Entwickeln qualitativ hochwertiger Anwendungen mit Visual Studio 2010

März 2010

Die Qualität ist im Softwaregeschäft von äußerster Wichtigkeit. Es ist eine Tatsache, dass die Kosten für die Problembehebung wesentlich höher sind, wenn die Probleme nicht rechtzeitig erkannt werden. Beispielsweise gilt in der Agile-Community das Motto „Failing Fast“, d.h. kontinuierliche Integration und fortlaufendes Testen werden als grundlegende Verfahren betrachtet. Visual Studio 2010 bietet eine umfassende Anzahl an Innovationen für den Bereich Qualitätssicherung und Test, damit Softwareteams überragende Ergebnisse erzielen können.

Gemäß dem *National Institute of Standards and Technologies* kostet schlechte Softwarequalität die Wirtschaft in den USA jährlich ca. 59 Milliarden US-Dollar[1]. Diese Kosten werden durch mangelnde Produktivität und vergeudete Ressourcen verursacht. Die Studie ergab, dass mit einer besseren Testinfrastruktur ein Drittel der Kosten eingespart werden könnten. Dem Standish Group Chaos Report (2009)[2] zufolge nahm die Anzahl der gefährdeten bzw. gescheiterten Projekte letztes Jahr erneut zu (siehe Abbildung 1).

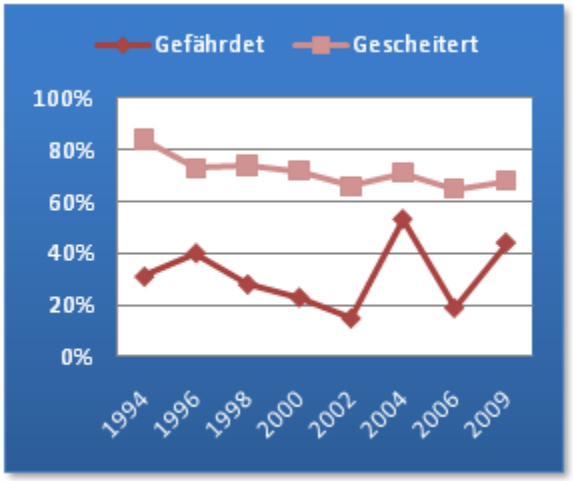


Abbildung 1: Chaos Reports zeigen keine wesentliche Verbesserung hinsichtlich erfolgreicher Projekte.

Es gibt zwar keine Patentlösung, aber es gibt bewährte Tools und Prozesse mit denen Organisationen das Scheitern von Projekten vermeiden können. Letztendlich zählt bezüglich der Qualität, ob der Kunde mit dem Endprodukt zufrieden ist. Das heißt, dass das Endprodukt die vom Kunden erwartete Funktionalität bietet, die Effizienz und Effektivität der täglichen Arbeit erhöht und keine Fehler aufweist, die bei der Arbeit mit der Software von den eigentlichen Aufgaben ablenken. Visual Studio 2010 wurde entwickelt, um Entwicklungsteams bestmöglich in die Lage zu versetzen, qualitativ hochwertige Software zu produzieren.

ERFÜLLEN DER KUNDENANFORDERUNGEN

Das Erfassen der Kundenanforderungen ist unabhängig von den verwendeten Tools schwierig, da die Anforderungen oft auf unvorhersehbaren Kundenwünschen basieren. Stabilität kann jedoch erreicht werden, wenn die Anforderungen des Kunden protokolliert und bis hin zu den Testfällen, den Testergebnissen und den festgestellten Softwarefehlern überwacht werden. Für eine vollständige Nachverfolgung ist es erforderlich, die Anforderungen mit der höchsten Priorität zu bestimmen, den Fortschritt bezüglich der Anforderungen (abgeschlossene und verbleibende Arbeit) zu überwachen und die Qualität der Anforderungen im Auge zu behalten. Eine Anforderung kann beispielsweise nicht als abgeschlossen gelten, wenn die Implementierung zahlreiche Fehler aufweist und die Kundenanforderung entsprechend der allgemeinen Qualitätskriterien nicht erfüllt ist. TFS ermöglicht eine derartige vollständige Nachverfolgung. Die gewünschten Informationen können mit den umgehend verfügbaren und anpassbaren Berichtsfunktionen schnell zusammengestellt werden.

---

[1] National Institute of Standards and Technology. (2002). *Planning Report 02-3, The Economic Impacts of Inadequate Infrastructure for Software Testing.* U.S. Department of Commerce.
[2] Standish Group, (2009). *Chaos Report*.

