

Testen von Softwaresystemen

WS 2004/2005

Test Management Tools

Bernthaler Christian
Schönbeck Andreas
Stadlhuber Gregor
Strecker Gert

Inhaltsangabe

1. TESTMANAGEMENT	3
1.1. Testfälle (Test Cases)	3
1.2. Test Packages	4
1.3. Test Suites	4
1.4. Test Cycles	5
2. EINSATZFELD	6
3. ERWARTUNGEN AN EIN TEST MANAGEMENT TOOL	7
4. MARKT EINBLICK	9
4.1. AP Test Manager	9
4.2. Test Manager 2.0	11
4.3. T-Plan Professional	12
4.4. SilkCentral Test Manager	13
5. RECHTFERTIGUNG FÜR DIE VERWENDUNG EINES COMPUTERGESTÜTZTEN TESTMANAGEMENTS	14
5.1. Vorteile	14
5.2. Nachteile	16
5.3. Umfang (Enterprise Size)	16
5.4. Conclusio	17

1. Testmanagement

Um einen Einblick für den Nutzen von einem computergestützten Testmanagement zu erhalten, sei an dieser Stelle der zugrundeliegende Aufbau von „Testing“ erklärt.

Wie bei einer gut durchdachten Software-Architektur, ist eine genaue Strukturierung der Testumgebung notwendig, um Überblick und eine deutliche Trennung von verschiedenen Testbereichen zu erreichen, und somit ein gut funktionierendes System zu erhalten.

Dazu ist es notwendig, die einzelnen Testfälle bestimmten Packages und Test Suites zuzuordnen, um sie dann in Test Cycles testen zu können. Diese Konzepte werden nachfolgend erläutert:

1.1. Testfälle (Test Cases)

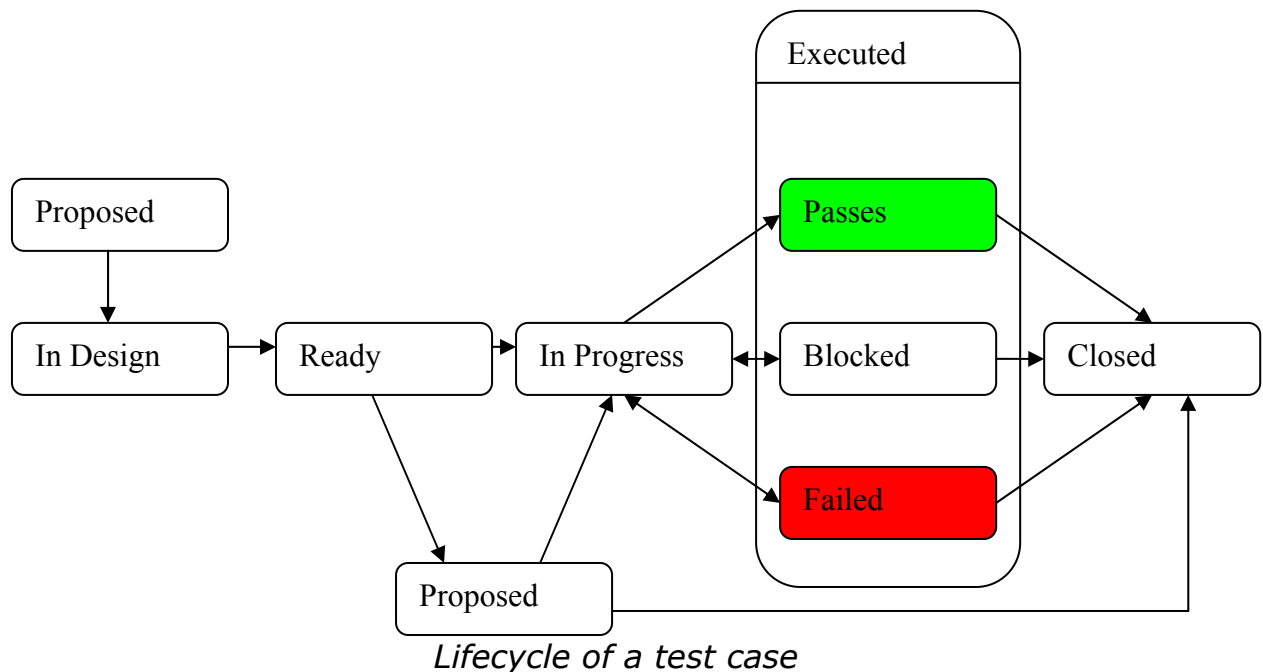
Definiert die kleinste Einheit beim Testmanagement.

