

e-business-Testing

Eine neue Herausforderung

Mit Genehmigung der Autoren, © IT Consulting IBM Switzerland

Überblick

- Das *Qualitäts Management (QM)* und insbesondere das Testen sind in der Entwicklung von e-business-Lösungen wichtige Erfolgsfaktoren, infolge akzentuierter, neuer Herausforderungen.
- Das *Risiko Management* ist die Grundlage für fokussiertes, zielgerichtetes Testen.
- Die *Erfahrungen* zeigen, daß die Aufmerksamkeit speziell der intuitiven Benutzbarkeit, dem Leistungsverhalten und der fehlerarmen Verarbeitung zu widmen ist.
- Das notwendige *methodische Testvorgehen* unterstützt den gesamten Lebenszyklus einer Lösung und erlaubt sowohl die traditionellen, phasenweisen Entwicklungsprozesse, als auch das evolutionäre Entwicklungsvorgehen. Ein mögliches Testvorgehen, das Full-Lifecycle-Testing (FLT), wird kurz vorgestellt.

Einleitung

e-business bietet neue Möglichkeiten, wie Geschäfte getätigt werden können und stellt dadurch eine große Herausforderung für das bestehende Geschäft dar. Nicht weniger herausgefordert sind aber die "Systembauer", die sowohl den neuen technischen und geschäftlichen Anforderungen gerecht werden müssen, als auch die Charakteristiken und Auswirkungen der zu entwickelnden Lösungen im Auge be-

halten sollten. Eine solche Auswirkung kann beispielsweise die interne und externe Visibilität einer Firmeneinheit sein, die beim Zusammenführen des Web und der internen Informationstechnologien erzeugt wird – und dies in einem Ausmaß und einer Schnelligkeit, wie sie vielleicht nie erwartet wurde. In diesem Artikel werden wir uns einigen dieser Herausforderung stellen, wobei wir uns auf die Qualitätssicherung und im engeren Sinn auf das Testen von e-business-Lösungen beschränken. Das folgende Schema zeigt die Gliederung dieses Artikels.

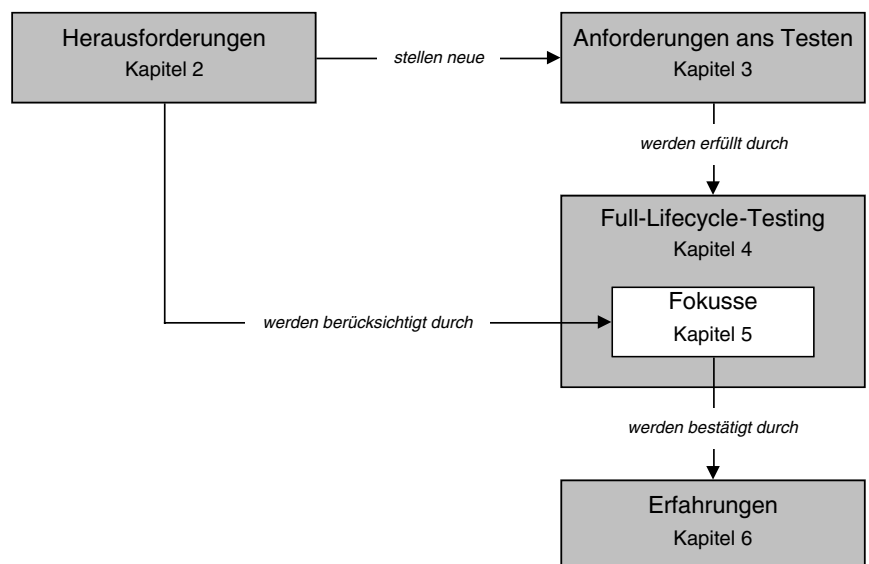


Abbildung 1: Gliederung dieses Artikels

e-business-Lösungen – neue Herausforderungen

Es läßt sich darüber diskutieren, ob die Möglichkeit des e-business die neuen Herausforderungen generiert oder ob

die Art der Umsetzung die eigentliche Ursache dafür ist. In jedem Fall werden wir bei der Erstellung von e-business-Lösungen aus folgenden Gründen herausgefordert:

