



□ Kerstin Nagel

[E-Mail:kerstin.nagel@mks.de]

Leiterin Marketing bei MKS Deutschland und verantwortlich für das Marketing in Mittel- und Südeuropa. Sie verfügt über langjährige Erfahrung in der IT, mit Schwerpunkt in den Bereichen Software und Netzwerktechnik.

Bessere Softwarequalität durch anforderungsbasiertes Testen

Studien haben ergeben, dass unvollständige, schlecht geschriebene oder auch fehlerhaft übermittelte Anforderungen für 50 bis 70 % aller gescheiterten Softwareprojekte verantwortlich sind. Die Grundursachen von 56 % aller in Projekten identifizierten Fehler entstehen bereits in der Anforderungsphase [Mog03].

Im Hinblick auf diese Zahlen ist es naheliegend, dass viele Unternehmen versuchen, solche Risiken zu verringern, indem sie formale Anforderungsmanagementprozesse einführen, die auch in der Qualitätssicherung von Nutzen sind. Denn durch eine anforderungsbasierte Testmethode, die sicherstellt, dass sämtliche Anforderungen getestet werden, können die Effizienz erhöht, Defect-Risiken verringert und die Softwarequalität insgesamt verbessert werden.

Ausgangspunkt des anforderungsbasierten Testens ist das Anforderungsmanagement. Zu Beginn des Entwicklungsprozesses erstellen und verwalten Business Analysten, Anwender, Produktmanager und andere Verantwortliche Anforderungs- und Spezifikationsdokumente, um das gewünschte System genau und vollständig zu beschreiben. Die Qualitätssicherung nutzt dann diese Anforderungen als Grundlage für die Planung und Durchführung ihrer Tests. Die Planung und Ausführung von Testfällen können dadurch mit den Anforderungen abgeglichen werden, die Qualitätssicherung kann so den Projektfortschritt messen und gewährleisten, dass alle Anforderungen überprüft werden. Sobald alle Anforderungen sorgfältig getestet, eventuell noch weitere codebasierte Tests durchgeführt wurden und die während des Testdurchlaufs ermittelten Fehler behoben sind, erteilt die Qualitätssicherung eine formale Freigabe.

Mit anforderungsbasiertem Testen lassen sich Probleme also schon frühzeitig im Anwendungslebenszyklus feststellen. Und je früher ein Problem erkannt wird, desto geringer sind die Kosten zur Beseitigung des Fehlers, denn die Kosten zur Behebung von Fehlern steigen im Laufe eines Softwareprojekts drastisch an. Wenn ein Problem bereits in der Anforderungsphase erkannt

wird, ist das Kosten-Nutzen-Verhältnis der Fehlerbehebung 1 zu 1, in der Programmierungsphase schon 7 zu 1 und in der Produktionsphase zwischen 40 und 100 zu 1 [Boe].

Trotz dieser Tatsache setzen derzeit viele Unternehmen noch keine anforderungsbasierten Testmethoden ein. Dies führt oft zu Problemen unterschiedlichster Art, die jedoch alle Auswirkungen auf den Projektfortschritt und die Softwarequalität haben.

So werden für die Verwaltung von Testprozessen oft noch manuelle Methoden verwendet. Dies bedeutet in der Regel einen enormen Zeitaufwand für die Erstellung von Berichten und die Evaluierung des Teststatus. Darüber hinaus ist die Überprüfung der Anforderungen schwierig, da

keine direkte Verbindung zu den Testfällen besteht. Die Auswirkungen von Anforderungsänderungen können kaum ermittelt werden, was immer wieder zu Projektunterbrechungen führt. Außerdem sind die Tests oft ungenau, da sich die Anforderungen im Laufe des Projekts geändert haben, ohne dass dies entsprechend kommuniziert und in den Tests berücksichtigt wurde.