Über einen Testfall müssen folgende Informationen zur Verfügung stehen:

- Identifikationsnummer: muss eindeutig sein
- Name: „sprechender“ Name, der den Testfall bezeichnet
- Ziel: es darf nur ein gut definiertes Ziel pro Testfall vorhanden sein
- Vorbedingungen: hier wird angegeben, unter welchen Bedingungen der Test ablaufen muss. (z.B.: Systemkonfiguration, Datenbankinitialisierung, etc.)
- Ablauf der Testschritte: in welcher Reihenfolge werden welche Funktionen gestartet
- Zusätzliche Attribute, wie z.B. Status, Priorität, usw.

Die einzelnen Testfälle werden aufgrund der großen Anzahl in Packages, nach Attributen (z.B. nach Priorität) oder in Suites verwaltet.

Life-cycle eines Testfalles:



1.2. Test Packages

Mit Hilfe von Packages kann eine hierarchische Struktur von Testfällen, die miteinander in Beziehung stehen aufgebaut werden. So können zum Beispiel Testfälle, welche dieselbe Funktion testen zu einer Gruppe zusammengefasst werden.

Es sollte darauf geachtet werden, dass für alle Testfälle eines Packages die gleichen Vorbedingungen gelten. Somit könnte auch eine Hierarchie aufgrund der Vorbedingungen erstellt werden.

Grundsätzlich sollten Testfälle gut strukturiert in geeignete Packages zusammengefasst werden, um eine spätere Lokalisation einzelner Testfälle zu erleichtern. Auch können in Folge gesamte Packages zur Ausführung kommen, um z.B. eine bestimmte Funktion zu testen. Alle Testfälle die eben diese Funktion betreffen müssen dann nicht mehr einzeln ausgewählt, sondern können über die hierarchische Struktur zur Ausführung gebracht werden.

1.3. Test Suites

Als Test Suites bezeichnet man Gruppen von Testfällen die zur Ausführung zusammengestellt werden. Zu Test Suites können einzelne Testfälle, aber auch Packages hinzugefügt werden.

Testcases werden aus verschiedenen Gründen zu Test Suites zusammengestellt: Um Tests mit denselben Einstellungen laufen zu lassen, oder um alle „high priority“ Tests auszuführen. Weiters können Test Suites für neue Funktionalitäten oder Bugfixes verwendet werden. Auch muss die Möglichkeit gegeben sein Arbeit auf mehrere Tester aufteilen zu können.

Da bei vielen Projekten nicht unbegrenzt Zeit zur Verfügung steht, um alle Tests auszuführen, sollen die wichtigsten Tests zuerst gestartet werden. Damit kann man kritische Fehler und Probleme frühzeitig erkennen und eventuell beheben, bevor die Testphase abgeschlossen ist. Um diese Probleme managen zu können werden Prioritäten vergeben:

- Anforderungen: die wichtigsten Anforderungen sollen zuerst getestet werden
- Häufigkeit der Verwendung: die wichtigsten und am häufigsten ausgeführten Funktionen sollten zuerst getestet werden.
- Kritische Systemteile: Systemnotwendige Teile zuerst testen
- Wahrscheinlichkeit von Fehlern: Teile mit hoher Fehlerwahrscheinlichkeit zuerst testen (z.B.: Teile mit hoher Komplexität, mit neuer Technologie implementiert,...)

1.4. Test Cycles

Testabläufe sind sehr wichtig im Testmanagement. Hier passiert die Abgleichung zwischen Entwicklung und Testen.

Ein Testablauf kann in folgenden Schritten beschrieben werden:

- Produkt wird erhalten: sichergehen, dass die richtige Version erhalten wurde
- Testsystem konfigurieren: Mit einem sauberen System starten
- Testbarkeit verifizieren: ist die Version stabil genug um getestet zu werden
- Feststellen was neu ist, oder was sich geändert hat
- Feststellen welche Fehler korrigiert wurden
- Alte Tests, die bereits kontrolliert wurden wiederholen
- Neue oder geänderte Bereiche testen
- Andere Bereiche testen
- Die Ergebnisse dokumentieren