- *Mehr Benutzer* – Im Gegensatz zu "internen" Systemen mit den Mitarbeitern als Benutzer, sind bei einer e-commerce-Lösung die Kunden und bei einem Extranet die Lieferanten neu zu den Benutzern zu zählen. Auch führt die weltweite Erreichbarkeit und das schnelle Wachstum, der am Internet angeschlossenen Personen und Firmen, potentiell zu mehr Benutzern. Die Anzahl Benutzer, die Zugriffsdauer etc. sind in diesem neuen Cybermarket viel schwieriger zu prognostizieren als in den etablierten Online-Systemen oder Systemen mit einer bekannten Benutzerzahl.
- *Unbekannte Benutzer* – Bei nach außen offenen Lösungen kennt man

das genaue Benutzerprofil selten (Web-Erfahrung, Hardware- und Softwareausstattung, etc.). Dies kann verschiedene Ursachen haben, seien dies Datenschutzgründe oder die oft fehlende Auswertung der Profile. Einflußmög-

lichkeiten bestehen in solchen Fällen kaum.

■ *Höhere Sicherheitsanforderungen* – Werden vorhandene Systeme mit dem Internet verknüpft, so stellt dies ein erhöhtes Sicherheitsrisiko dar. Hacker können bei mangelnden Sicherheitsvorkehrungen Zugang zu internen Daten und Systemen finden. Auch ist nur ein kleiner Teil der technischen Infrastruktur des Internet unter direkter Kontrolle einer Unternehmung. Eventualitäten, wie zum Beispiel der Ausfall eines Knotenrechners, müssen antizipiert werden.

■ *Höhere Qualitätsanforderungen* – e-business heißt, daß auftretende Probleme wie z. B. Systemausfälle und ein falscher Anwendungsablauf mehr beeinträchtigen, als “nur” die Produktivität der eigenen Mitarbeiter. Bei längerer Laufzeit (24 Stunden an 7 Tagen) ist also noch weniger Ausfallzeit erlaubt. Die Benutzer bewerten neben dem Inhalt vor allem die intuitive Benutzbarkeit, das Leistungsverhalten und die Vertrauenswürdigkeit einer Internet-Seite. Die – vielleicht bessere – Konkurrenz ist nur ein paar Mausklicks entfernt.

■ *Faster to market* – Der erhöhte Konkurrenzdruck und der schnelle Technologiewandel erfordern eine noch effizientere Systementwicklung und eine noch schnellere Einführung, bei mindestens gleichbleibender Qualität.

■ *Neue Technologien (Distributed OO, Java, Multi-Tier Architektur etc.)* – Neue Technologien erhöhen normalerweise die Komplexität und die gegenseitige Abhängigkeit von Anwendungen. Sie können aber auch zu kürzeren Entwicklungszeiten führen und schnellere Korrekturen ermöglichen. Diese inkrementelle, schrittweise Entwicklung erfordert aber mehr Pla-

nungs- und Koordinationsaufwand. Treibende Technologien wie Multimedia stellen zusätzlich hohe Anforderungen an die Netzinfrastruktur bezüglich höherer Bandbreite, zeitkritischen Abläufen, etc.

■ *Mangelnde Skills und Ressourcen* – In der ganzen IT-Branche sind kompetente Fachleute mit profunder Erfahrung schwer zu finden. Der Markt ist ausgetrocknet. Dies gilt ganz besonders für den Bereich der Qualitätssicherung. Die Rollen der Testspezialisten haben noch nicht die notwendige Attraktivität und Anerkennung. Hier besteht dringender Handlungsbedarf.

Diese zum Teil divergierenden Zielsetzungen – höhere Qualität und zugleich eine kürzere Entwicklungszeit, um der Dynamik der Märkte gerecht zu werden – setzen noch mehr Druck auf das Qualitätsmanagement, welches selbst bei den herkömmlichen Anwendungen noch immer nicht die verdiente Aufmerksamkeit erhält.

e-business-Testing – die wichtigsten Anforderungen

Wie muß ein strukturiertes Qualitätsmanagement aussehen, welches den Herausforderungen des vorhergehenden Kapitels gerecht werden soll? Beschränken wir die Fragestellung auf den Bereich Testen, so lassen sich einige wichtige Anforderungen an eine geeignete Testmethodik formulieren:

■ *Erweiterung des Testhorizonts* – e-business betont verschiedene Bereiche, die besonders getestet werden müssen. Es sind dies zum Beispiel die Sicherheit (Zugriffsrechte, Schutz vor Hackern, geschützte Übertragung etc.), die Benutzbarkeit, das Verhalten

unter sehr großer Last, die Skalierbarkeit und die Schnittstellen zu den traditionellen Host-Systemen. Keiner dieser Bereiche betrifft ausschließlich e-business-Lösungen, sie verdienen in diesem Umfeld aber besondere Beachtung.

■ *Testen mit Konzentration auf Risikobereiche* – Der Schere des fasterto-market und dem gleichzeitigen Anspruch nach höherer Qualität kann nur begegnet werden, wenn man sich bewußt ist, daß nicht alles getestet werden kann. Es ist angebrachter, sich auf das Wesentliche zu beschränken, dieses aber ausgedehnt und kompromißlos zu testen. Bei den ausgeklammerten Bereichen muß man das dadurch eingegangene Restrisiko kennen. Das Ziel muß eine Steigerung sowohl der Effizienz, als auch der Effektivität sein.

■ *Benutzer früh einbeziehen* – Da man den Benutzer und sein Verhalten nicht kennt und ihn vor der Benutzung von Internet-Applikationen meistens nicht schulen kann, müssen repräsentative Mengen von Benutzern früh in den Entwicklungsprozeß eingebunden werden. Ausgewählte Mitarbeiter (Intranet), Lieferanten (Extranet) und Konsumenten (Internet) sollten schon bei der Erfassung der Anforderungen involviert sein. Das Verhalten dieser Benutzer kann später in einem Usability-Labor¹ getestet werden. Zum Test der Systemsicherheit vor Angriffen können zum Beispiel “ethical hacker” herangezogen werden. Als besonders hilfreich haben sich nach Abschluß der Entwicklung interne Testläufe oder ein Pilotbetrieb mit eingeschränkter Anzahl von Benutzern erwiesen. Einige Entwickler von Spielen lassen ihre Spiele gleich über das Internet testen, indem sie an Wettbewerben oder vereinbarten Sonntagnachmittagen ihre

Beta-Releases zur allgemeinen Teilnahme zur Verfügung stellen. Die Tester – meist Kinder – prüfen dabei die Software auf Stabilität und Komfort.

■ *Alle Sichten berücksichtigen* – e-business hat mindestens so viel mit dem Geschäft, wie mit der Technologie zu tun. Deshalb sollte aus Sicht Technik, Benutzer und Management getestet werden.

■ *Sequentielles und iteratives Vorgehen unter einen Hut bringen* – e-business-Lösungen bestehen meist aus Erweiterungen (z.B. Anbindung an die traditionellen Host-Systeme) und Neuentwicklungen (z.B. im Endanwender-Bereich). Die traditionellen Systeme wurden vermutlich streng sequentiell nach dem Wasserfall-Modell² entwickelt und müssen entsprechend sequentiell getestet werden. Die Neuentwicklungen hingegen können aufgrund neuer Technologien iterativ nach einem Spiralmodell³ entwickelt werden. Ein geeignetes Testvorgehen muß beiden Ansätzen gerecht werden.

■ *Automatisierte Tests* – Neue Technologien erlauben es, Entwicklungen in mehrere Einheiten zu unterteilen oder diese wiederzuverwenden. Der Trend zu gekapselten Komponenten und der Aspekt des faster-to-market führen dazu, daß vermehrt Systeme erweitert oder Teile davon erneuert werden. Bei den nachfolgenden Regressionstests⁴ sind Automatisierungen ins Auge zu fassen.

■ *Testen vollständiger Geschäftsprozesse* – e-business-Lösungen kommen, wenn sie konsequent eingeführt werden, nicht ohne Anpassungen der Geschäftsprozesse aus. Ein lokales Unternehmen, das seine Produkte auf dem Internet anbietet, muß beispielsweise seine Prozesse so anpassen, daß Lieferungen ins und Zahlungen aus dem Ausland möglich sind. Auch muß

eine Unternehmung, die eine E-Mail-Adresse auf seiner Web-Site veröffentlicht, eine interne Support-Organisation aufbauen. Diese Neuerungen und Erweiterungen der Prozesse sind vor der Auslieferung in die Produktion vollständig mitzutesten.

nach dem Design ein Designreview etc. Dadurch werden Fehler in der Phase entdeckt, in der sie entstanden sind, und es wird ein Chaos in der letzten Testphase verhindert.