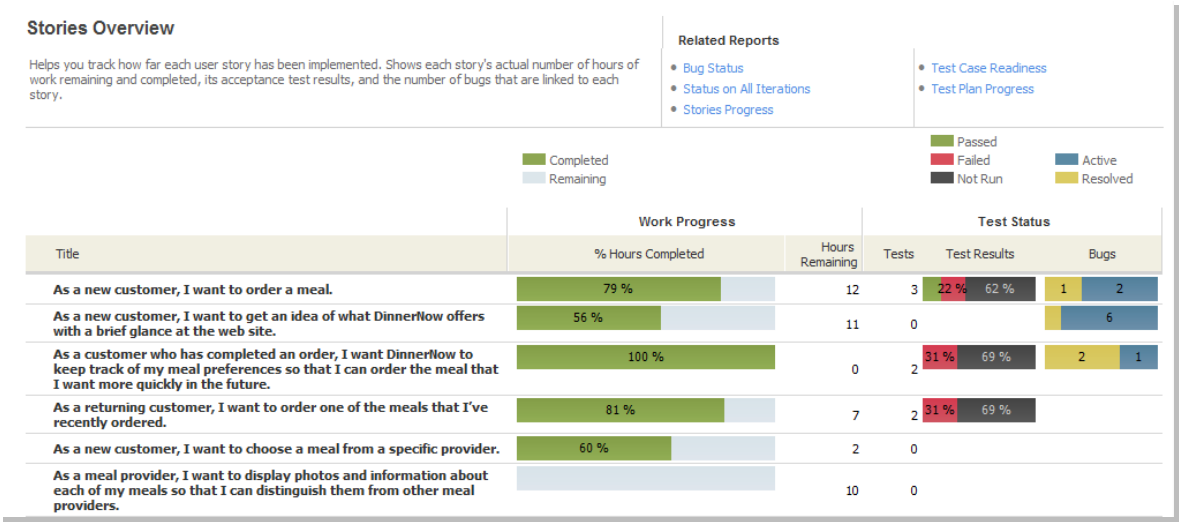


Abbildung 2: Der Bericht „Übersicht über Stories“

Auf den ersten Blick erscheint der in Abbildung 2 dargestellte Bericht ziemlich schlicht zu sein. Dieser Bericht enthält jedoch eine Menge an wichtigen Daten, z.B. die Anzahl der aktiven und behobenen Probleme für jede Anforderung. Es gibt nun kein Rätselraten mehr bezüglich der erfüllten Kundenanforderungen. Sie haben die endgültige Antwort unmittelbar vor sich.

DREI PHASEN ZUR GEWÄHRLEISTUNG DER QUALITÄT

Die Entwicklung qualitativ hochwertiger Anwendungen umfasst viele Schritte. Sie können den gesamten Prozess jedoch auf drei wesentliche Phasen reduzieren: Design, Entwicklung und Testen.

QUALITÄT BEGINNT MIT DEM DESIGN...

Viele Architekturtools sind von der Entwicklung, den Anforderungen und den Testtools getrennt. Dies macht das Erstellen von Systemarchitekturen ausgesprochen schwierig. Es ist deshalb keine Seltenheit, dass Systemarchitekturen verworfen oder ignoriert werden, sobald die Programmierung beginnt. Entwickler müssen sich ständig auf andere Dokumente oder Tools beziehen, was sie letztendlich unterlassen. Dies hat zur Folge, dass das Design nicht mit dem eigentlichen Code übereinstimmt. Die Nachvollziehbarkeit der Implementierung ist so auf technischer Ebene fast unmöglich. Dies wirkt sich auch auf die Systemwartung aus. Wenn man ein System nicht versteht, gestaltet sich dessen Wartung meist schwierig. Visual Studio 2010 Ultimate umfasst jedoch neue Architekturtools auf der Basis der UML-Sprache (Unified Modeling Language), mit denen Projektteams nicht nur Softwarelösungen in ihren Entwicklungsumgebungen entwerfen, sondern auch die Designs mit den Anforderungen verknüpfen können, damit diese einfach referenziert und auf dem neuesten Stand gehalten werden können. In Abbildung 3 sind ein Anwendungsfall- (Use

Case) und ein UML-Aktivitätsdiagramm (UML Activity Diagram) sowie ein neues Architekturtool, das Ebenendiagramm (Layer Diagram), dargestellt.

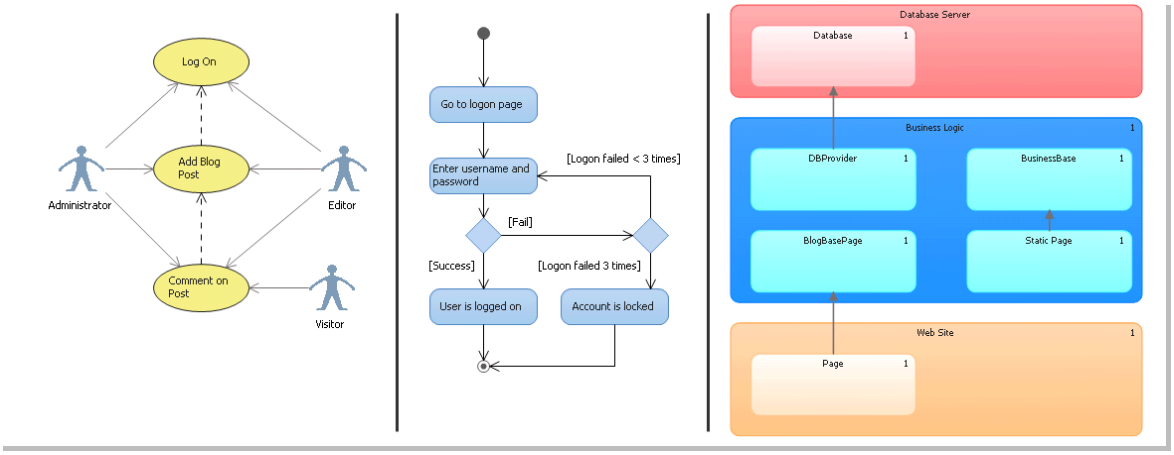


Abbildung 3: Anwendungsfall, Aktivitäts- und Ebenendiagramm.

Das Ebenendiagramm definiert nicht nur das Design der Funktionsbereiche und Komponenten (bei einer Ebene kann es sich beispielsweise um eine Methode, eine Klasse, einen Namespace, eine Assembly oder eine willkürliche Gruppierung handeln) sowie deren Kommunikationspfade, sondern es kann auch verwendet werden, um Abweichungen vom Design zu begrenzen. Werden Abweichungen vom ursprünglichen Entwurf anhand bestimmter Zielvorgaben festgestellt, kann entweder das Design angepasst oder der Code geändert werden.

