

# Nichtfunktionaler Abnahmetest: Planung, Durchführung und Automatisierung

Beitrag für den 22. TAV-Workshop der GI-FG 2.1.7 „Test, Analyse und Verifikation von Software“  
17./18.2.2005, Hochschule Bremen

N. Fleischmann, U. Hehn, method park Software AG (Erlangen)  
Kontakt: Uwe.Hehn@methodpark.de

**Zusammenfassung:** Abnahmetests sind ein wesentlicher Bestandteil des Abnahmeverfahrens für Softwareprodukte durch den Kunden. Im Folgenden wird eine Vorgehensweise für den Abnahmetest nichtfunktionaler Anforderungen vorgestellt. Dabei fokussieren wir uns auf die Aspekte Planung, Durchführung und Automatisierung aus dem Blickwinkel eines Testdienstleisters.

## 1. Abnahmetests im Software-Lebenszyklus

*Die folgende Beschreibung des Themas Abnahmetests basiert auf dem Lehrplan für den ISTQB Certified Tester Foundation Level [ISTQB]:*

Der Abnahmetest folgt dem Systemtest vor der Inbetriebnahme bzw. der Auslieferung der Software.

Beim Abnahmetest stehen die Sicht und das Urteil des Kunden bzw. des Anwenders der Software im Vordergrund. Häufig sind Kunde oder Anwender auch direkt daran beteiligt.

Zu prüfende Aspekte sind insbesondere die vertragliche Akzeptanz bzw. die Benutzerakzeptanz des Softwaresystems. Eine typische Art der Durchführung des Abnahmetests ist der so genannte Feldtest. Hier werden die Tests in der bestimmungsgemäßen Umgebung durchgeführt. Der Softwarehersteller liefert dazu stabile Vorabversionen der Software an einen ausgewählten Kundenkreis. Diese ausgewählten Kunden führen die Tests entsprechend der vom Hersteller vorgegebenen Testszenarien durch, oder sie setzen das Produkt probenhalber unter realistischen Bedingungen ein. Anschließend geben sie ihre Fehlermeldungen bzw. auch weitere Ein-

drücke oder Kommentare an den Hersteller zurück.

## 2. Abnahmetest: Ja oder Nein?

Entsprechend der obigen Beschreibung wird der Abnahmetest verstanden als Voraussetzung für die Abnahme des Softwareprodukts durch den Kunden. Der Abnahmetest ist u.a. wichtig, weil damit typischerweise die Vertragsabnahme und in der Folge die Rechnungsstellung für ein abgeschlossenes Entwicklungs- und/oder Systemintegrationsprojekt verbunden ist.

Das Ergebnis eines erfolgreichen Abnahmetests, ist der Nachweis der praktischen Einsetzbarkeit des Systems aus Sicht des Endnutzers bzw. des Kunden bzw. der Erfüllung wesentlicher Vertragsbestandteile.

Tatsächlich wird der Abnahmetest in der Praxis nur in einem Teil der Entwicklungsprojekte durchgeführt. Insbesondere ist das Aufstellen einer eigenen Abnahmetestspezifikation wohl eher die Ausnahme als die Regel. Damit verzichtet man jedoch auf eine wichtige risikoreduzierende Maßnahme: Durch genaues Festlegen eines Abnahmeprozesses verbunden mit einem definierten Abnahmetest kann eine zu Unrecht verweigerte Abnahme weitgehend ausgeschlossen werden.

## 3. Die Basis für den Abnahmetest

Entsprechend dem allgemeinen V-Modell wird der Abnahmetest auf Basis der Abnahmetestspezifikation durchgeführt, die wiederum auf dem Lastenheft des Kunden bzw. dem daraus abgeleiteten und mit dem Kunden abgestimmten Pflichtenheft basiert.

Im Folgenden werden diesbezügliche Unterschiede zwischen Abnahmetests für funktionale und nichtfunktionale Anforderungen betrachtet.