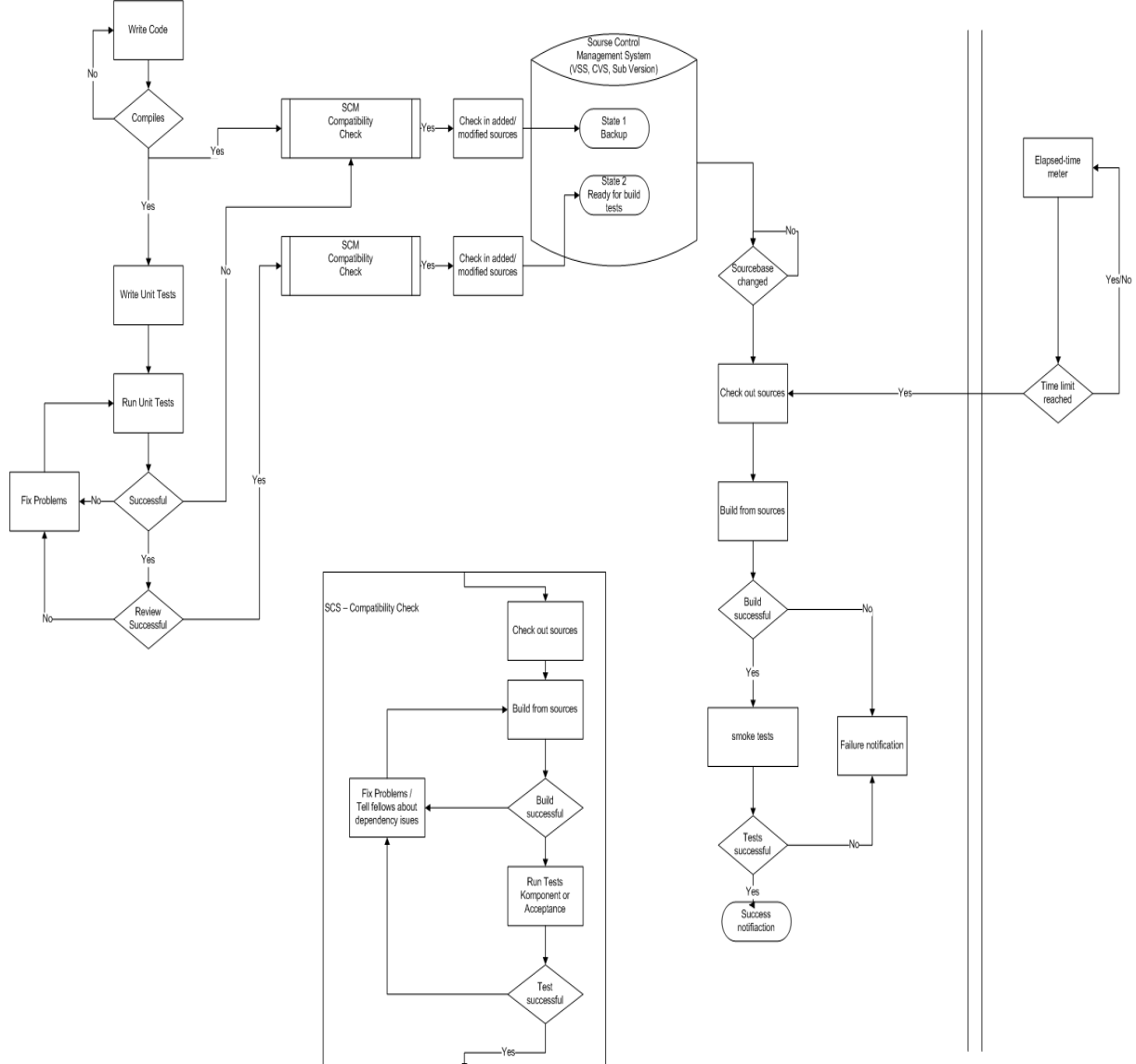
2. Einsatzfeld

Ein wichtiger Bestandteil bei der Entwicklung einer Software stellt das Testen des Erstellten Codes dar. Ohne Testing ist es heute unverantwortlich Softwarelösungen in Betrieb gehen zu lassen.

Bei diesem Prozess sollten sowohl Entwickler, (Test-) Manager als auch Kunden beteiligt sein. Letztere kann man im Bereich von Test Management vorerst vernachlässigen. Diese Stakeholder treten erst bei Acceptance-Tests und bei Bug-Tracking verstärkt in den Vordergrund.

Der Entwickler erschafft das Material unter Rücksichtnahme von Testing-Methoden, welche der Test Manager zu verwalten und analysieren hat um schließlich Feedback geben zu können.

Aus der Sicht eines Entwicklers, wäre ein möglicher Entwicklungsablauf wie folgt dargestellt.



Die Definition der durchzuführenden Tests, die Erstellung und Ausführung, und die Bereitstellung der Ergebnisse fallen in die Verantwortung des Test Managers und ist somit Einsatzgebiet für ein computerunterstütztes Testmanagement.

3. Erwartungen an ein Test Management Tool

Um sich in einem Betrieb für den Einsatz eines Test Management Tools zu entscheiden, sollte man auf gewisse Kernkriterien acht geben, welche den betrieblichen Anforderungen wie Mitarbeiteranzahl und Projektumfang entsprechen sollen, sodass man auf diese Weise ein gutes Preis-Leistungsverhältnis erzielen kann.