WIRD MIT DEN ENTWICKLERN FORTGESETZT...

Visual Studio 2010 umfasst viele Tools, mit denen Entwickler sicherstellen können, dass die Kundenanforderungen tatsächlich erfüllt werden. Features wie die statische Codeanalyse für .NET-Code und Datenbankcode bieten Einblick in potenzielle Leistungs-, Stabilitäts-, Wartbarkeits-, Skalierbarkeits- und Sicherheitsprobleme, bevor diese zu wirklichen Problemen werden. Mit Hilfe von Codemetriken können Entwickler komplexe Codeabschnitte identifizieren und den Code anpassen, bevor er zu unübersichtlich wird. Außerdem können Entwickler umfassende Leistungstests des Codes ausführen, unabhängig davon, ob es sich um eine WPF-Anwendung oder eine ASP.NET-Anwendung handelt, um sicherzustellen, dass die Anwendungen die Skalierbarkeitsanforderungen der Kunden erfüllen. In Abbildung 4 ist ein Lasttest dargestellt. Zu den verschiedenen Aspekten der getesteten Systeme werden detaillierte Informationen angezeigt. Auf diese Weise wird sichergestellt, dass genügend Informationen für die Erkennung von Leistungsproblemen erfasst werden. Außerdem können Sie Lasttests miteinander vergleichen, um die Auswirkungen von Änderungen bezüglich der Leistungsfähigkeit im Zeitablauf zu bestimmen.

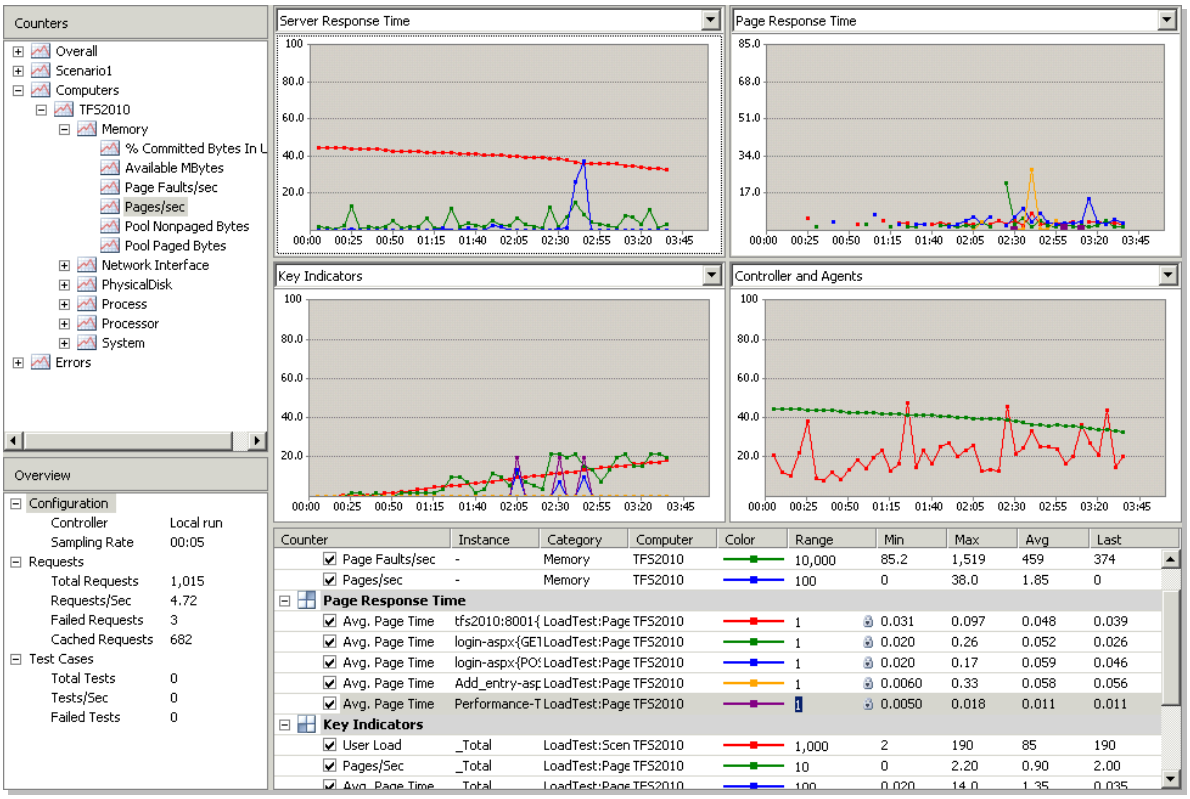


Abbildung 4: Ergebnisse eines Lasttests

Werden Leistungsprobleme während eines Lasttests festgestellt, lassen sich die entsprechenden Anwendungsbereiche mit den problematischen Codeabschnitten in detaillierten Berichten analysieren. In Abbildung 5 ist die Benutzerauslastung einer Anwendung während eines Leistungstests dargestellt. Die Abbildung stellt nicht nur die Benutzerauslastung selbst, sondern auch die aufgerufenen Seiten sowie alle Ausnahmen und anderen aufgetretenen Fehler dar.