Die Unternehmen sind sich dieser Probleme jedoch durchaus bewusst und führen daher oft nach und nach Insellösungen ein, um Fehlerquellen zu verringern. Dabei entstehen aber immer wieder neue Probleme, denn durch unterschiedliche Produkte für das Anforderungs-, Test- und Defect-Management entstehen automatisch getrennte Datenbanken. Getrennte Datenbanken bedeuten aber,

Cost to find and fix a requirements defect by phase	Phase	Average Cost to Repair Defect	Normalized Values
	Requirements	\$139	1
	Design	\$455	3
	Coding	\$1,000	7
	System Testing	\$7,000	50
	Production	\$14,000	100

Abb. 1: Die Kosten zur Behebung von Fehler steigen im Laufe des Projekts an [Boe]

dass Änderungen in einem Bereich in einem anderen Bereich nicht berücksichtigt werden, dass bereichsübergreifende Prozesse nicht umsetzbar sind und nicht zuletzt, dass es keine alle Phasen des Anwendungslebenszyklus umfassende Reportingquelle gibt, die Projekt- oder Abteilungsleiter für Auswertungen nutzen können.

Die einzige Möglichkeit, all diese Probleme zu überwinden, ist der Einsatz einer einheitlichen Plattform, die alle Bereiche des Anwendungslebenszyklusses abdeckt. Denn erst wenn alle Artefakte in einer einzigen Datenbank hinterlegt sind, können die unterschiedlichen Entwicklungsphasen miteinander verbunden werden, ohne dass Informationen nicht übermittelt werden oder sogar verloren gehen. Eine der wenigen Plattformen dieser Art, die es am Markt gibt, ist MKS Integrity vom kanadischen Softwarehersteller MKS (www.mks.de).

Alle Entwicklungsphasen in einem Produkt

MKS Integrity ist eine Application Lifecycle Management-Plattform, die alle Aktivitäten und Artefakte im Zusammenhang mit der Softwareentwicklung koordiniert und verwaltet. Die Plattform umfasst Requirements Management, Software Configuration Management, Change Management, Test Management, Defect Management, Release Management und Portfolio Management.

MKS Integrity bietet einheitliche Funktionen über alle Bereiche des Application Lifecycles hinweg. Es definiert Hierarchien, Beziehungen und Change Management-Prozesse für zahlreiche nutzerdefinierte Anwendungsartefakte. Es gibt zwar auch andere Hersteller, die diese Funktionalitäten anbieten, das Besondere an MKS Integrity ist jedoch, dass es im Gegensatz zu den Konkurrenzprodukten, die durch Zukäufe entstanden sind, von Anfang an als ein einziges Produkt konzipiert wurde. Es basiert also auf einer Architektur und entsprechend werden alle Vorgänge im Entwicklungszyklus in einer einzigen Datenbank abgelegt.

Der Prozess, mit dem in der Regel die Entwicklung einer Software beginnt, ist das Anforderungsmanagement. Viele Unternehmen definieren ihre Anforderungen noch in Business-Tools wie Microsoft Word, Excel oder Project. Diese Anforderungsdokumente können ohne großen Aufwand in MKS Integrity importiert werden. Alternativ können die Benutzer die Vorteile der Document View von MKS Integrity nutzen, um Anforderungen direkt innerhalb des Sys-

tems zu schreiben. Die Document View ist eine Oberfläche, die Microsoft Word sehr ähnlich ist. Der Nutzer kann Fließtext verfassen, seine gewohnten Formatierungen wie Schriftgröße und Schriftart durchführen sowie Grafiken und Tabellen einfügen. Sind die Anforderungen erst einmal im System vorhanden, wird über den vorab individuell definierten Workflow der Freigabezyklus für diese Anforderungen gestartet. Sobald eine Freigabe erfolgt, können Testautoren – ähnlich wie Entwickler bei der Erstellung von Pflichtenheften – Testfälle erstellen, die direkt mit den Anforderungen verknüpft sind. Durch diese direkte Verbindung mit den Bereichen Entwicklung und Qualitätssicherung können vorgeschlagene Anforderungsänderungen auf ihre Auswirkungen hin analysiert werden. Zudem wird sichergestellt, dass alle vorgenommenen Anforderungsänderungen an nachgeschaltete Bereiche wie Design und Test kommuniziert werden.

Testplanung auf Basis von Anforderungen

Sobald eine Softwareanwendung durch eine Reihe von Anforderungen definiert und entwickelt wurde, kann die Qualitätssicherung einen gezielten, direkt auf diesen Anforderungen basierenden Test durchführen und dabei, wenn gewünscht, bei neuen Entwicklungsständen sicherstellen, dass das System mit den ursprünglichen Parametern verglichen wird und den Geschäftsanforderungen entspricht.

