

Automatisiertes Testen - Grundlagen

||> Status quo

Jede Softwareentwicklung durchläuft während ihrer Entstehung und nach Fertigstellung eines Release vielfältige Testroutinen und Prozeduren und doch hat jede ausgelieferte Software Fehler. Test-Ingenieure versuchen alles menschenmögliche, diese Fehler vor Auslieferung aufzuspüren, dennoch bleiben Fehler und Fehlverhalten immer wieder unentdeckt, die selbst mit optimierten manuellen Testverfahren nicht zu finden sind. Automatisiertes Testen ist die beste Möglichkeit, die Effektivität, Effizienz und Gesamtabdeckung zu steigern und zu optimieren.

||> manuelle Testverfahren

Manuelles Testen wird von Menschen vollzogen, die sich Schritt für Schritt durch Benutzeroberflächen kämpfen, mit dem Bestreben durch vielfältige Nutzung und Eingabe-Variationen/Kombinationen vergleichbare Resultate hinsichtlich des Verhaltens der getesteten Software zu erlangen. Manuell erstellte, in Umfang und Qualität individuell unterschiedliche, Dokumentation begleitet diesen Software-Test-Prozess.

Tests werden oftmals innerhalb der Entwicklungs-Zyklen wiederholt, auch oft aufgrund von Code-Änderungen oder anderer Variablen wie Betriebssystem-Varianten und unterschiedliche Hardware-Plattformen.

||> automatisiertes Testen

Automatisiertes Testen bietet hingegen die Möglichkeit, voraufgezeichnete und vordefinierte Aktionen wieder und wieder zu durchlaufen, Resultate der Testläufe objektiv und direkt zu vergleichen und den verantwortlichen Testingenieuren automatisch generierte Berichte über Erfolg oder Misserfolg der Testroutinen zu liefern. Sind automatisierte Testverfahren erstmalig definiert und erstellt, so können diese jederzeit wiederholt und darüberhinaus dazu benutzt werden, um Aufgaben zu erfüllen, die sowohl qualitativ als auch quantitativ bei manuellen Testverfahren unmöglich wären.

Automatisiertes Testen - der Nutzen

||> Nutzen

Automatisiertes Testen als essentielle Komponente erfolgreicher Software-Entwicklungs-Projekte

Automatisiertes Testen ist lange Zeit einerseits als geschäftskritisch in Hinsicht auf große Software-Projekte und Organisationen betrachtet worden, auf der anderen Seite als zu kostenintensiv oder sogar zu aufwändig, um in kleinen Unternehmen eingeführt werden zu können.

Mit unseren eingesetzten Tools können wir jedoch sowohl die Bedürfnisse von KMU-Betrieben in gleicher Weise wie die Anforderungen von Großbetrieben abdecken. Weltweit agierende Konzerne wie IBM, HP, Intel, Autodesk, Intuit, Macromedia, Motorola, Symantec und Sony setzen die selben Tools für automatisiertes Testen ein.

||> Kosteneffizienz

Automatisiertes Testen verringert Kosten - Zeit und Geld

Tests müssen innerhalb der Software-Entwicklungs-Zyklen oftmals wiederholt werden. Jedes mal, wenn Source-Code sich verändert, ist ein neuer Testlauf nötig. Jedes Software-Release muss auf allen unterstützten Betriebssystemvarianten und allen Hardware-Plattformen getestet werden. Manuelle Wiederholungen sind sowohl kosten- als auch zeitintensiv.

Dieses rekursive Testen von vorgefertigten automatisierten Testroutinen hingegen verursacht keinerlei Zusatzkosten (sieht man von Rechnerleistung ab) und kann dazu viel schneller als manuelles Testen durchgeführt werden. Zeitliche Aufwände von Tagen können auf Stunden reduziert werden. Eine Zeitersparnis, die sich direkt auf eine Kostenersparnis übertragen lässt.

||> Genauigkeit

Automatisiertes Testen verbessert die Genauigkeit und verringert die Fehleranzahl

Selbst der genaueste und pflichtbewussteste Tester wird im Zuge von monotonem, wiederholendem manuellen Testen nicht fehlerfrei arbeiten können. Automatisierte Tests führen ein und die selben Schritte beliebig oft in gleicher Präzision durch und dokumentieren diese in detaillierten Test-Berichten.

