell vorliegt, die Entwicklung analysiert eventuelle Fehler in der Implementierung und das Anforderungsmanagement hat die Möglichkeit Unzulänglichkeiten in der Anforderungsanalyse aufzudecken.

Komplette Prozessunterstützung mit mZT

Damit ergibt sich die Möglichkeit eine komplette Prozessunterstützung durch mZT aufzubauen. Eine visuell unterstützte und dokumentierte Verfolgung der Anforderungen bis in die Testergebnisse verspricht erhebliche Vorteile in der Praxis.

Ein Tailoring für die vorhandenen Prozesse und Werkzeuge bindet mZT optimal in bestehende Prozesse ein. Verschiedenste Ausprägungen sind denkbar. Werden Anforderungen ins Modell gezogen? Welche Ebene wird behandelt, Unit- bis Systemtest? Welche Dokumentation wird automatisch erstellt etc.

Ein schrittweise Einführung von mZT vereinfacht den Einstieg und zeigt frühzeitig die Vorteile, die erreicht werden. Die automatische Generierung von Testfällen kann an ausgewählten Komponenten oder Anforderungen erprobt werden und sich damit in der Praxis beweisen.