Der nächste Schritt beginnt mit der Testentwicklung. Testautoren bereiten einen oder mehrere Testfälle vor, mit denen jede Anforderung geprüft wird. Dadurch wird die Anforderung in der endgültigen Version berücksichtigt. Die Testfälle werden dann um detaillierte Informationen zu den Testschritten sowie um die erwarteten Ergebnisse und Pass-/Fail-Kriterien erweitert, die für die Evaluierung der Testergebnisse verwendet werden. Für die automatisierte Prüfung wird eine Verbindung zum Test-Skript im externen Testausführungstool erstellt.

Der Vorteil dieser Vorgehensweise liegt auf der Hand: Durch die Verknüpfung der Anforderungen mit den Testfällen entsteht eine Feedbackschleife zwischen den Business Analysten und der Qualitätssicherung. Diese Feedbackschleife trägt zur Präzisierung der Anforderungen und zur Verbesserung der Produktqualität bei. Durch Rückfragen kann ermittelt werden, was genau getestet werden soll. Dadurch werden Fehler oder Lücken in den Anforderungen aufgedeckt.

Der Prozess, der alle anderen Entwicklungsprozesse überlagert, ist das Change Management. Ein effektives Change Management ist für den Erfolg eines Unternehmens von zentraler Bedeutung. Die Prozesssteuerungsfunktionen von MKS Integrity sorgen dafür, dass nur autorisierte Anwender Änderungen vornehmen können. Durch eine umfassende Auswirkungsanalyse können zudem vor der Annahme einer Änderung die mit ihr verbundenen Risiken ermittelt werden. Wurde eine Änderung

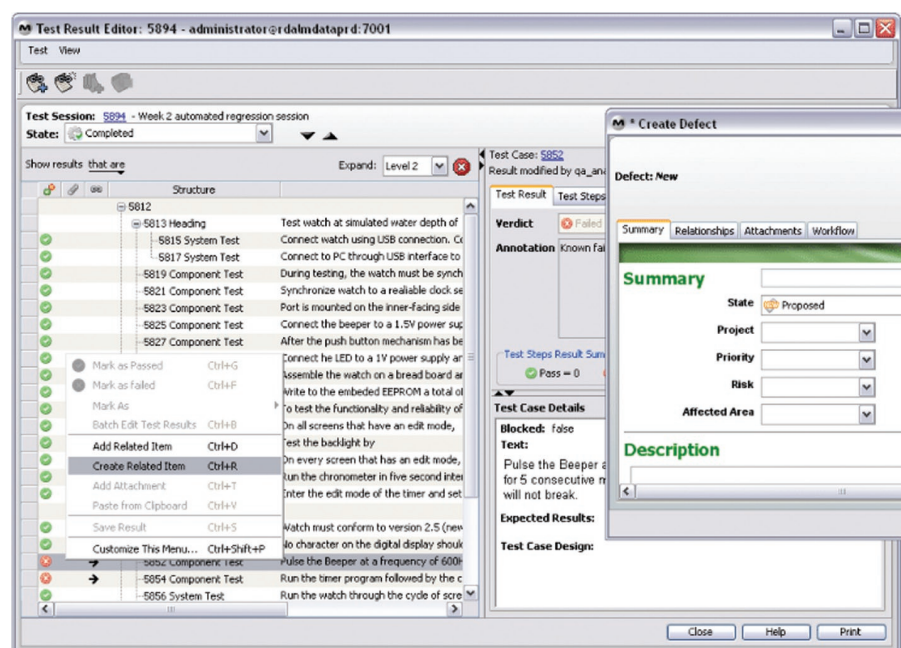


Abb. 2: Anlegen eines Defects, der automatisch mit einem fehlgeschlagenen Testfall und der zugehörigen Anforderung verknüpft ist