Ein Test Management Tool soll den Test Manager hinsichtlich seiner Aufgaben mit nützlichen Informationen und statistischen Auswertungen vom Festlegen der Tests bis hin zum Ausführen und Auswerten selbiger unterstützen. Darunter fallen folgende zu unterstützenden Aufgaben:

- Übersicht über alle TestCases
 - Welche / wieviele sind durchgeführt worden?
 - Welche / wieviele waren erfolgreich?
 - Welche / wieviele hatten Fehler?
 - Zeitverbrauch
 - Wurden Tests ausgelassen?
- Testabdeckung
 - des gesamten Systems
 - modulspezifisch
 - sind Kernfunktionalitäten ausreichend abgedeckt?
- Fehlerkorrelation
 - In welchen Modulen gibt es wie viele Fehler?
 - In welchen Modulen gibt es welche Fehler?
 - Welcher Entwickler macht wie viele Fehler?
 - Welcher Entwickler macht welche Fehler?
- Test-Trajektorie
 - Archivierung und Bereitstellung von Ergebnissen absolvierter Testläufe
- Dokumentation

Sowohl die einzelnen Testfälle also auch die definierten Testläufe müssen durch eine computergestütztes Test Management übersichtlich verwaltet

werden. Es soll die Möglichkeit bestehen, einzelne Testfälle schnell auszuwählen und in bestimmte Testabläufe zu integrieren.

Somit schlägt ein Test Management Tool die Brücke zwischen dem tatsächlichen Ausführen von Code, also dem Testing, und der zur Verfügungstellung der dadurch entstehenden Daten.

Test Management Werkzeuge lösen daher nicht die Probleme in bestehendem Programmcode, sondern stellen Informationen, aus denen Lösungsansätze abgeleitet werden können, zur Verfügung.

4. Markt Einblick

Im Allgemeinen ist vorweg zu nehmen, dass sich eine Evaluierung der Test Management Tools als kein leichtes Unterfangen entpuppt hat, da der Großteil der Software nicht als Shareware, oder zum Download, verfügbar ist.

Dies kann die Wahl des adäquaten Testmanagement Tools verfälschen. Somit werden die Bedürfnisse des computerunterstützten Testmanagement eventuell nicht erfüllt.

Ebenso hat sich gezeigt, dass es gerade bei Anwendungen die in Java programmiert sind, teilweise zu massiven Installationsproblemen kommen kann.

4.1. AP Test Manager

AP Test Manager ist ein Test Management Tool welches die Tests, eines Software oder Hardware Projekts, sowie deren Ergebnisse, aufzeichnet und überwacht.

Letztere sind nach den Tests für alle, die sie brauchen, über einen Webserver zugänglich.

Die Hauptfeatures von AP Test Manager sind:

- webbasierte Testdefinition, -ausführung und Ergebnisbekanntgabe
- zentrale Verwaltung der Tests und deren Ergebnisse
- Planung von Überwachung und Bekanntgabe der Soll- und Ist-Zeiten
- Automatischer Vergleich der alten mit den neuen Ergebnissen
- einfache Einbindung in bereits existierende Projekte und Methoden
- höchst konfigurierbar und flexibel
- Hierarchische Test Suite Struktur
- Tests können gruppiert werden (siehe Kapitel 1)
- Ausführen von ausgewählten Testgruppen
- Kompatibel zu bestehenden Sourcecodeverwaltung- und Bugtracking Tools
- Testcase Im- und Export

Dieses Produkt stellt ein Server- / Client Anwendung dar, die auf jedem beliebigen Webserver installiert werden kann, und mit den gängigen Browsern zugänglich ist.