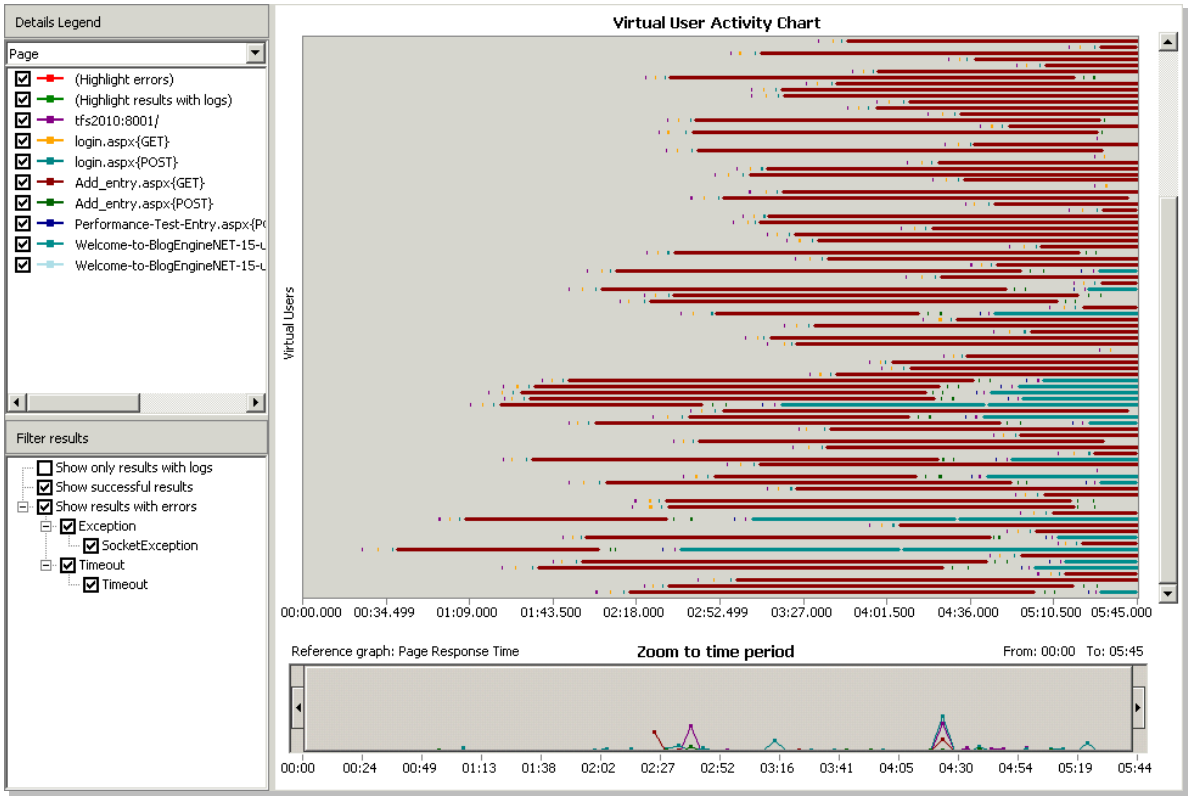


Abbildung 5: Ein Diagramm zu den Aktivitäten virtueller Benutzer während eines Lasttests

Um ein möglichst fehlerfreies Endprodukt sicherzustellen, müssen die Entwickler Probleme frühzeitig in ihren Debugsitzungen ermitteln und beheben können. Dies und vieles mehr ermöglicht IntelliTrace™, eine der neuesten Innovationen von Microsoft. IntelliTrace™ bietet Entwicklern die Möglichkeit, eine Anwendung auszuführen, die Anwendung an einer beliebigen Stelle anzuhalten und in der Debugsitzung vorwärts oder rückwärts zu gehen, um den Anwendungsstatus zu einem bestimmten Zeitpunkt zu überprüfen. Unter anderem zeichnet IntelliTrace™ den kompletten Verlauf der Stapelüberwachung (Stack Trace) auf, damit die Entwickler keine willkürlichen Haltepunkte an den hoffentlich richtigen Stellen festlegen müssen. Außerdem können die Entwickler beim Auftreten einer Ausnahme sofort an die entsprechende Stelle springen, die Problemursache identifizieren und das Problem beheben. Darüber hinaus können die Tester für die kompilierte Anwendung IntelliTrace™-Protokolle erstellen, damit die Entwickler beim Auftreten von Problemen alle Vorgänge analysieren können, die während des jeweiligen Tests aufgetreten sind. Die Entwickler sind somit in der Lage, Probleme schneller als jemals zuvor zu reproduzieren und zu beheben.

Es zählt zu den üblichen Aufgaben von Entwicklern, vor dem Einchecken des Codes in ein Versionskontrollsystem und dessen Weitergabe an die Tester einige Tests auszuführen. Dabei können die Entwickler anhand von Komponententests (Unit Tests) feststellen, ob der Code erwartungsgemäß ausgeführt wird. Bei Entwicklungsteams, die viele Komponententests ausführen müssen (d.h. testorientierte Entwicklungsteams), kann eine Analyse der von den Codeänderungen

betroffenen Tests (Test Impact Analysis, TIA) helfen, die Anzahl der auszuführenden Tests zu reduzieren. Außerdem lässt sich mit TIA sicherstellen, dass die erforderlichen Tests auch tatsächlich ausgeführt wurden. TIA basiert auf einer Analyse der Codeänderungen und setzt diese mit Komponententests in Beziehung, um zu bestimmen, ob eine Codeänderung die Ergebnisse eines zuvor ausgeführten Tests beeinflusst. Auf diese Weise können die Entwickler ihren Code stichprobenweise überprüfen, bevor dieser an die Tester oder das QA-Team weitergegeben wird.

