

TestBench meets TestFrame™: State of the Art Testdesign

Dierk Engelhardt, Tilo Linz
imbus AG

dierk.engelhardt@imbus.de
www.testbench.info

Generischer, iterativer Prozess

Der Softwaretestprozess wird in der Regel als iterativer Prozess beschrieben, der aus den folgenden Phasen besteht:

- Testplanung,
- Testdesign,
- Testautomatisierung und
- Testdurchführung.

Diesem Standard-Testprozess liegen Rollen- und Rechtedefinitionen, Use Cases und erzeugbare Dokumente zu Grunde, wie in Abbildung 1 dargestellt ist.

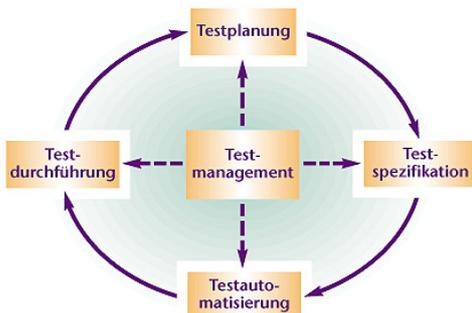


Abbildung 1: Der Testprozess

Innerhalb dieses Prozesses kommt der Testspezifikation eine besondere Bedeutung zu, da sie starke Auswirkungen auf die Steuerung und Bewertung der folgenden Phasen hat. Gerade auch die Testautomatisierung ist sehr von Art und Umfang der Testspezifikation abhängig.

Testspezifikation

Bei der Testspezifikation (auch als Testdesign bezeichnet) wird festgehalten, wie zu verifizieren und zu validieren ist. Ein formales, modulares Design von Testfällen ist der wesentliche Erfolgsfaktor für die Wirksamkeit von systematischen Softwaretests. Die imbus TestBench z.B.

nutzt dabei die sogenannte Interaktionsmethode. Die Interaktionsmethode basiert auf wiederverwendbaren Ablaufsequenzen - eben den Interaktionen - und abstrakten Datentypen. Letztere werden in Äquivalenzklassen zerlegt, wobei jeder Äquivalenzklasse ein oder mehrere Repräsentanten zugeordnet sein können. Die den Testfällen zugeordneten Testdaten werden dadurch formal definiert abstrahiert. Die Testfälle werden in beliebig vielen Hierarchiestufen in Interaktionsschritten gegliedert. Eine Interaktion ist hierbei je nach Abstraktionsebene ein fachlicher oder auch ein technischer Testschritt, darunter versteht man im Allgemeinen das Anreizen des zu testenden Systems zum Beispiel durch Eintragen von Daten, Senden einer Nachricht oder ähnliches, oder die Prüfung einer erwarteten Sollreaktion des Systems, zum Beispiel ein erzeugtes Datum oder eine empfangene Nachricht. Interaktionen sind parametrierbar, sie können bei jeder Verwendung unterschiedliche Daten senden, empfangen und verarbeiten.

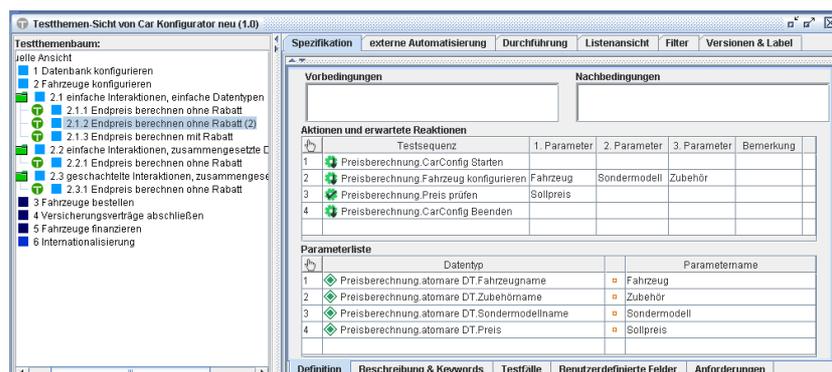


Abbildung 2: Mittels der Interaktionsmethode werden Testabläufe aus einzelnen, wieder verwendbaren Testelementen zusammengesetzt.

Durch Abstraktion und Kapselung von Interaktionen und Datentypen wird eine robuste Test-Architektur gewährleistet, die stabil gegenüber Veränderungen des

Testobjektes, von Testfällen oder Testdaten ist.

