

FEHLER LIEBEN LERNEN



Es gibt keine fehlerfreie Software. Das ist wahrscheinlich eine der schmerzhaftesten Erkenntnisse für einen Studenten der Informatik, wenn er die Welt mit seinen Programmierkünsten erobern will.

Dass die Betriebssysteme von PCs und Smartphones so ihre Tücken haben, hat man im Laufe der Jahre zu akzeptieren gelernt. Ein Neustart bewirkt in der Regel ja Wunder. Aber wie können dann Weltraumprogramme, Kernkraftwerke oder medizinische Geräte guten Gewissens unter immer stärkere Kontrolle komplexer Softwaresysteme gegeben werden? Die alten Softwarehasen zucken da nur mit den Schultern und verweisen auf die immensen Anstrengungen für Test und Qualitätssicherung in jedem größeren Softwareprojekt.

In vielen Anwendungsbereichen sieht die Realität aber anders aus: Test und Qualitätssicherung werden in der Regel zwar umfangreich geplant und beabsichtigt, aber in vielen Entwicklungsabteilungen und Softwarehäusern wird dann doch beim nahendem Auslieferungstermin im Zweifel nicht der Termin verschoben, sondern auf ein paar Tests verzichtet. Was noch fataler ist: In Erwartung lästiger umfangreicher Tests lässt so mancher Softwareentwickler schon mal Fünfe gerade sein und liefert sein Paket ohne große eigene Überprüfungen dem Testteam aus, denn diese prüfen es ja sowieso noch Herz und Nieren.

Zu allem Überfluss soll Software zunehmend kontinuierlich ausgeliefert werden, um Benutzerbedürfnisse schnell befriedigen zu können. Wie soll man da noch systematisch testen? Die Frage stellt berechtigterweise **Mark Paluch** in seinem Artikel in dieser Ausgabe von OBJEKTSpektrum. Die Antwort auf die Frage heißt oft „Testautomatisierung“, aber häufig genießen Testautomatisierungen heute einen schlechten Ruf. Der Aufwand, die Testautomaten an die Softwareänderungen anzupassen, ist oft viel zu groß und damit wird das Testen zum größten Flaschenhals im gesamten Entwicklungsprozess. Das erhöht den Zeitaufwand für Auslieferungen immens und ist zudem teuer.

Wie in so vielen Bereichen ist die agile Entwicklung auch im Testbereich ein Motor zur Veränderung. Mit den Beiträgen von **Ramon Anger** und **Frederik Eichler** sowie dem von **Kerstin Dittert** haben wir gleich zwei Artikel in diesem Heft über den Zusammenhang von Testautomatisierung und agiler Softwareentwicklung. Kerstin Dittert schreibt: „Kent Becks Vision vom ‚den Test liebenden Programmierer‘ ist vielerorts wahr geworden.“ Das ist aus meiner Sicht vielleicht etwas zu pathetisch, aber es hat einen wahren Kern: Die entscheidende Veränderung bei agilen Projekten ist, dass Qualität nicht nachträglich in die Software hineingetestet, sondern durch geeignete Praktiken und Werkzeuge schon früh in die Software hineingearbeitet wird.

Der Aufwand für die Erstellung und Pflege der Testautomaten soll durch die Modellierung von Tests und durch die Generierung von Testfällen aus Modellen reduziert werden. **Anne Kramer** und **Martin Beissner** beschreiben diesen Ansatz, der aber natürlich relativ hohe Initialaufwände verursacht. **David Faragó** und seine Kollegen haben deswegen untersucht, wann sich die initiale Investitionen in modellbasiertes Testen trotzdem lohnen.

Ricardo Wickel stellt in seinem Beitrag eine generelle Investitionsstrategie für die Testautomatisierung vor. Er betont dabei, dass die besten Investitionen im Bereich Unit-Test möglich sind, und verweist dabei auf dieselbe Testautomatisierungspyramide wie **Markus Gärtner**, der aber eben auch betont, dass nicht alle Tests sinnvoll zu automatisieren sind.

Trotzdem ist bei all diesem Hype meiner Ansicht nach guter Rat teuer, wie man Unit-Tests praktisch umsetzt. Daher finde ich die „12 Tipps für guten Unit-Test-Code“ von **Sven Thiergen** auch so lesenswert, wenn es dem Leser darum geht, durch handwerklich saubere Arbeit Qualität zu erzeugen.

Qualität ist auch in der Redaktion unser Anspruch. Deswegen gehen die eingereichten Artikel durch einen mehrstufigen, aber nicht automatisierbaren Review-Prozess. Eine zentrale Rolle spielt dabei unser Redaktionsteam, das Ausgabe für Ausgabe jeden eingereichten Artikel unter die Lupe nimmt.

Über viele Jahre waren **Matthias Bohlen** und **Torsten Weber** wertvolle Stützen in diesem Team, die mir meinen Einstieg in die redaktionelle Verantwortung mit ihrem großen Sachverstand erleichtert haben. Beide haben sich jetzt aber entschlossen, sich erst einmal nicht mehr die Nächte mit OBJEKTSpektrum-Artikeln zu versüßen. Ich kann das verstehen, bedaure aber ihr Ausscheiden sehr.

Als Nachfolger haben wir glücklicherweise zwei neue Kollegen gewinnen können: **Hans-Peter Korn** und **Gerhard Wanner** verstärken ab dieser Ausgabe unser Redaktionsteam. Gerhard Wanner hat langjährige praktische Erfahrung in der professionellen Softwareentwicklung und arbeitet heute als Professor an der Hochschule für Technik Stuttgart. Hans-Peter Korn ist ein sehr erfahrener Praktiker und Berater im Bereich Software-Engineering und selbst als Autor aktiv. Beide schätze ich persönlich sehr und freue mich schon auf die kommenden kontroversen Debatten.

Ihr Thorsten Janning