### 3.1. Funktionaler Abnahmetest

Ein Abnahmetest der Funktionalität eines Softwaresystems setzt prinzipiell ein vollständiges und konsistentes Pflichtenheft bzw. Lastenheft voraus.

Fehlen diese grundlegenden Informationen bzw. sind sie nur unvollständig, ist die Überprüfung der Funktionalität eines Softwaresystems aufgrund der nur partiell bekannten Funktionalität im Allgemeinen sicherlich nicht adäquat möglich. Allenfalls kann der fachliche Abnahmetest von erfahrenen Fachleuten der jeweiligen Disziplinen aufgrund ihrer Kenntnisse der Anwendungsdomain durchgeführt werden.

### 3.2. Nichtfunktionaler Abnahmetest

Eine Reihe nichtfunktionaler Anforderungen hingegen, wie Robustheit, Performanz, Installierbarkeit, Konfigurierbarkeit, Skalierbarkeit, können jedoch mit gutem Erfolg ohne detailliertes Wissen der grundlegenden Funktionsweise des Systems überprüft werden. Eine explorative, nicht spezifikationsbasierte Vorgehensweise [Bach] ist hier prinzipiell zulässig und möglich.

Aus dem Gesichtspunkt der Validierung ist es sogar sinnvoll, solche nichtfunktionale Aspekte von einem unabhängigem Testteam durchführen zu lassen, weil die Wahrscheinlichkeit höher ist, dass dadurch neue Fehler erkannt werden. Im Pflichtenheft nicht detaillierte nichtfunktionale Anforderungen können aus Kundensicht bewertet werden.

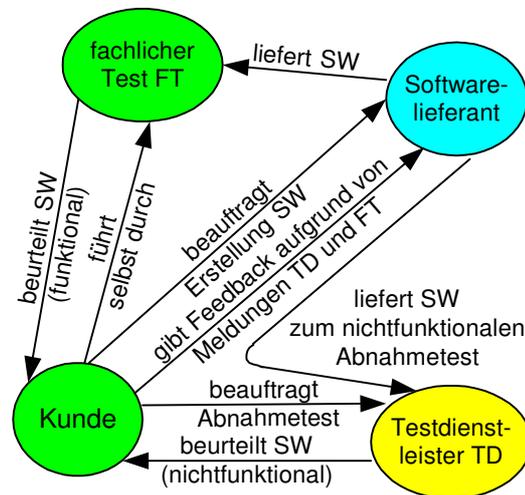
Ein Abnahmetest für diese Aspekte, die ja für die Benutzbarkeit des Produkts wesentlich sind, sollte daher grundsätzlich durchgeführt werden. Der Teil des Abnahmetests, der sich auf die Prüfung der nichtfunktionalen Anforderungen bezieht, wird im Folgenden als *Nichtfunktionaler Abnahmetest* bezeichnet.

## 4. Planung des Nichtfunktionalen Abnahmetests

### 4.1. Nichtfunktionale Abnahmetests als Dienstleistung

Im Folgenden betrachten wir die Vorgehensweise für einen Nichtfunktionalen Abnahmetest auf

der Basis nur partieller Kenntnis der Funktionalität aus der Sicht eines Testdienstleisters. Der Kunde lässt Software von einem Softwarelieferanten erstellen, führt die fachliche Abnahme selber durch, und beauftragt einen Testdienstleister mit dem Nichtfunktionalen Abnahmetest.



### 4.2. Aufwandsschätzung

Zunächst muss der Umfang des Nichtfunktionalen Abnahmetests geklärt werden, insbesondere:

- Welche Teilsysteme sind zu prüfen?
- Welche nichtfunktionalen Aspekte sind zu betrachten (Robustheit, Performanz, Installierbarkeit, Konfigurierbarkeit, Skalierbarkeit, ...)?
- Welche möglichen Konfigurationen des Systems sind im Test zu betrachten?