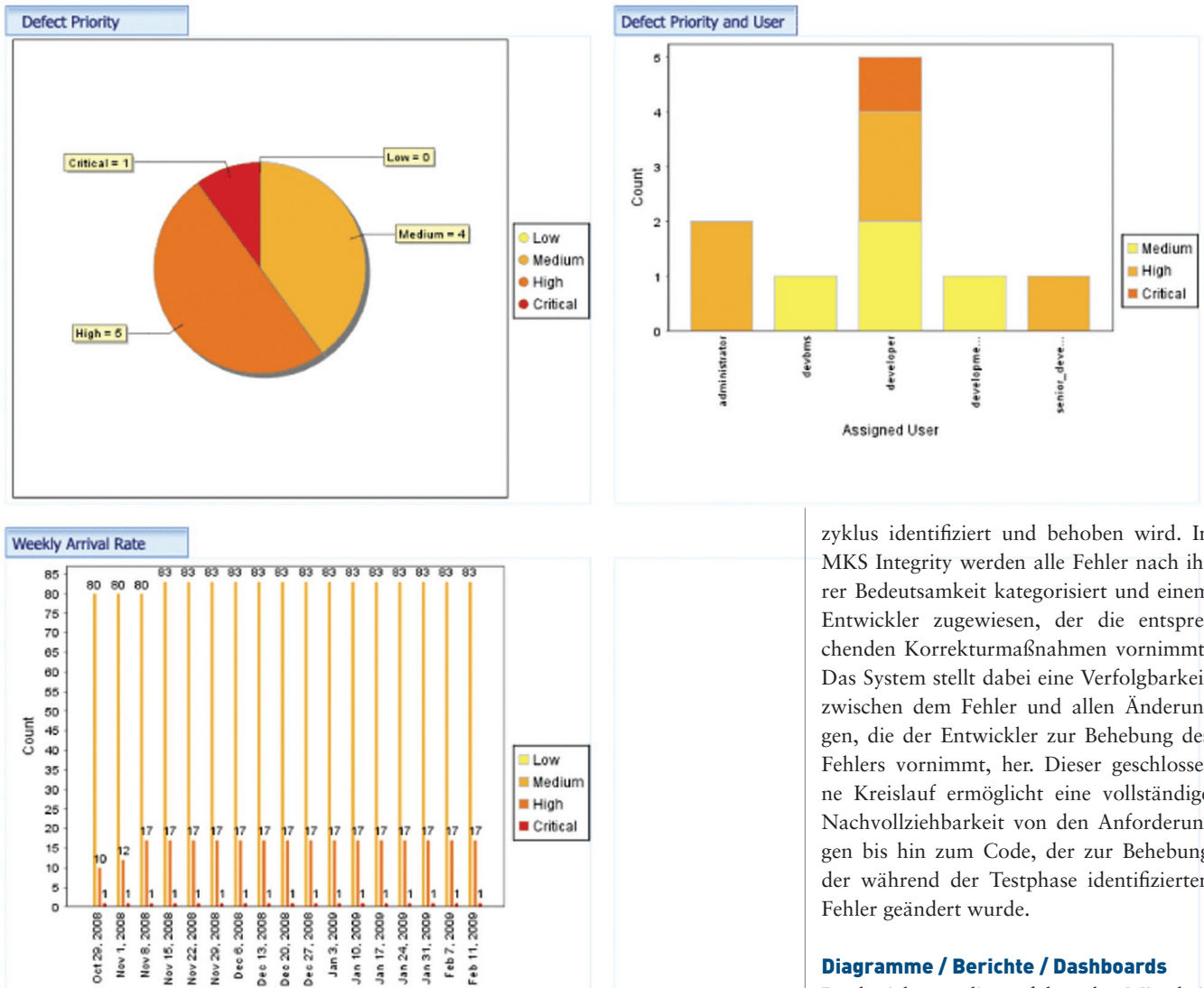


Abb. 3: Diagramme liefern Echtzeit-Einblicke in kritische Testmetriken

akzeptiert, dann werden Elemente, auf die sich diese Änderung auswirken könnte, zur Überprüfung markiert, sodass Korrekturmaßnahmen ergriffen werden können.

Automatische und manuelle Tests zuverlässig durchführen

Sobald die Testfälle vorbereitet und logisch gruppiert sind, beauftragt ein Manager der Qualitätssicherung einen Testingenieur mit der Durchführung der Tests. Im Falle einer automatisierten Prüfung interagiert MKS Integrity über sein Automated Test Execution Framework mit externen Testausführungsprodukten wie HP Quick Test Professional von Hewlett Packard und iTKO LISA von Penta-T. MKS Integrity baut dazu eine Verbindung mit dem externen System auf, liefert ihm die Informationen, die zur Ausführung des Tests benötigt werden und erfasst die Ergebnisse. Die Testergebnisse

werden dann in MKS Integrity zur Prüfung und Interpretation angezeigt.