Außerdem können die Entwickler mit Visual Studio 2010 codegestützte Tests der Benutzeroberfläche (Coded UI Tests) ausführen und so das Testen von UI-Elementen automatisieren. Visual Studio 2010 kann den Testcode dabei anhand eines von einem Tester oder Entwickler zuvor ausgeführten manuellen Tests automatisch generieren. Der codegestützte Test kann dann manuell ausgeführt oder in den Buildprozess einbezogen werden, um UI-Regressionstests zu automatisieren. Visual Studio Lab Management stellt überdies eine virtualisierte Testumgebung bereit, in der die Entwickler ihren Code auf neu installierten Systemen ausführen können, die mit den für die jeweilige Anwendung definierten Zielumgebungen vergleichbar sind. Den Entwicklern stehen also leicht zu replizierende Umgebungen zur Verfügung, die das Testen und Experimentieren erheblich vereinfachen.

Die Automatisierung des Buildprozesses war schon immer eine Hauptfunktion von Team Foundation Server (TFS). In TFS 2010 vereinfachen jetzt neue Features das Auffinden von Codedefekten, bevor sich diese in den Hauptcode einschleichen können. TFS 2010 stützt sich dabei auf Windows Workflow Foundation 4.0, um leistungsstarke Workflowfunktionen zu unterstützen, deren Verwendung quasi nur von Ihrer Fantasie begrenzt wird. Eines der innovativsten Features ist die Unterstützung von abgegrenzten Eincheckbuilds (Gated Check-In), eine wichtige Funktion für aufeinander folgende Integrationsbuilds (Continuous Integration). Bei aufeinander folgenden Integrationsbuilds müssen Codedefekte bereits vor dem Einchecken ermittelt werden, da Unterbrechungen während des Buildprozesses bedeuten würden, dass der im Versionskontrollsystem gespeicherte Code bereits fehlerhaft ist. Die Behebung derartiger Probleme bedeutet Mehraufwand und kostet Zeit. Bei Einsatz von abgegrenzten Eincheckbuilds werden jedoch der Buildprozess sowie alle erforderlichen Tests bereits vor dem Einchecken des Codes ausgeführt. Scheitert der Buildprozess oder schlägt ein Test fehl, wird der Code nicht eingecheckt. Auf diese Weise wird die Arbeit der anderen Entwickler im Team nicht beeinträchtigt, während die für den fehlerhaften Code verantwortlichen Entwickler die erforderlichen Korrekturen vornehmen. Abgegrenzte Eincheckbuilds stützen sich auf eines der Markenzeichen von TFS, dem „Shelf Set“-Feature. Sollte ein abgegrenzter Eincheckbuild fehlschlagen, kann der mit der Problembehebung

betraute Entwickler andere Entwickler hinzuziehen, ohne dass der fehlerhafte Code eingecheckt werden muss.

WIRD VON DEN TESTERN ÜBERPRÜFT...

Tester zählen traditionell nicht direkt zum Entwicklungsteam, da das Testen bisher meist erst nach dem Abschluss der Entwicklungsarbeiten durchgeführt wurde. Aufgrund der zu Beginn erwähnten Statistiken kommen Organisationen nun jedoch zu der Erkenntnis, dass die Tester ein wichtiger Bestandteil des Entwicklungsteams sind. Visual Studio 2010 bezieht die Tester auf jeden Fall direkt in die zentrale Projektarbeit ein, beispielsweise mit Microsoft Test Manager (siehe Abbildung 6) und anderen professionellen Tools für die Testverwaltung und -ausführung.