Zweiter Schritt ist die Klärung der Verfügbarkeit des zu testenden Systems zur Ermittlung von geeigneten Testfällen, unter anderem:

- Wann steht eine erste, bereits weitgehend stabile Version des Systems zur Verfügung?
- Wie kann ein grundlegender Knowhow-Transfer des für die Exploration des Systems minimal erforderlichen Fachwissens erfolgen?
- Wann werden die Releaseversion des Softwaresystems sowie ggfs. weitere Zwischenversionen vorliegen?

Sind diese Information verfügbar, kann eine erste Festlegung von Abnahmetestfällen erfolgen, die als Basis für die Aufwandsschätzung des Abnahmetests dienen.

### 4.3. Iterative Vorgehensweise

Sinnvollerweise wird der Abnahmetest in mehrere Phasen aufgeteilt, entsprechend der vorgesehenen weiteren Zwischenversionen bzw. der Releaseversion der Software. Dabei werden die Aktivitäten *Erstellen bzw. Überarbeiten von Testfällen* und die *Ausführung von Tests* parallel, jedoch um eine Phase versetzt, durchgeführt.

Wenn  $TF_x$  das Erstellen von Testfällen in der Phase  $x$  bezeichnet,  $DT_x$  das Durchführen der in der Phase  $x$  aufgestellten Testfälle, sowie  $(DT_x(TF_{x-1}) \parallel TF_x)$  die parallele Durchführung folgender Aktivitäten in Phase  $x$

- Durchführung der Testfälle der  $(x-1)$ . Phase, und
- Ergänzen der vorhandenen Testfälle  $TF_{x-1}$  (Ergebnis:  $TF_x$ ) für die Durchführung von Tests in der nächsten Phase,

ergibt sich hieraus unmittelbar die folgende parallele und versetzte Vorgehensweise ( $n = \text{Zahl der Testphasen} = \text{Zahl der bis zur Releaseversion zu testenden Softwareversionen}$ ):

$$\begin{aligned}
 TF_1 &\rightarrow (DT_2(TF_1) \parallel TF_2) \\
 &\rightarrow \dots \\
 &\rightarrow (DT_{n-1}(TF_{n-2}) \parallel TF_{n-1}) \\
 &\rightarrow DT_n(TF_{n-1})
 \end{aligned}$$

Der eigentliche Abnahmetest findet in der letzten - der  $n$ -ten - Phase statt. Alle Phasen außer der letzten dienen der Vorbereitung der bestmöglichen Durchführung. Die erste Phase dient darüber hinaus zur Abschätzung der Zahl der Testfälle und des zu erwartenden Aufwandes für die Durchführung der Testfälle.

Anmerkungen:

- Je nach Länge einer Testphase  $x$  und der Zahl der Testzyklen innerhalb dieser Phase kann der Test statt auf  $TF_{x-1}$  ggfs. auch schon auf der überarbeiteten Version  $TF_x$  der Testfälle aufgesetzt werden.
- Eine Forderung an den Kunden ist, dass sich der Unterschied von  $TF_{x-1}$  zu  $TF_x$

aus der Releasenote der Softwareversion  $x$  ergeben muss.

- Der Aufwand für  $TF_x$  (Testdesign) nimmt mit zunehmendem  $x$  ab, wogegen der Aufwand für  $DT_x$  (Testdurchführung) mit zunehmendem  $x$  zunimmt.

### 4.4. Review der Testfälle durch den Kunden

Nach Abschluss jeder Phase  $x$  kann und sollte der Kunde die aktuelle Version der Testfallspezifikation  $TF_x$  überprüfen und freigeben.

## 5. Durchführung des Abnahmetests

Die Durchführung des Abnahmetests setzt die Verfügbarkeit einer angemessenen Testumgebung (wie verfügbarer Testrahmen, geeignete und angemessen viele Testdaten, ggfs. Testautomatisierungswerkzeuge) und der notwendigen Softwarestände voraus.