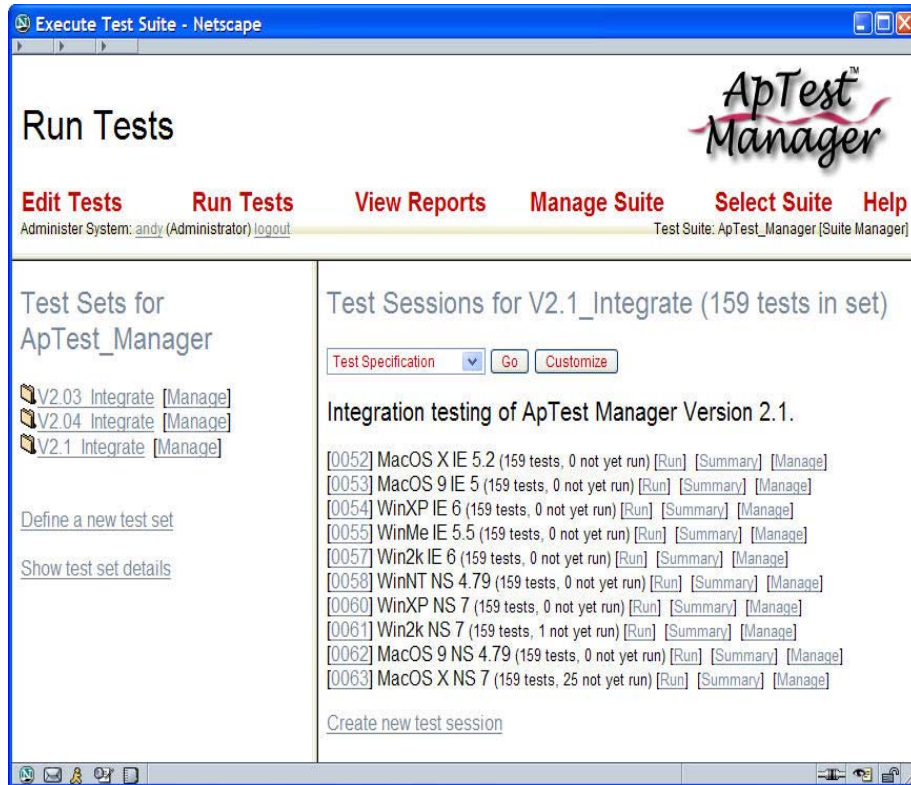
Weiters können die Testcases in verschiedenste Formate exportiert - (z.B. CVS für Excel Spreadsheets, usw.) - und Testergebnisse von anderen Softwareprodukten importiert werden.

Testen von Softwaresystemen

Test Management Tools

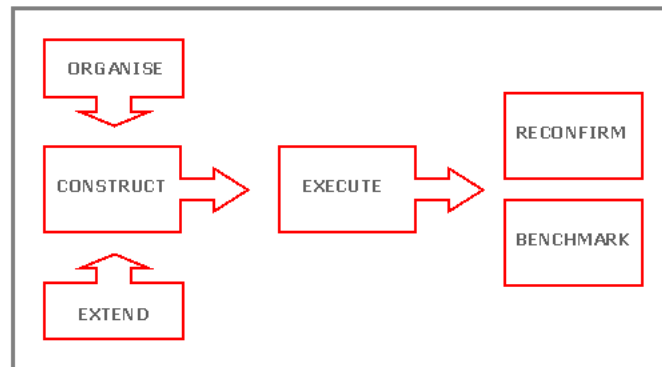
Der Hersteller verspricht ein stabiles Laufzeitverhalten des AP Test Managers bei einer Anzahl von bis zu 30000 Tests und 100 Benutzern.

Der Kostenaufwand für die Anwendung beläuft sich auf 500 USD pro Arbeitsplatz bzw. für die Serverversion auf 10.000 USD. Für kleine Firmen kann diese Ausgabe ein massives Problem darstellen, da der Betrieb dieses Produktes erst mit Server einen Einsatz wirklich sinnvoll erscheinen lässt.



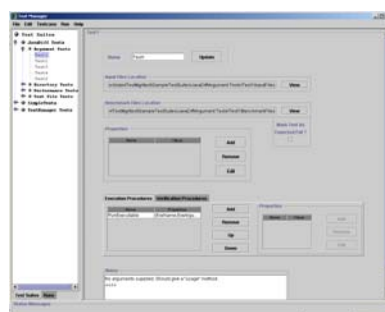
4.2. Test Manager 2.0

Test Manager ist eine automatisierte Testsoftware die in Java programmiert ist. Sie bedient sich einer komponentenbasierten Technik um Testdurchläufe zu verwalten, erleichtern und zu automatisieren.



Das Programm implementiert folgende 6 Features:

- Definition von einem oder mehreren Testcases d.h. spezifizieren der Testcase-Eingabedaten sowie anlegen und verifizieren von Ausführungsmethoden aufgrund von bereitgestellten Komponenten
- Ausführung der vorher definierten Testcases
- Ein Benchmark wertet die Ergebnisse aus und prüft diese auf Korrektheit
- Nachkontrolle: Nachdem das System aufgrund der vorherigen Testergebnisse angepasst wurde, wird der Test erneut durchlaufen und danach die Ergebnisse verglichen.
- Organisieren der Testcases in Gruppen um z.B. häufig verwendete Eingabedaten und Testeigenschaften innerhalb der selben Testgruppe teilen zu können
- Erweitern der bereitgestellten Testkomponenten mit eigenem Code



4.3. T-Plan Professional

T-Plan Professional ist eine weitere Testmanagement Software die im Zentrum des Softwareentwicklungszyklus.