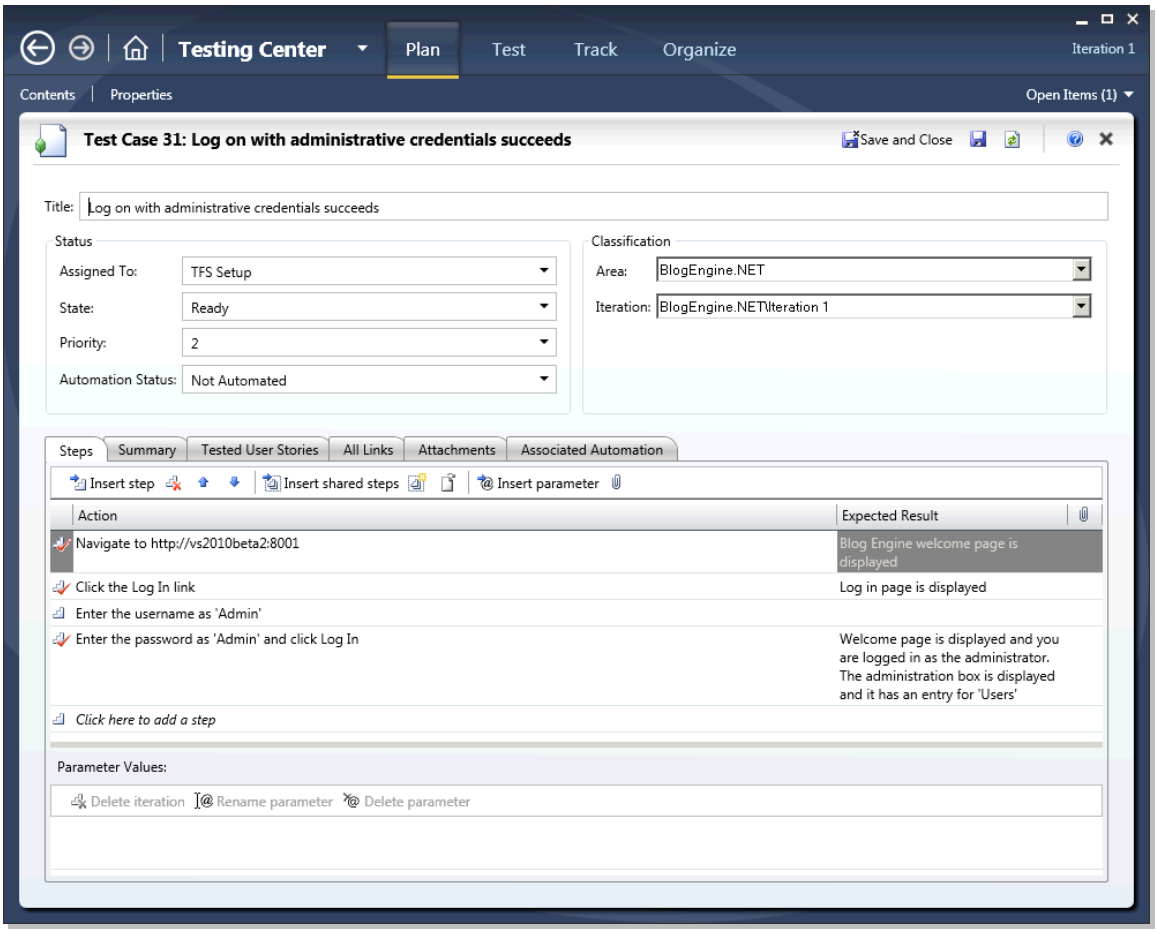
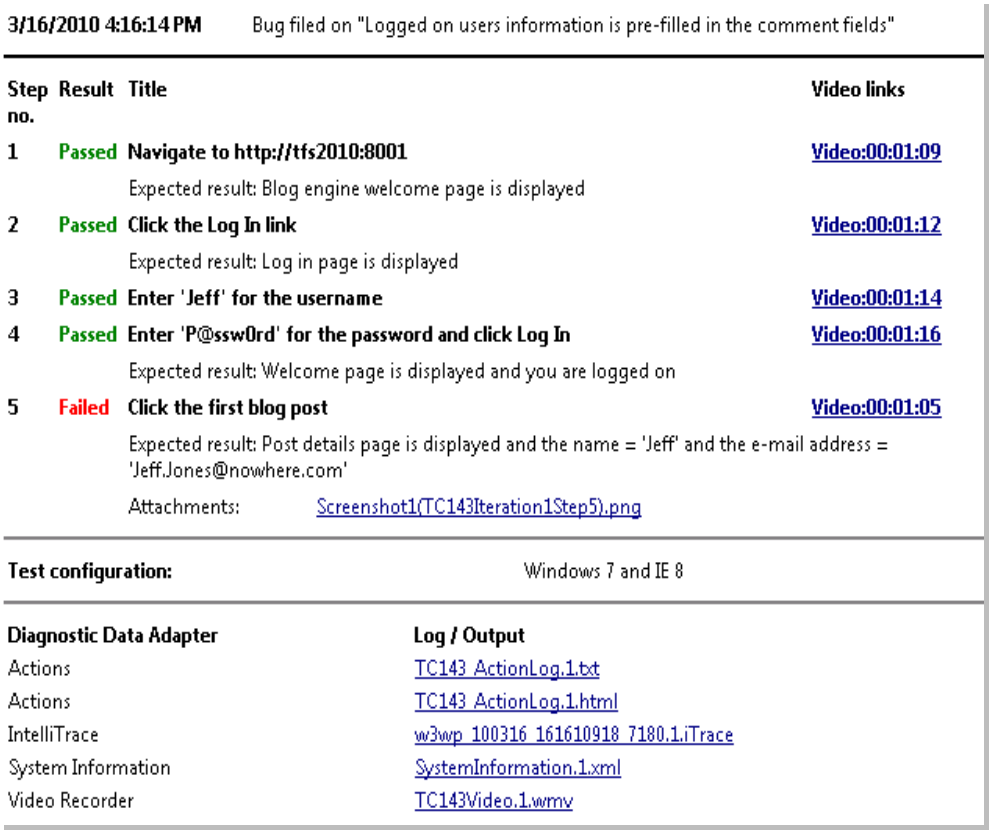


Abbildung 6: Microsoft Test Manager.

Zu Beginn eines Projekts kann das QA-Team die Testfälle direkt den jeweiligen Anforderungen zuordnen, damit die Entwickler über die auszuführenden Testfälle informiert sind. Anschließend kann der Code mit diesen Testfällen getestet werden, um zu bestimmen, ob der Code den Kundenanforderungen gerecht wird. Die Qualität einer Anwendung lässt sich mit zahlreichen Testtypen sicherstellen. Manuelle oder allgemeine Tests machen ca. 70 Prozent aller Tests aus, die

von einem Entwicklungsteam ausgeführt werden sollten. Es sind jedoch auch andere Testtypen verfügbar, die sich von den üblichen Komponenten- und Leistungstests bis hin zu Auslastungs- und automatisierten Tests erstrecken. Tester können detaillierte Diagnoseinformationen erfassen und selbst Videos der Testsitzungen erstellen, um ermittelte Probleme nachvollziehbar und tatsächlich behebbar an die Entwickler zu kommunizieren.