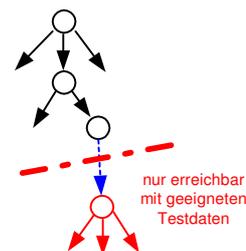
### 5.1. Notwendige Voraussetzungen

Um die geplanten Testphasen und die damit verbundenen Meilensteine einhalten zu können, ist es notwendig, dass diese Voraussetzungen zu Beginn jeder Phase ohne wesentliche Verzögerungen erfüllt werden.

Insbesondere die letzte Testphase, in der die Releaseversion getestet wird, sollte plangemäß begonnen und beendet werden, da die Abnahme des Softwareprodukts entscheidend vom Ergebnis dieser Überprüfung abhängen wird.

### 5.2. Testdaten

Zur Erreichung einer genügenden „Eindringtiefe“ in den Testling muss der Kunde geeigneter Testdaten bereitstellen. Dadurch wird verhindert, dass dem Testdesigner bzw. dem Durchführer des Tests bestimmte Teile der Anwendung nicht zugänglich sind.



## 6. Automatisierung

Es muss kritisch geprüft werden, ob eine Automatisierung der Abnahmetests sinnvoll und vom Aufwand her angemessen ist. Für eine Automatisierung eines Teils der Nichtfunktionalen Anforderungen spricht Folgendes:

Die Nutzbarkeit eines Softwaresystems in der realen Umgebung mit u.U. vielen Nutzern ist oft ein sehr wichtiger Aspekt. Im Systemtest eines Softwaresystems kann die reale Umgebung oft nicht genügend gut simuliert werden. Für die Untersuchung der Performanz (Lasttest) kann eine Automatisierung sinnvoll sein, um eine genügend hohe Last zu erzeugen.

Hierzu werden je nach Einsatzumgebung spezifische Lastgeneratoren verwendet. Beispielsweise im Umfeld von Datenbank-basierenden Applikationen gibt es hierfür eine Reihe brauchbarer kommerzieller Werkzeuge.

## 7. Ein Fallbeispiel

Anhand eines konkreten Projekts sollen einige der angesprochenen Aspekte konkretisiert werden:

Das betrachtete System ist ein Client-Server-System, basierend auf aktueller Windows-Technologie. Prinzipiell kann eine beliebig hohe Zahl von Clients auf den Server zugreifen.

Die fachlichen (funktionalen) Abnahmetests werden vom Kunden durchgeführt. Ziel des ausgelagerten Nichtfunktionalen Abnahmetests ist es, die Stabilität des Systems, die Robustheit bzgl. Benutzereingaben und eine ausreichende Antwortzeit auch bei großer Zahl von Anwendern und bei starker Nutzung des Systems nachzuweisen. Das Ergebnis des Abnahmetests ist Basis für die eigentliche Abnahme des Systems.

Die zu testende Software wird in mehreren Versionen geliefert, ist jedoch sowohl vom Funktionsumfang als auch von der Oberfläche bereits weitgehend stabil. Die oben angegebenen Voraussetzungen für die Durchführung eines Feldtests sind damit erfüllt.

### 7.1. Planung

Die Planung wird entsprechend der oben beschriebenen Vorgehensweise durchgeführt. Ergebnis der ersten Phase sind a) eine Liste der durchzuführenden Testfälle und b) darauf basierend eine Abschätzung des zu erwartende Ge-

samtaufwandes für die Durchführung der Testfälle. Hinzu kommt der geschätzte Aufwand für den Aufbau einer geeigneten Testumgebung sowie einer Automatisierung der Performanztests (Erzeugen von realistischer Last) inklusive der benötigten Ressourcen HW, SW, Testautomatisierungs-SW).