Zur Ausführung manueller Tests verwenden Testingenieure den Test Results Editor. Sie erhalten damit eine präzise Übersicht über die auszuführenden Testschritte sowie die erwarteten Ergebnisse und die Pass-/Fail-Kriterien. Zu den Ergebnissen können detaillierte Kommentare protokolliert werden. Auch die Ergebnisse der vorab von den zuständigen Testingenieuren definierten automatisierten Prüfungen können über diesen Editor angezeigt werden.

Defects beheben und Projektkosten senken

Eines der Schlüsselkriterien für den Reifegrad eines Produkts ist die Anzahl und der Schweregrad der offenen Defects. Wie bereits erwähnt, sind die Kosten geringer, je früher ein Fehler im Entwicklungslebens-

zyklus identifiziert und behoben wird. In MKS Integrity werden alle Fehler nach ihrer Bedeutsamkeit kategorisiert und einem Entwickler zugewiesen, der die entsprechenden Korrekturmaßnahmen vornimmt. Das System stellt dabei eine Verfolgbarkeit zwischen dem Fehler und allen Änderungen, die der Entwickler zur Behebung des Fehlers vornimmt, her. Dieser geschlossene Kreislauf ermöglicht eine vollständige Nachvollziehbarkeit von den Anforderungen bis hin zum Code, der zur Behebung der während der Testphase identifizierten Fehler geändert wurde.

Diagramme / Berichte / Dashboards

Doch nicht nur die ausführenden Mitarbeiter in der Softwareentwicklung profitieren von den Funktionen von MKS Integrity. Manager in der Qualitätssicherung, Projektmanager und Geschäftsführer sind auf genaue und zeitnahe Informationen angewiesen, um Ressourcen planen, schnelle Korrekturmaßnahmen vornehmen und allgemein Pläne und Budgets verlässlich verwalten zu können. Allzu oft werden Berichte manuell erstellt, was meist zu Verzögerungen führt. Zudem besteht aufgrund des „Faktors Mensch“ bei der Erfassung der Daten das Risiko der Subjektivität. Insellösungen verfügen jeweils über ein eigenes Daten-Repository und können daher keine Gesamtübersicht über den Anwendungslebenszyklus liefern.

MKS Integrity enthält umfangreiche Funktionen zur Erstellung von Grafiken, Diagrammen und Berichten. Funktionsspezifische Dashboards zeigen die von den einzelnen Teammitgliedern benötigten Informationen in Echtzeit an. Die Informationen können für ein einzelnes Projekt oder über Filtereinstellungen für alle relevanten

Projekte angezeigt werden. Mit einem Klick lassen sich die Lebenszyklusdaten, die sich hinter einem Diagramm oder Bericht verbergen, über ein Drop-Down-Menü anzeigen. Wenn zum Beispiel ein Manager der Qualitätssicherung in einem Diagramm drei offene Fehler von hoher Priorität erkennt und den entsprechenden Bereich anklickt, wird er direkt zu den Fehlern geleitet. Diese kann er genauer untersuchen und sofort Korrekturmaßnahmen anstoßen.

MKS Integrity ist also tatsächlich ein Allround-Tool, das allen an der Softwareent-

wicklung beteiligten Mitarbeitern eines Unternehmens nützt. Die automatischen anforderungsbasierten Testprozesse unterstützen in hohem Maße speziell die Qualitätssicherung und tragen dazu bei, dass freigegebene Softwareprojekte die Termin-, Kosten- und Geschäftsanforderungen erfüllen. Mit dem richtigen Application Lifecycle Management-Produkt wird Software also besser und sicherer. Jedes Unternehmen, das diese beiden Ziele erreichen möchte, sollte daher den Einsatz eines solchen Tools in Erwägung ziehen. ■

□ Referenzen

[Mog03] Gary E. Mogyorodi: „What Is Requirements-Based Testing?“, Bloodworth Integrated Technology, Inc. STSC Crosstalk, März 2003.

[Boe] B. Boehm und V. Basili: „Software Defect Reduction Top 10 List,“ - IEEE Computer.