In Abbildung 7 ist ein mit Microsoft Test Manager protokollierter Testabschnitt dargestellt, der ein ermitteltes Problem reproduzierbar beschreibt. Alle Schritte sind detailliert aufgeführt, einschließlich der erfolgreichen und fehlgeschlagenen Schritte. Außerdem sind Anlagen bzw. Kommentare für alle Schritte verfügbar. Die Zeitindizierung in den Videos ermöglicht es den Entwicklern, genau an die Stelle zu springen, an der das Problem aufgetreten ist. Außerdem ist das IntelliTrace™-Protokoll einbezogen, welches detaillierte Debuginformationen für die problematische Testsitzung umfasst.



3/16/2010 4:16:14 PM Bug filed on "Logged on users information is pre-filled in the comment fields"

Step no.	Result	Title	Video links
1	Passed	**Navigate to http://tfs2010:8001**	Video:00:01:09
		Expected result: Blog engine welcome page is displayed	
2	Passed	**Click the Log In link**	Video:00:01:12
		Expected result: Log in page is displayed	
3	Passed	**Enter 'Jeff' for the username**	Video:00:01:14
4	Passed	**Enter 'P@ssw0rd' for the password and click Log In**	Video:00:01:16
		Expected result: Welcome page is displayed and you are logged on	
5	Failed	**Click the first blog post**	Video:00:01:05
		Expected result: Post details page is displayed and the name = 'Jeff' and the e-mail address = 'Jeff.Jones@nowhere.com'	
		Attachments: Screenshot1(TC143Iteration1Step5).png	

Test configuration: Windows 7 and IE 8

Diagnostic Data Adapter	Log / Output
Actions	TC143_ActionLog.1.txt
Actions	TC143_ActionLog.1.html
IntelliTrace	w3wp_100316_161610918_7180.1.iTrace
System Information	SystemInformation.1.xml
Video Recorder	TC143Video.1.wmv

Abbildung 7: Schritte für die Problemreproduzierung

Die TIA-Analyse ist ein neues Feature, das den Testern alle erforderlichen Informationen vermittelt, um basierend auf den am Code vorgenommenen Änderungen diejenigen Tests mit der höchsten Priorität auszuwählen. Wurde bereits erfolgreich getesteter Code geändert, werden die Tester gewarnt und können die entsprechenden Tests erneut ausführen, um Regressionsprobleme zu reduzieren bzw. auszuschließen. Lab Management geht noch einen Schritt weiter, indem die

Testinfrastruktur virtualisiert wird. Virtuelle Umgebungen machen es unter anderem möglich, beim Auftreten eines Problems einen Snapshot der virtuellen Umgebung zu erstellen. Dieser Snapshot kann dann an die Entwickler weitergegeben werden, um das Problem genau in der Umgebung zu reproduzieren, in der es aufgetreten ist.

... UND SICHTBAR FÜR DAS MANAGEMENT

Visual Studio 2010 und TFS 2010 sind von Grund auf auf die Gewährleistung der Qualität ausgerichtet, allerdings sollte das Management auch über den Projektstatus informiert werden. Immerhin müssen die an den Projekten beteiligten Manager Informationen über den Status und die Qualität der Software an die Kunden kommunizieren. Wird beispielsweise erst in der letzten Minute ein Problem gefunden, ist der Kurs meist nur noch schwer zu ändern. TFS umfasst jedoch detaillierte Berichte auf der Grundlage von SQL Server Analysis Services. Zusammen mit Microsoft Office SharePoint Server und Excel Services sind diese Informationen (nicht nur die Rohdaten) jederzeit verfügbar, aktuell und genau. In Abbildung 8 ist eines der verfügbaren Dashboards mit Qualitätsinformationen dargestellt.