Es ist in 3 Module aufgeteilt:

Modul 1: Analyse (was ist zu testen)

Hier geht es um die Findung der Anforderungen ab denen das Testen startet. Das zu Testende System wird in funktionale Bereiche unterteilt, die eine detaillierte Messung der Testbedingungen zulassen. Weiters können den Testbedingungen Prioritäten zuteilt werden die eine gezielte Analyse zulassen.

Modul 2: Design (wie ist zu testen)

Es kann eine Testspezifikationshierarchie zur Modellierung des Testplans erstellt werden. Innerhalb dieser Hierarchie werden dann die „Testscripte“ erstellt. Jedes Testscript kann mit den Testergebnissen verknüpft werden.

Modul 3: Manage (wann ist zu testen)

In diesem Modul kann der Ausführungszeitpunkt des Testscripts geplant werden. Weiters werden in diesem Modul die Testergebnisse graphisch und statistisch, nach den Gesichtspunkten:

- was war geplant zu testen?
- was wurde versucht zu testen (Testscripts)?
- was wurde terminlich geplant zu testen?
- was wurde wirklich getestet? , usw.

ausgewertet.

Die Hauptfeatures sind

- Referenzierung von externen Dokumenten
- Anforderungsdatenbank
- Flexibilität durch die Möglichkeit neue Attribute, Icons, Entities und Relations hinzuzufügen
- einheitliches Testframework
- Wiederverwendbarkeit von Tests

Die Vorteile dieses Produkts liegen

- in der guten und leichten Kommunikation zwischen den Projektmitarbeitern.
- Ein durchgängige Methodik für alle Testprozesse
- Import und Export der Ergebnisdaten von/in Word oder Excel
- Offene Architektur die Integrationen und Erweiterungen zulässt