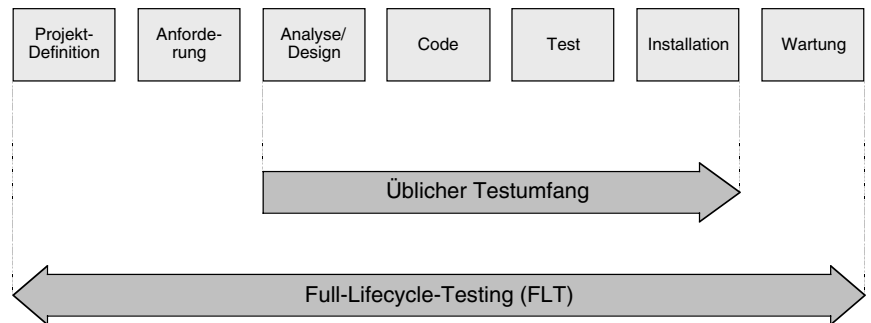


Abbildung 2: Full-Lifecycle-Testing

Full-Lifecycle-Testing – ein mögliches Testvorgehen

Im Folgenden stellen wir kurz ein Testvorgehen vor, das auf der Full-Lifecycle-Testing (FLT) Methode von IBM basiert und flexibel genug ist, um sowohl bei klassischen, als auch bei e-business-Testvorhaben eingesetzt zu werden. Wir beschränken uns dabei auf die grundlegenden Charakteristiken:

■ *Full-Lifecycle* – Das traditionelle Testen ist auf die eigentliche Konstruktionsphase beschränkt (Abbildung 2) und findet die Fehler zu spät im Entwicklungsablauf. Je später man aber einen Fehler findet, desto teurer wird seine Korrektur. FLT dehnt deshalb das Testen konsequent auf den ganzen Lebenszyklus einer Software aus und führt neben dynamischen auch statische Tests⁵ durch. Jede Phase des Entwicklungszyklus⁶ erhält dabei eine entsprechende Testphase, d.h. nach dem Erstellen der Anforderungen wird ein Anforderungsreview durchgeführt,

■ *Verschiedene Testebenen* – Das Testen läuft durch verschiedene Ebenen im Entwicklungszyklus. Jede abgeschlossene Ebene repräsentiert einen Meilenstein im Projektplan und besitzt eine bekannte Stufe an Integration und Qualität. Diese Integrationsstufen sind die Testebenen (Abbildung 3). Sie haben bestimmte Ziele und unterstützen die drei Sichtweisen der Technik, des Benutzers und des Managements. FLT ergänzt die gebräuchlichen dynamischen Testebenen durch die erwähnten Anforderungstests und Designtests.

■ *Verwendung risikobasierter Fokusse* – Das eigentliche Ziel des Testens ist es, Fehler und ihre Risiken in einem System zu minimieren. Es ist unmöglich und wirtschaftlich nicht sinnvoll, absolut vollständig zu testen. Daher werden zuerst in einer Risikoanalyse die wichtigsten Faktoren und Risiken für den Anwender, den Entwickler und den Service Provider (z.B. Internet Service Provider oder Betriebsorganisation) eruiert. Daraus

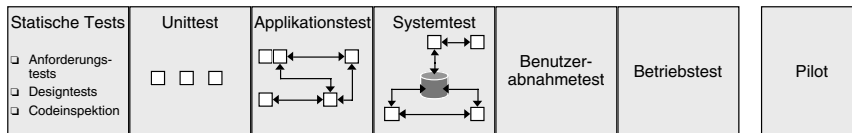


Abbildung 3: Die Testebenen des FLT

leitet der Verantwortliche sogenannte Fokuse ab, welche die Risiken adressieren und minimieren. In den Testplänen werden diese Fokuse auf die verschiedenen Testebenen abgebildet. Hier einige Beispiele von gebräuchlichen Fokussen:

Fokus	Beschreibung
Richtigkeit	Fähigkeit des Systems, die Daten gemäß vorgeschriebener Regeln zu verarbeiten und die Richtigkeit und Vollständigkeit der Daten zu gewährleisten.
Benutzbarkeit	Aufwand, der erforderlich ist, um das System zu erlernen und damit zu arbeiten.
Operabilität	Fähigkeit, das System mit wenig Bedienungsaufwand und/oder Fehleranfälligkeit zu betreiben (manuell oder automatisch).
Antwortzeiten (Performance)	Fähigkeit des Systems, gewisse Funktionen innerhalb einer vorgeschriebenen Zeit auszuführen.
Portabilität	Fähigkeit des Systems, in verschiedenen Betriebssystemumgebungen zu laufen.

Tabelle 1: Beispiele von Fokussen

■ *Iterative Entwicklung im FLT* – Die Testebenen des FLT können auch für evolutionäre Entwicklungsprozesse (iterativ und prototyping) verwendet werden, falls sich die Iterationen auf die Ebenen bis und mit dem Applikationstest beschränken (Abbildung 4). Ab dem Systemtest werden die integrativen Aspekte maßgebend. Deshalb ist das phasenmäßige Weiterführen gemäß klar definierter Übergaberegeln von den Konstruktions- in die Testebenen zwingend. Treten in diesen späteren Ebenen (z.B. dem Systemtest) Fehler auf, durchlaufen die fehlerhaften Applikationsteile nochmals

die Ebenen ab der Fehlerquelle. Die Verantwortung bleibt aber beim Systemtest-Team, welches das System in der Zwischenzeit weitertestet. Im bekannten V-Diagramm sind diese evolutionären Prozesse dem traditionellen

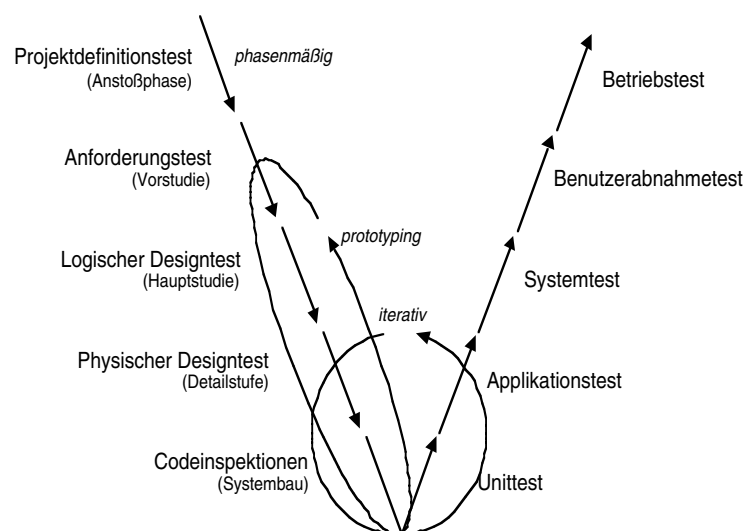


Abbildung 4: V-Diagramm mit phasenweisem und evolutionärem Vorgehen

len, phasenweisen Vorgehen überlagert.

■ *In einem Testplan definierter Testprozess* – Das Testen ist ein Prozeß, der in vier grundlegenden Schritten abläuft:

1. Planen
2. Vorbereiten
3. Durchführen
4. Abschließen

Diese Schritte werden sowohl auf der Ebene des Gesamtprojektes, als auch der einzelnen dynamischen Testebene durchgeführt und sind in einem Mastertestplan (einer pro Projekt), respektive den Detailtestplänen (einer pro Testebene) festgehalten. Sie sind Bestandteil des übergeordneten Projektplans und bedingen ein integrierendes Projektmanagement. Master- und Detailtestpläne enthalten nicht nur die zeitliche Planung der Tests, sondern machen auch u.a. Aussagen zu den Zielen, zu den Fokussen, zum Umfang, zu den Testarten, zur Infrastruktur, zu den benötigten Testdaten und zur Organisation. Spezielle Beachtung verdienen die primären Metriken *Testfalldurchführung* und *Fehlerstatus*, die den Testfortschritt und die Qualität überwachen lassen. Checklisten und

Vorlagen sind bei der Erstellung solcher Testpläne sehr hilfreich.