### 7.2. Automatisierung des Performanztests

Für die Automatisierung der Erzeugung von Last ist im gegebenen Kontext kein kommerzieller Lastgenerator erhältlich. Daher wurde der Aufbau einer Testumgebung mit n „Lasterzeugenden Clients“ eingeplant (mit realistisch großer Zahl n von Last-Clients).

Je Last-Client wird mit einem geeigneten Werkzeug ein lasterzeugendes Bedienszenario simuliert, so dass prinzipiell eine beliebige Last des Servers und des Kommunikationsmediums möglich ist.

Da keine darüber hinaus gehende Automatisierung angestrebt wird, war zu klären, ob eine Automatisierung mit elementaren Windows-Mitteln ausreichend ist, oder ob der Einsatz eines kommerziellen Werkzeugs erforderlich ist. Als Ergebnis der Evaluation mehrerer Ansätze entschied man sich für ein leistungsfähiges kommerzielles Werkzeug, das auch eine Weiterpflege der Tests für die folgenden Iterationen der Testphasen erlaubt.

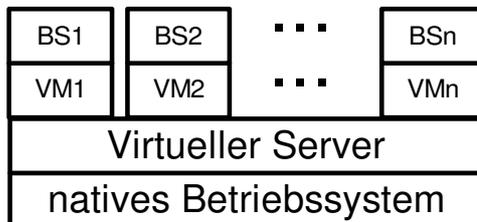
### 7.3. Durchführung des Performanztests

Aufgrund der großen Zahl von notwendigen Last-Clients wird der Lasttest – im Gegensatz zu den weiteren Tests – in der Umgebung des Kunden durchgeführt. Der Kunde ist für den Aufbau der entsprechenden Testumgebung zuständig; der Testdienstleister ergänzt den Aufbau bzgl. der Automatisierung der Last-Clients und der manuellen Durchführung der Performanzmessungen.

### 7.4. Testumgebungen und deren Verwaltung

Da sowohl für den Test beim Testdienstleister als auch beim Test in der Umgebung des Kunden eine reproduzierbare Testumgebung notwendig ist (wesentliche Parameter: Betriebssystemvariante, Testdatenbank in verschiedenen Ausprägungen), wurde der Einsatz der Virtual-

PC- bzw. Virtual-Server-Technologie von Microsoft eingesetzt. Ein Virtual-Server erlaubt es, mehrere virtuelle Maschinen auf einem realen Rechner zu instanzieren, auf denen unabhängig voneinander beliebige Betriebssystem- und Anwendungskonfigurationen installiert werden können.



Damit ist es möglich

- Einerseits beliebige Systemkonfigurationen aufzubauen (betreffend der o.a. und weiterer Parameter), und
- andererseits die Testdaten für eine Testdurchführung sehr einfach wiederherstellen zu können,
- und das unabhängig für alle Tester, die in dem Projekt mitarbeiten,

da der komplette Zustand einer virtuellen Maschine in einer Datei exportiert und von dort wieder importiert werden kann.

## 8. Ausblick

Das hier beschriebene Vorgehen für den *Nicht-funktionalen Abnahmetest* hat sich in realen Projekten bewährt. Vorteilhaft ist, dass sich eine zuverlässige Aufwandsschätzung durch die versetzten parallelen Aktivitäten in den aufeinander folgenden Phasen direkt ableiten lässt.

Bezüglich der Automatisierung von Abnahmetests, die bisher wenig betrachtet wurde, erscheinen weitere Verbesserungen möglich zu sein.

## 9. Literatur

[ISTQB]

Grundlagen des Softwaretestens,  
Lehrplan zum Basiskurs,  
ISTQB Certified Tester, Foundation Level,  
Version 2.2., Juli 2003,  
<http://www.isqi.org/isqi/documents/>

[Bach] James Bach:

Scripted Versus Exploratory Testing,  
Proc. EuroSTAR 2003, Amsterdam Dec 2003