Automatisiertes Testen - Teamplayer

||> die Abdeckung

Automatisiertes Testen steigert die Gesamtabdeckung von Testverfahren

Automatisiertes Testen steigert den Umfang und die Detailtiefe von Tests, was zu verbesserter Software-Qualität führt. Langzeit-Tests, die oftmals innerhalb manueller Testverfahren vermieden werden, können unbeaufsichtigt parallel zum Entwicklungsprozess vollzogen werden.

Diese können gleichzeitig auf mehreren unterschiedlichen Hardwareplattformen und Betriebssystemen erfolgen. Automatisiertes Testen kann z.B. zur Tiefen-Analyse in Bezug auf Speicher-Ressourcen, Datenbank-Tabellen, Datei-Inhalte, interne Programm-Stati etc. benutzt werden, um das Programmverhalten genau zu bestimmen. Automatisierte Test-Tools können zur Abhandlung von Tausenden verschiedener komplexer Test-Fälle innerhalb jedes Testlaufes genutzt werden, was manuell unmöglich wäre.

||> die Einsatzgebiete

Automatisiertes Testen unterstützt sowohl Entwickler als auch Tester

Automatisiertes Testen kann Entwicklern Fehler vor der Auslieferung an die QA-Abteilung aufzeigen. Tests können jedesmal, wenn Source-Code verändert und z.B. in ein Versionskontrollsystem eingchecked wurde, automatisch ablaufen und das Entwicklungsteam und/oder den speziellen Entwickler benachrichtigen, wenn diese fehlerhaft sind. Dies spart Entwicklungszeit und steigert das Vertrauen in die Arbeit.

||> das Team

Automatisiertes Testen fördert teamorientiertes Arbeiten

Durch die Einsparung von wiederholten manuellen Testverfahren wird Entwicklungsteams die Möglichkeit gegeben, sich mehr auf herausfordernde, interessante Aufgaben zu konzentrieren. Teammitglieder verbessern ihre Fähigkeiten und das Vertrauen in ihre geleistete Arbeit steigt, was zu einem messbar gesteigerten Rückfluss an Produktivität führen kann.

Automatisiertes Testen - Funktionen

||> die Tools

Die von uns eingesetzten Software-Tools bilden ein mächtiges und kostengünstiges Programmpaket, das allen Anforderungen von IT-Abteilungen, Produktingenieuren, QA-Verantwortlichen und IT-Beratern gerecht wird. Dies ermöglicht die Verbesserung von Software-Test-Verfahren mittels Erhöhung von Effizienz, Vermeidung von nicht gewünschter Komplexität in Testverfahren und gleichzeitiger Verminderung von laufenden Kosten.

||> die Funktionen

Ein Auszug aus der Funktionspalette

- Last, Stress und Skalierungstest von Web Servern
- Distributed Testing (Netzwerk-relevante Testroutinen)
- Objektorientiertes Testen
- Direktes Komprimieren und Archivieren von Testergebnissen
- Funktionsaufrufe aus .net-Bibliotheken und dll's
- Testen von Corba-Objekten
- Vielfältiger Compiler-Support (Visual Basic, Visual C++, Delphi, C++Builder, Java, Visual Studio .NET, Visual FoxPro)
- Web Page Testing
- Unterstützung von sprachunabhängigem Scripting (VBScript, DelphiScript, JScript, C++Script, C#Script ...)
- Erstellungs von Testscripts durch Aufzeichnung und/oder direktes Kodieren
- Sammeln von Testroutinen in Test-Paketen