Vergleicht man das Full-Lifecycle-Testing-Modell mit den im vorhergehenden Kapitel geforderten Elementen, erkennt man, daß sich FLT gut für das Testen von e-business-Lösungen eignet :

- Der *Testhorizont* kann sehr leicht durch zusätzliche Fokusse erweitert werden.
- Die *Konzentration auf Risikobereiche* ist durch die vorgängige Risikoanalyse und die Definition der Fokusse gegeben.
- Der *Einbezug der Benutzer* wird im FLT durch die Testebene Benutzerabnahme erzwungen. Die statischen Tests und der streng geplante und definierte Testprozeß fördern schon eine

frühe Teilnahme der Benutzer im Entwicklungsprozeß.

- Das Konzept der Testebenen, mit der schrittweisen Integration der Anwendung, berücksichtigt *alle drei Sichten* der Technik, des Benutzers und des Managements.
- FLT läßt sowohl die *inkrementelle als auch die phasenmäßige Entwicklung* zu. Man muß sich aber bei inkrementellen Vorgehen der anspruchsvolleren Planung bewußt sein.
- Die *Automatisierung der Testdurchführung* ist keine grundlegende Eigenschaft eines Testvorgehens. Das Konzept der Testebenen von FLT, welches den gesamten Testaufwand in einzelne Ebenen mit klaren Zielen unterteilt, vereinfacht aber den Einsatz von entsprechenden Testwerkzeugen.

- Der in FLT vorgesehene Benutzerabnahmetest wird in einer möglichst produktionsnahen Testumgebung (z.B. mit Echtdateien), *basierend auf den Geschäftsprozessen*, durchgeführt.

e-business-Testing – besondere Fokusse

Wie erwähnt, können den meisten neuen Herausforderungen mit einer Risikoorientierung und den daraus abgeleiteten Fokussen begegnet werden. In e-business-Anwendungen treten daher einige Fokusse vermehrt auf und bedingen entsprechende Maßnahmen:

Fokus	Ausprägung	Mögliche Maßnahmen
Korrektheit	Funktionalität, Fehlerbehandlung, etc.	Klare Anforderungen, Problem Management, Verwendung echter Daten
Benutzbarkeit	Erlernbarkeit, Intuitives Verhalten, Ergonomie	Früher Einbezug der Benutzer, Usability-Labor
Inhalt	z. B. die korrekte Anzeige eines Produktes und der dazugehörenden Nummer, Beschreibung, Preis, etc.	GUI-Tests, Einbezug von Fachpersonen, Pilot vor der Einführung
Sicherheit	Schutz der eigenen, operationellen Systeme und der Übertragung	Hacker als freie Tester einsetzen
Kapazität (Volumina)	Bandbreite, Hitraten, Datenmengen, Transaktionszeiten	Testinfrastruktur zur Simulation, Testtools zur Lasterzeugung
Antwortzeiten (Performance)	Alle Pfade End-to-End vom Endanwender über die internen Pfade in die bestehenden Systeme und zurück	Tests über reales Internet zu verschiedenen Zeiten aus verschiedenen Domänen
Schnittstellen	Zu den internen, bestehenden Systemen	Schrittweise Integration der Systeme
Konfiguration	Verschiedene Client Konfigurationen (Browser, Applets, Sicherheits- und Spracheinstellungen etc.)	Vielzahl an Konfigurationen testen, Tests mit eigenen, (privaten) Endgeräten
Integration	Einfluß der neuen auf die bestehenden Anwendungen/Systeme	Schrittweise Integration, Pilot vor der Einführung
Regression	Automatisierte Tests	Tools

Tabelle 2: Besondere Fokusse beim e-business-Testen

e-business-Testing – einige Erfahrungen

Die vielen Erfahrungen bestätigen sowohl die speziellen Herausforderungen als auch die Notwendigkeit der Fokussierung im Bereich Testen. Anhand von konkreten Anwendungsbeispielen aus verschiedenen Branchen und Regionen werden diese Erfahrungen zusammengefaßt:

- Beispiel N – Großes Auswertungs- und Abfragesystem, Japan
- Beispiel L – Erweiterung einer bestehenden Web-Site mit Online-Bestellmöglichkeit, USA
- Beispiel M – Virtueller Einkauf für Konsumenten, Schweiz

In allen Beispielen wurde intensiv getestet. Nur das Beispiel N hatte umfassende Tests geplant und auch während der ganzen Entwicklung strukturiert durchgeführt, analog dem vorgestellten Testmodell FLT (Full-Lifecycle-Testing). Ebenso hat nur Beispiel N die Testfokuse gezielt bei den Tests berücksichtigt. Die andern haben die Testfokuse teilweise angesprochen, aber mit unterschiedlichem Aufwand und Erfolg, wie die Tabelle 3 zeigt.

Anhand dieser Anwendungsbeispiele lassen sich die Kernaussagen der vorhergehenden Kapitel bestens bestätigen, auch wenn nicht alle Erkenntnisse sich ausschließlich auf e-business beschränken:

- Das *Testmanagement* ist einer der wichtigsten kritischen Erfolgsfaktoren. Die konsequente Planung, Vorbereitung und Durchführung haben im Beispiel N zum Erfolg geführt. Im Beispiel L war die Testplanung ungenügend, der Produktionsstart mußte verschoben werden wegen inhaltlicher

Fokus	Beispiel N	Beispiel L	Beispiel M
Korrektheit	✓	✓	✓
Benutzbarkeit	✓		✓
Inhalt	✓	✓	✓
Sicherheit	✓	✓	
Kapazität	✓		
Antwortzeiten	✓	✓	✓
Schnittstellen	✓	✓	✓
Konfiguration	✓		
Integration	✓		✓
Regression	✓	✓	

Tabelle 3: Beachtung der Testfokuse von Projektbeispielen

und technischer Probleme. Mit unverhältnismäßigem Aufwand und Belastung der Mitarbeiter konnte der Betrieb doch noch erfolgreich, wenn auch verspätet, aufgenommen werden.

- Für *Benutzertests* hat sich der Einsatz von Studenten unter Anleitung von professionellen Testexperten bewährt (Beispiel N), sowie ein Pilotbetrieb mit einer Partnerfirma (M). Im dritten Fall führte die ungenügende Testplanung zu späten Tests und zur Neugestaltung der Benutzerschnittstelle (Registrierung). Der Produktionsstart mußte verschoben werden.

- Die Prüfung des *Inhalts* verdient ein besonderes Augenmerk. Im Beispiel M haben sich Probleme in der Produktion, bei der Abstimmung von Bild, Text und Preis ergeben. Die Kontrolle des Inhalts während der Produktion ist ebenfalls ratsam (z.B. mit den Teststudenten im Beispiel N).

- Die Tests der *Kapazität und Antwortzeiten* sind bei unkontrollierbarer Anzahl der Benutzer unbedingt notwendig. Tests sowohl im Labor als auch in der realen Umgebung mit auf-

gezeichneten, tatsächlichen Abläufen (Use Cases) ermöglichten die problemlose Bewältigung der Spitzenzeiten mit Hitraten von über 100 000 pro Minute (N). Beispiel M hat die Performance durch eine schrittweise Einführung und der präventiven Lastverteilung der Web- und Datenbank-Server kontrolliert.

- Die *Konfigurationstests* wurden in den Beispielen L und M reduziert durch eine limitierte Anzahl unterstützter Konfigurationen. Als Folge habe sich dafür im Beispiel L Kunden über inkompatible Browser beklagt. Beispiel N hat, soweit möglich, Tests im Labor durchgeführt. Trotzdem sind bei "exotischen" Konfigurationen Probleme durch laufend unterbrochene Verbindungen und lange Blockierung von Ressourcen aufgetreten.

- Die *Integration* der beteiligten Systeme wurde sorgfältig getestet (N). Die Anzahl der Verknüpfungen von verschiedenen traditionellen Systemen, sowie die Ablaufvielfalt wurde in L und M unterschätzt. Dadurch sind

Probleme inhaltlicher und technischer Art aufgetreten.

