



Bild: Je weiter fortgeschritten das Projekt, desto teurer wird die Fehlersuche.

Tools sind kein HR-Ersatz

Kein Tool entbindet den Tester jedoch davon, sinnvolle und qualitätssichernde Tests zu erstellen. In der Praxis wird oft sehr schnell die Forderung laut, alle möglichen Ablaufpfade mögen doch bitte automatisch getestet werden. Den Sinn oder vielmehr Unsinn dieser Anforderung möchte ich an einem einfachen Beispiel erläutern. Ein Programm kann bei Windows üblicherweise auf drei Arten beendet werden: erstens mit der Schaltfläche „X“ im Fenster oben rechts; zweitens mit dem Menüpunkt „Datei/Beenden“; drittens mit der Tastenkombination „Alt-F4“. Hinzu kommt die Frage „Speichern: Ja, Nein, Abbrechen?“, wenn nichtgespeicherte Änderungen existieren. Das sind insgesamt zwölf mögliche Ablaufvarianten, die sich aus sieben Einzelaktionen kombinieren lassen. Dabei sind effektiv nur zwei Funktionsgruppen zu testen (Aktion an der Oberfläche und auszuführende Operation „Beenden“).

Sicher, wenn die Tests einmal programmiert sind, dann kostet die Durchführung „nichts“. Doch bei einem Fehler bekomme ich dann drei Fehlermeldungen mit derselben Ursache. Und alles muß dokumentiert werden, ohne einen zusätzlichen Informationsgewinn. In der Praxis erscheint es sinnvoll, mit Hilfe von Oberflächentests lediglich die Funktionen der Bedienoberfläche zu testen. Diese Tests prüfen nur, ob die Aktionen an der Oberfläche mit den eingegebenen Werten die richtigen

Funktionen anstoßen. Das ähnelt dem Testablauf der Unit-Tests. Eine der Bedienungsanleitung gemäße Programmarchitektur mit Präsentations- und Logikschicht sollte diese Trennung der Tests unterstützen.

Ob sich – zusätzlich oder alternativ – der Aufwand lohnt, automatische UI-Tests für das Programm als Ganzes (automatischer Black-Box-Test) zu erstellen, sollte im Einzelfall entschieden werden. Dafür spricht unter anderem: ein absehbarer Produkt-Lebenszyklus mit häufigen Nachbesserungen und einer „stabilen“ Benutzeroberfläche, die weiterentwickelt, aber nicht ständig verändert wird; eine klar strukturierte



„Für die Entscheidung
„automatisch oder manuell“
gilt die einfache Regel:
so viel automatisch
wie möglich,
so wenig manuell
wie nötig“.

Oberfläche mit dem Schwerpunkt auf Texteingaben; wenig „Mustererkennung“ zur Identifizierung von Schaltflächen, Farben, Pop-Ups und so weiter; ein „fertiges“ Programm, in dem die Programmfehler der Alpha- und Beta-versionen behoben sind.

Abwägungsszenarien

Sehr aufwendig wird die Erstellung automatischer Test bei einer Oberfläche, bei der „Drag & Drop“, das Halten und Ziehen der Maus und andere grafische Elemente den Großteil der Bedienung ausmachen. Das bedeutet nicht, daß es sich nicht lohnen kann. Dies sollte jedoch sorgfältig kalkuliert werden. Als Faustregel kann dabei gelten, daß der Aufwand für die Testprogrammierung etwa genauso groß ist wie der Aufwand für die Programmierung der Oberfläche.

Unerlässlich ist eine Testautomatisierung bei agilen Entwicklungsmethoden wie zum Beispiel „Scrum“: Ohne eine automatische Testdurchführung sind die nach jedem „Sprint“ vorzunehmenden Tests der gesamten Software nicht effizient durchführbar. Umgekehrt waren natürlich die automatischen Testmethoden eine Grundvoraussetzung dafür, daß sich die Methodik der Softwareerstellung überhaupt in diese Richtung entwickelt hat. Die automatische Testdurchführung erlaubt und gewährleistet die zuverlässige Ausführung aller Tests.

Bei dem hohen Maß an Testabdeckung sollte aber immer im Hinterkopf bleiben, daß auch die besten Tests nur das prüfen, woran die Entwickler gedacht haben. Es bleibt immer eine Grauzone an Ausnahmefehlern, die nicht oder nur mit einem unverhältnismäßig hohen Aufwand getestet werden kann. Eins sollte jedoch klar sein: Die zahlreichen Varianten zum Testen von Software sind ähnlich vielfältig wie die Softwareerstellung selbst. Die berühmte eierlegende Wollmilchsau gibt es nicht. Und ich kann mir nicht vorstellen, daß es sie dereinst geben wird. Vielmehr sollten technische Neuerungen und Möglichkeiten immer unter Berücksichtigung der bisherigen Erfahrungen betrachtet werden, um neuerstellte Software mit dem richtigen Maß zu testen.

DIPL.-INF. VOLKER BRASE

WEB-TIPP:
www.infolab.de