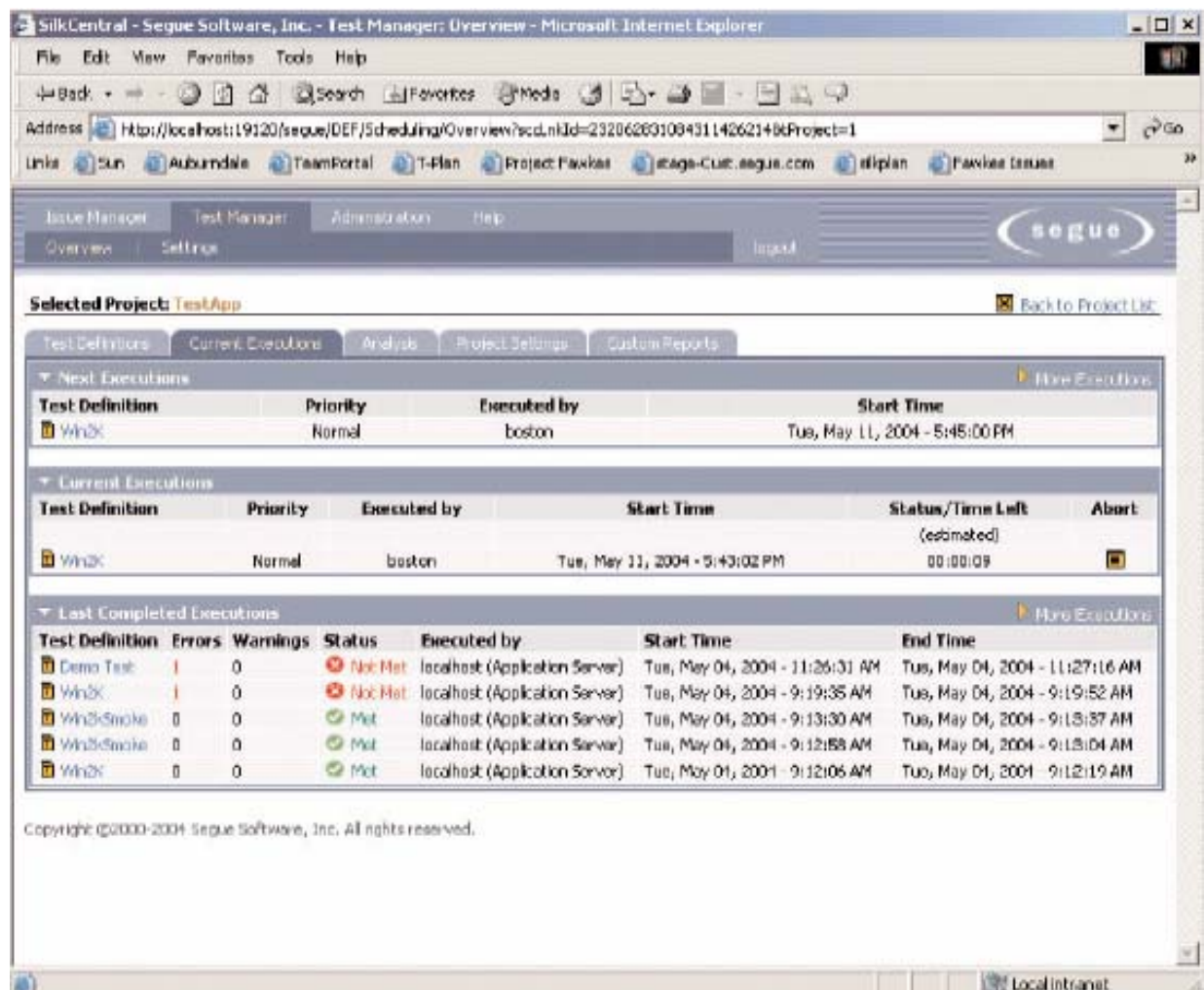
4.4. SilkCentral Test Manager

Der SilkCentral Test Manager ist eine webbasierte Lösung die das ganze Spektrum von Endanwenderanforderungen über Planung, Ausführung bis hin zur Analyse von Tests anbietet.

Auch hier bildet ein zentraler Speicher den Grundstock des gesamten Testprozesses. Ein Webinterface bietet die Möglichkeit immer und überall auf die Testdaten zuzugreifen.

Die Hauptfunktionen sind:

- Organisation der Anforderungen und Spezifikationsdokumente in Word, die dann mit den Testplänen, Testfehlern verknüpft werden können um die Nachvollziehbarkeit zu erhöhen
- Anlegen von Testplänen für manuelle und automatische Test über das Webinterface
- Anlegen und Ausführen von Projekt- und Regressionstests mittels des Webinterfaces
- Webbasierte Auswertung der Tests



5.Rechtfertigung für die Verwendung eines computergestützten Testmanagements

In diesem Kapitel werden die Vor- und Nachteile von Testmanagement Tools abgewogen. Es wird hier versucht einen Rat zu geben, wo man Testmanagement Tools einsetzen kann, und wo es keinen Sinn macht solche zu verwenden. Es kann hier aber keine Lösung angeboten werden, da die Entscheidung über den Einsatz von Testmanagement Tools immer der Situation entsprechend getroffen werden muss.

Abschließend wird dann auch noch eine kurze Zusammenfassung des ganzen Themas angeführt:

5.1. Vorteile

Allgemein lässt sich sagen, dass manuelles Testmanagement schnell zu einer unüberschaubaren Aufgabe wird. Testmanagement Tools unterstützen beziehungsweise automatisieren den Vorgang des manuellen Testmanagements mit ihren Funktionen.

- Testmanagement Tools lassen es zu den „Überblick“ zu bewahren, da praktisch alle Tools auf dem Markt logische Teststrukturen erstellen können mittels den folgenden schon im Eingangskapitel erklärten Konstrukten (Die einzelnen Hersteller verwenden zumeist andere Begriffe für diese Konstrukte, die aber alle denselben Zweck erfüllen):
 - Testcases
 - Testpackages
 - Testsuites
 - Testcycles
- Testprioritäten: Viele „große“ Projekte haben nicht genug Zeit um immer alle Tests durchlaufen zu lassen. Eine Auswahl welche Tests höhere Prioritäten haben wird nötig, um zumindest diese ausführen zu können. Testmanagement Tools unterstützen dieses Konzept zumeist. (vgl. Kapitel 1 und 4)
- Es gibt in jedem Projekt viele kleine Änderungen und spezielle Details die recht leicht übersehen werden, aber sich auch in den Tests widerspiegeln müssen. Testmanagement Tools ermöglichen es hier die Kontrolle zu behalten.

- Testmanagement Tools haben eine einheitliche Form Reporte über Testergebnisse, Stati, etc. zu erstellen, die zusätzliche Informationen abbilden können wie zum Beispiel:
 - Wie oft wurde der Test ausgeführt?
 - Wie ist die Tendenz der Fehler?
 - Wie ist die Produktivität des Programms
 - Wie weit wird ein Produkt mit Tests abgedeckt?
 - ...

Diese sind dann also auch in verschiedensten Projekten wieder gleich. Das hat eine einmalige Einarbeitungszeit zur Folge. Außerdem können Testmanagement Tools diese Reporte auch meist über einen Webserver bekanntmachen, sodass sie für alle Stakeholder zugänglich sind. Sprich für die Entwickler genauso wie für das Management und die Kunden. Letzteres ist ein besonders wichtiger Punkt, da das Vertrauen des Kunden in das Produkt dadurch gestärkt werden kann.