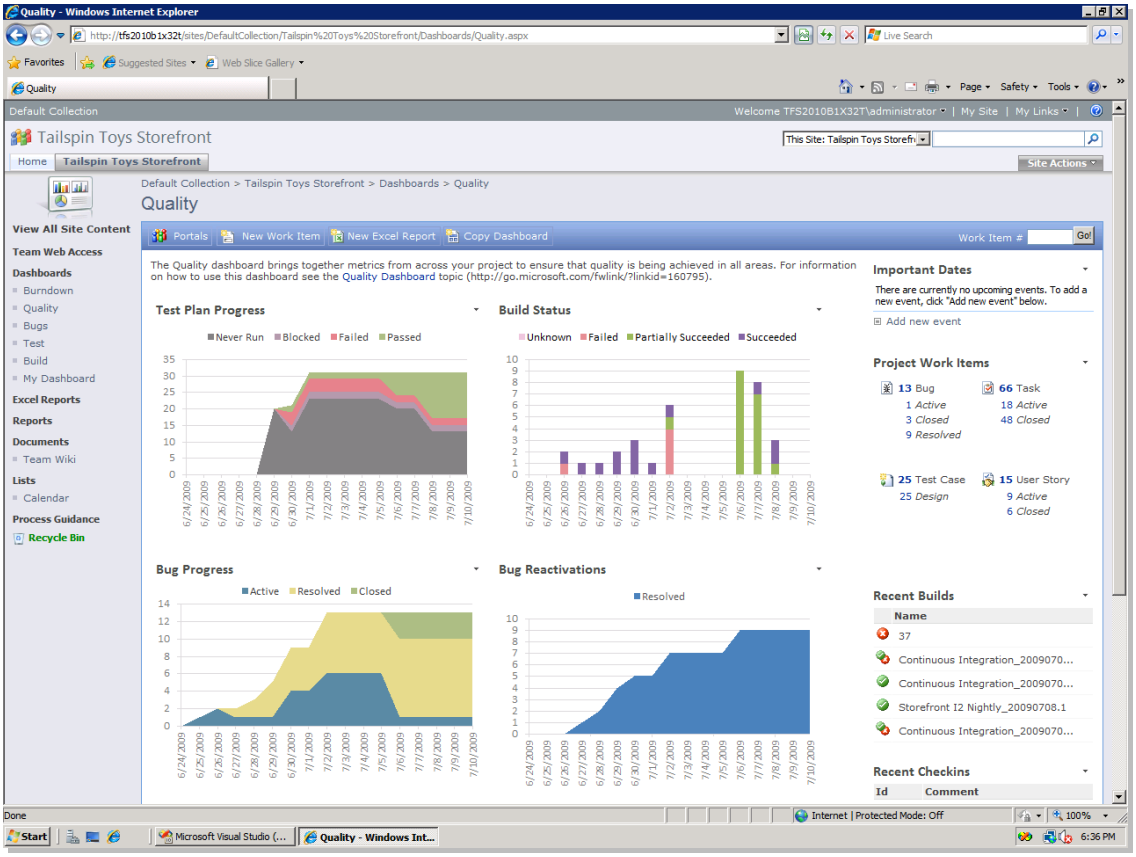


Abbildung 8: Ein Qualitätsbericht

Der Testplanstatus zeigt an, wie viele Tests erstellt, ausgeführt und erfolgreich abgeschlossen wurden. Der Buildstatus weist außerdem die Anzahl der automatisierten Builds aus sowie die

erfolgreichen und fehlgeschlagenen Builds zusammen mit den Ergebnissen der automatisierten Tests, die als Bestandteil des Buildprozesses ausgeführt wurden. Diese Informationen ermöglichen eine tägliche Bestandsaufnahme bezüglich der Qualität Ihrer Anwendungen. Problemstatus und Problemreaktivierung bieten detaillierte Informationen über den Status aller für eine Anwendung protokollierten Probleme sowie den aufgrund von nicht korrekt behobenen Problemen entstehenden Arbeitsaufwand.

Diese Informationen ermöglichen es dem Management, erforderliche Kurskorrekturen rechtzeitig vorzunehmen. Werden viele Probleme protokolliert, können entsprechende Maßnahmen ergriffen werden, um die Problemanzahl zu reduzieren. Sollten einige Tests nicht ausgeführt werden, können zusätzliche Tester eingesetzt werden, um sicherzustellen, dass ein qualitativ hochwertiges Produkt rechtzeitig und zur Zufriedenheit des Kunden ausgeliefert wird.

DIE QUALITÄT IST VON BEDEUTUNG

Alle Entwicklungsteams verfolgen in ihren Projekten das gleiche Ziel: Eine qualitativ hochwertige Anwendung zu entwickeln. Dieses Ziel ist ohne die richtigen Tools jedoch nur schwer zu erreichen. Von der Anwendungsarchitektur über die Entwicklung bis hin zum Testen und zur Wartung bietet Visual Studio 2010 die geeigneten Tools, um die Anwendungsqualität sicherstellen. Unabhängig davon, welche Rolle Sie im Entwicklungsprozess spielen, können Sie mit diesen Tools die Anforderungen Ihrer Kunden erfüllen.

WAS ANDERE SAGEN

ICONICS

Globaler Softwareentwickler erwartet aufgrund einer Entwicklungslösung die Verdoppelung der Produktivität

ICONICS, ein Microsoft Gold Certified Partner, bietet industrielle Automatisierungs- und Visualisierungssoftware für Unternehmen in mehr als 60 Ländern an. Die Entwicklungsteams des Unternehmens umfassen Mitarbeiter aus drei Ländern. Die geografische Entfernung stellte eine Herausforderung für die Zusammenarbeit dar. Die Entwickler verfügten bisher nicht über die erforderlichen Tools, um das Projektmanagement zu unterstützen, den Datenzugriff zu ermöglichen, die Entwicklung zu beschleunigen und die Testprozesse zu automatisieren. Im Jahre 2009 erweiterte ICONICS seine Entwicklungsumgebung mit Microsoft Visual Studio 2010 Ultimate, Visual Studio Team Foundation Server 2010 und Visual Studio Lab Management 2010. Diese vollständig integrierte Lösung stellt einheitliche Tools für globale Teams bereit, um das Projektmanagement zu vereinfachen und die Lebenszyklusprozesse der Produkte zu rationalisieren. Das Ergebnis ist, dass ICONICS die Kosten reduzieren und die Produktqualität verbessern kann. Außerdem wird ein Anstieg der Produktivität um 100 Prozent erwartet.

„Die neuen Testfunktionen in Visual Studio 2010 sind revolutionär. Wenn immer mehr Teams diese Tools verwenden, wird sich unsere Produktivität verdoppeln. Wir können automatisierte Tests über Nacht ausführen, ohne dass jemand anwesend sein muss. Wir erwarten, dass Features wie IntelliTrace die zum Debuggen von Anwendungen erforderliche Zeitdauer künftig wesentlich reduzieren werden.“ - Chris Elsbree, Leitender Softwarearchitekt, ICONICS

„Die Vorteile von Visual Studio 2010 basieren auf Leistung und Qualität. Features wie die hierarchische Arbeitsaufgabenverfolgung, die Verzweigungsvisualisierung (Branch Visualization) und IntelliTrace sowie die Möglichkeit mit dem neuen System ältere Versionen zu erstellen, tragen zu einer Steigerung unserer Effizienz bei. Dies führt letztendlich zu einer besseren Produktqualität.“ - Russ Agrusa, Präsident und CEO, ICONICS