Diese Erfahrungen haben die Verantwortlichen der Beispiele L und M veranlaßt, in Zukunft ein strukturierteres und zielgerichteteres Testvorgehen mit statischen Tests und Testfokussen vorzusehen, mit genügend qualifizierten Testpersonen, unter Berücksichtigung der Sichten Technik und Benutzer.

Zusammenfassung

Ohne Qualitätsmanagement ist ein e-business-Projekt nicht erfolgreich durchführbar. Sowohl die konstruktive Qualitätssicherung (Verhinderung von Fehlern) als auch die analytische Qualitätssicherung (Entfernung von Fehlern, Testen) sind kritische Erfolgsfaktoren. In diesem Artikel haben wir die wichtigsten Aspekte des Testvorgehens besprochen und gezeigt, wie den neuen Herausforderungen (vor allem schnellere Entwicklung bei besserer Qualität) durch ein erprobtes Full-Lifecycle-Testverfahren risikobewußt, effektiv und erfolgreich begegnet werden kann.

Literatur

- [Bach 1996] Bach James: Testing Internet Software. In: American Programmer, December 1996, P. 12–17.
- [Dibachi 1999] Dibachi Rhonda: Testing e-commerce. In: Software Testing & Quality Engineering, March/April 1999, P. 56–62.
- [Frühauf, Ludewig, Sandmayr 1997] Frühauf Karol, Ludewig Jochen, Sandmayr Helmut: Software-Prüfung; Eine Anleitung zum Test und

zur Inspektion. Zürich: vdf Hochschulverlag an der ETH, 1997.

- [Jones 1997] Jones Capers: Applied Software Measurement; Assuring Productivity and Quality. New York: McGraw-Hill, 1997, Second edition.

- [Jones 1994] Jones Capers: Assessment and Control of Software Risks. Englewood Cliffs NJ: Yourdon Press, 1994.

- [Haist, Fromm 1991] Haist Fritz, Fromm Hansjörg: Qualität im Unternehmen; Prinzipien – Methoden – Techniken. München, Wien: Hanser, 1991.

- [Hetzl 1988] Hetzel Bill: The Complete Guide to Software Testing. Weelesley MA: QED Information Sciences, 1988.

- [Humphrey 1997] Humphrey Watts S.: Introduction to the Personal Software Process. Reading MA: Addison-Wesley 1997.

- [Myers 1979] Myers Glenford J.: The Art of Software Testing. New York: Wiley, 1979.

- [Perry 1995] Perry William: Effective Methods for Software Testing. New York: Wiley, 1995.

- [QA Institute 1990] QA Institute Seminars-Workshops: Effective Methods of System Testing. Florida: 1990.

- [Wallmüller 1990] Wallmüller Ernest: Software-Qualitätssicherung in der Praxis. München, Wien: Hanser, 1990.

Endnoten:

¹ Im Usability-Labor wird das Benutzerverhalten untersucht und ausgewertet in Bezug auf Abläufe, abgestimmter Einsatz von Tastatur und Maus, Augenbewegungen und deren psychologische Auswirkungen.

² Das Wasserfall-Modell nach Boehm (1976) beschreibt den traditionellen Soft-

ware-Lebenszyklus, wobei die Phasen der Entwicklung (Analyse, Design, Implementierung, Test, Wartung) sequentiell durchlaufen werden und aufeinander aufbauen.

³ Das Spiralmodell nach Boehm (1986) beschreibt ein iteratives Vorgehen, wobei die einzelnen Phasen der Entwicklung mehrere Male und in einer definierten Reihenfolge durchlaufen werden.

⁴ Regressionstests prüfen, ob unveränderte Anwendungsteile weiterhin fehlerlos funktionieren, trotz anderweitiger Änderungen.

⁵ *Dynamisches Testen* heißt das Durchführen von physischen Programmen am System. *Statisches Testen* heißt die Verifikation von Zwischenresultaten “am Schreibtisch” durch Reviews, Code Inspektionen, usw., ohne Durchführung von Programmen.

Christoph Brian, Dipl. El.-Ing. ETH, IBM Consulting Group Schweiz, Basel
Markus Büchi, Dipl. El.-Ing. ETH, IBM Consulting Group Schweiz, Zürich