- Informationen bleiben leichter erhalten bei Personaländerungen, verglichen mit manuellem Testmanagement. Bei manuellem Testmanagement bleiben unter Umständen viele Informationen auf der Strecke, da diese nicht einheitlich oder überhaupt nicht dokumentiert sind.
- Testmanagement Tools unterstützen ein zentrales Testrepository. Alle Tests sind damit global gespeichert und können von dort abgerufen und verwendet werden.
- Die oben genannten Punkte vereinheitlichen das Testmanagement für die einzelnen Projekte. Ist der Umgang mit dem Testmanagement Tool einmal geläufig ist der Einsatz bei weiteren Projekten höchst effektiv.
- Testmanagement Tools können zumeist leicht mit verschiedensten Technologien, die zum Testen nahezu unumgänglich sind, verwendet werden. Nur um ein Beispiel zu nennen wird hier CVS und JUnit angeführt.

5.2. Nachteile

- „Mit Kanonen auf Spatzen schießen“: Der Umfang eines Projekts erfordert nicht immer gleich ein Testmanagement Tool. Bleiben die Tests in einem gewissen Rahmen wird der Einsatz eines Testmanagement Tools vielleicht unnötig sein. Allerdings ist die Grenze ab der sich der Einsatz eines solchen Tools rentiert recht niedrig anzusetzen.
- Ein weiterer Nachteil von Testmanagement Tools ist, dass sie das Management nur unterstützen aber nicht erledigen. Sie helfen gewisse typische Fragen beim Testmanagement zu beantworten (siehe weiter oben); die Antwort selbst gibt das Testmanagement Tool aber nicht. Die Arbeit des Testmanagements bleibt also bestehen.
- Die professionellen Testmanagement Tools sind relativ kostspielig, wenn man von den Open Source Lösungen absieht.
- Wie bei jedem Einsatz neuer Technologien ist auch hier vor dem Verwenden eine sorgfältige Evaluierung des Tools notwendig, falls keine Erfahrungswerte vorhanden sind. Der dadurch entstandene Arbeitsaufwand muss dabei in Relation mit dem Gewinn durch das Tool stehen, der sich allerdings schnell einstellen sollte, sofern das richtige Tool gewählt wurde.

5.3. Umfang (Enterprise Size)

Es lässt sich hier sehr schwer eine Aussage treffen, ab welcher Firmengröße ein Testmanagement Tool notwendig und sinnvoll ist. Unter Umständen ist ein Testmanagement Tool sogar für einen Ein-Mann-Betrieb rentabel. Man muss aber bedenken, dass Testmanagement Tools im Allgemeinen nicht unter einer freien Softwarelizenz oder einer Open Source Lizenz stehen, und daher nicht kostenfrei sind.

Man muss hier also einen Kompromiss finden um zu entscheiden, ob sich für eine Firma ein Testmanagement Tool amortisiert. Zieht man hier Open Source Software in Betracht muss natürlich berücksichtigt werden, dass diese Programme keinerlei Funktionalität oder Qualität garantieren.

Hat eine Firma aber schon mehrere Entwickler beschäftigt, die am selben Projekt arbeiten, soll hier schon ein Testmanagement Tool nahe gelegt werden.

Am anderen Ende der Skala, also bei sehr großen Projekten soll hier auf die Spezifikationen der einzelnen Tools verwiesen werden. Nur als Beispiel: Der AP Test Manager verspricht ein einwandfreies Funktionieren bei einer hohen Anzahl gleichzeitigen Tests und Benutzern. Wenige Firmen

werden mehr als in diesem Versprechen angegebenen Zahlen überschreiten. Hier und dort wird das aber passieren, und das ist dann der Punkt an dem Testmanagement Tools an ihre Grenzen stoßen.

5.4. Conclusio

Abschließend bleibt zu sagen, dass Testmanagement Tools leicht unterschätzt werden. Es wird sogar das Testing an sich schon unterschätzt.

Softwareentwicklung ist, soviel ist sicher, ohne Testing nicht vorstellbar. Sobald getestet wird, muss aber auch ein gewisses Ausmaß an Testmanagement betrieben werden. Was würde ein Testlauf bringen, ohne daraus die nötigen Schlüsse zu ziehen? Und was würden diese Schlüsse bringen, wenn sie nie an die Entwickler herangetragen würden?

Testmanagement tritt also auf die eine oder andere Weise in jedem Entwicklungsprozess zu Tage. Wird der Aufwand für manuelles Testmanagement zu groß wird ein Testmanagement Tool fast unumgänglich.

Somit zeigt sich, dass der zunächst unscheinbar wirkende Part eines Testmanagement Tools eine massive Rolle spielt. Einem guten Softwareentwickler sollte das bewusst sein.