Diese Abstraktion ermöglicht eine Spezifikation auf der fachlich, funktionalen Ebene. Testdesigner können sehr frühzeitig mit der Testspezifikation beginnen, unabhängig von Testautomatisierung und Testdurchführung. Dabei bilden die Interaktionen das fachliche Vokabular des Testers zur Beschreibung der Testfälle. Dieses Vokabular ist für ein System in der Regel endlich, ebenso wie die notwendigen Datentypen. Ist das Vokabular einmal vollständig, können Testspezifikationen sehr schnell erstellt werden. Die Wartbarkeit erhöht sich enorm durch die Verwaltung des Vokabulars in einem zentralen Repository, so dass eine Änderung am Vokabular an einer einzigen Stelle stattfindet und alle Referenzen darauf automatisch aktualisiert werden.

Testdurchführung

Da es die Hierarchisierung der Interaktionen ermöglicht in Interaktionen wieder Interaktionen aufzurufen und das beliebig tief, können fachliche Interaktionen aus technischen Interaktionen zusammengesetzt werden. Technische Interaktionen bilden somit den an der Schnittstelle zum Testobjekt notwendigen Ablauf des Tests auf Ebene von z.B. Fenstern und Controls bei GUI-Applikationen, ab.

Da in der imbus TestBench die Beschreibung, also die Spezifikation der Tests erfolgt, ist ein System vonnöten, das die manuelle und automatisierte Durchführung der Tests ermöglicht. Für die manuelle Durchführung stellt die TestBench selbst einen Durchführungsassistenten zur Verfügung, der sowohl im Online- als auch im Offline-Betrieb (d.h. beispielsweise im Offshore-Testdurchführungslabor, bei Tests in Fahrzeugen etc) dem manuellen Tester eine komfortable Umgebung zur Darstellung der Testschritte, zur Protokollierung der Ergebnisse und zur Erfassung von Soll/Ist-Abweichungen liefert. Zur automatisierten Durchführung wurde der TestFrame™ von LogicaCMG angebunden. Dieser bringt für die meisten grafischen Oberflächen fertig

automatisierte Interaktionen auf Ebene der GUI-Controls mit, so dass aus diesen Controls die fachliche Logik aggregiert werden kann. Damit minimiert sich der Aufwand für die Erstellung der Testautomatisierung nochmals erheblich.

Schnittstellen

Die imbus TestBench bietet eine generische Schnittstelle zur Ankopplung von Werkzeugen für Anforderungsmanagement und Fehlermanagement. Innerhalb der TestBench werden die Anforderungsdaten verschiedenster Werkzeuge gleichartig dargestellt. Die Abbildung zwischen dem konkreten Managementwerkzeug und dem allgemeinen Datenmodell der imbus TestBench leistet ein toolspezifischer Wrapper, der Bestandteil der TestBench-Serverarchitektur ist. Damit können die wichtigsten Inhalte umgesetzt werden:

- Anforderungsbasierte Testplanung
- Fehlerbasierte Testplanung und –durchführung
- Anforderungsbasierter Testfallentwurf
- Anforderungsbasiertes Reporting
- Management von Änderungen

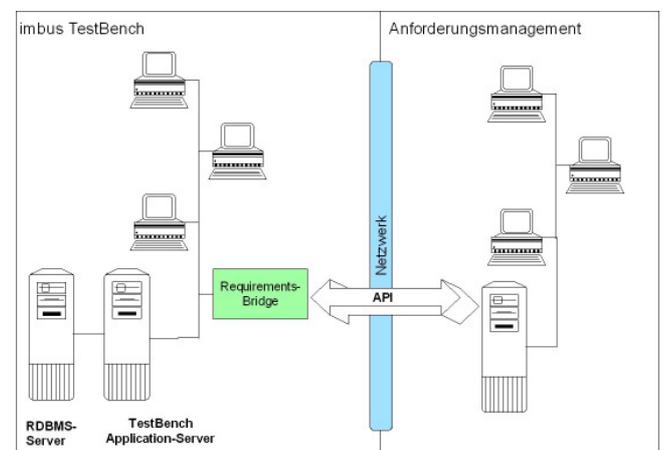


Abbildung 3: Anbindung von Anforderungsmanagementsystemen

Fazit

Durch die oben beschriebenen Methoden konnten bei Anwendern 30 % mehr Testfälle bei gleichem Aufwand bearbeitet werden, was zu 45% weniger Fehlern im Endprodukt führte. Ebenso konnten 30 % Aufwandsersparnis bei Testmanagement und Testauswertung gemessen werden.