

## Regressionstests für LV-Bestandsführungssysteme

# Wege zur umfassenden Testautomatisierung

Bestandsführungssysteme gehören in der heutigen IT-Landschaft der Lebensversicherungsunternehmen zu den strategischen Kernsystemen. Häufige Anpassungen und Erweiterungen sowie die steigende Komplexität der eingesetzten Bestandsführungs-Software stellen eine große Herausforderung für die Qualitätssicherung dar, die sich langfristig nur durch eine umfassende Automatisierung von Software-Tests bewältigen lässt.

Dr. Arno Rasch ist Software-Engineer und Testexperte bei vtmw.

Bild: vtmw



Peter Kernwein ist aktuarieller Berater bei vtmw.

Bild: vtmw



Dirk Möller ist Senior-Fachberater Test bei vtmw.

Bild: vtmw



Die Anforderungen an die Bestandsführung in der Lebensversicherung befinden sich in einem ständigen Wandel, der sich in den letzten Jahren weiter beschleunigt hat. Wichtige Beispiele hierfür sind die durch den verschärften Wettbewerb und die Entwicklungen auf den Kapitalmärkten erzwungene Einführung neuer Produkte, Veränderungen der rechtlichen Rahmenbedingungen (wie VVG-Reform und Solvency II) sowie die Integration migrierter Bestände. Durch diesen Wandel ergibt sich die Notwendigkeit ständiger Anpassung und Erweiterung der bestehenden LV-Bestandsführungssysteme, was seinen Niederschlag in der häufigen Einführung neuer Software-Versionen findet. Mit den kürzer werdenden Entwicklungszyklen und zugleich wachsender Komplexität steigt naturgemäß der entsprechende Testbedarf. Dabei wird nicht nur neue oder erweiterte Funktionalität einer neuen Software-Version gegen entsprechende Anforderungen getestet, sondern es muss zugleich sichergestellt werden, dass durch die Weiterentwicklung der Software ihre bisherige

Funktionalität nicht beeinträchtigt wird. Letzteres ist die Kernaufgabe des Regressionstests.

### Regressionstest

Der Regressionstest soll gewährleisten, dass nach einer Modifikation der Software keine zusätzlichen Fehler in bereits getestete Komponenten hinzugekommen sind. Hierzu wird der Inhalt von Softwareobjekten wie Tabellen, Schnittstellen, GUI oder Dateien, die während der Testausführung erzeugt bzw. modifiziert werden, mit dem Inhalt der entsprechenden Objekte einer früheren Software-Version, den Referenzwerten, verglichen. Jeder Regressionstestfall muss zuvor also mindestens einmal erfolgreich durchgeführt worden sein.

Für den Testbetrieb bedeutet dies im Allgemeinen, dass bereits erfolgreich absolvierte Testfälle bei jeder Aktualisierung der Software erneut durchgeführt und die erzeugten Ergebnisse auf Abweichungen gegenüber den Referenzwerten geprüft werden müssen. Bei jeder Abweichung ist zu entscheiden, ob diese gewünscht bzw. im Rahmen der

Rechengenauigkeit toleriert wird oder ob sie durch einen Fehler verursacht wurde. Im ersten Fall sind die Referenzdaten zu aktualisieren, während im zweiten Fall der projektübliche Fehlerbehandlungsprozess angestoßen wird. Zusätzlich zur Überprüfung der Korrektheit der ermittelten Abweichungen sollte geprüft werden, ob – etwa nach einer Fehlerkorrektur – tatsächlich alle beabsichtigten Änderungen eingetreten sind (Überprüfung auf Vollständigkeit).

In der Praxis wird der Regressionstest teilweise noch manuell durchgeführt, was aufgrund der im Ver-

### Literatur

- [1] Liggesmeyer, P.: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, Heidelberg/Berlin, 2002.
- [2] Fewster, M. /Graham D.: Software Test Automation – Effective use of test execution tools, ACM Press, New York, 1999.
- [3] Langer, M. / Krammer, S.: Auswertung funktionaler Softwaretests: Eine Lücke in Theorie und Praxis, in: Objektspektrum 3/2005, S.68-71.

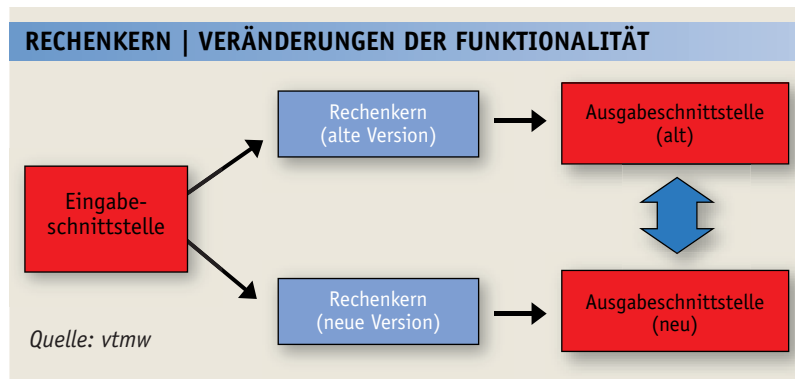
hältnis zum Ersttest üblicherweise geringen Fehlerrate unter wirtschaftlichen Gesichtspunkten nicht sinnvoll ist [1]. Stattdessen empfiehlt es sich, durch Einsatz von Automatisierungswerkzeugen mögliche Fehler bei der manuellen Eingabe der Testdaten zu vermeiden und zugleich die Testdurchführung zu beschleunigen. Der anschließende Vergleich mit den Referenzdaten kann ebenfalls maschinell erfolgen. Die Palette der hierfür eingesetzten Werkzeuge reicht von einfachem Dateivergleich bis hin zu flexiblen und konfigurierbaren Lösungen zur Analyse von Testergebnissen in diversen strukturierten Formaten [2,3]. Die Testauswertung hingegen – also die Entscheidung, ob Abweichungen erwünscht sind oder nicht – lässt sich im Allgemeinen nicht gut automatisieren. Für diese Aufgabe ist insbesondere im versicherungstechnischen Bereich ein fundiertes aktuarielles Know-how unverzichtbar.

### Idealfall: Zentraler Testfallkatalog

Ein weiteres Problem in der Praxis ist ein oftmals unzureichender Regressionstestfallbestand. Während sich die zu testende Software ständig weiterentwickelt, werden die Regressionstestfälle nur selten aktualisiert, was zu mangelnder Testabdeckung oder unnötiger Redundanz führen kann. Idealerweise wird ein zentraler Testfallkatalog angelegt, der es den Fachtestern erlaubt, dort ihre Testfälle und die entsprechenden Referenzdaten in strukturierter und leicht lesbarer Form zu spezifizieren. Dies erleichtert die Klassifikation nach Fachthemen oder Testzielen und ermöglicht letztlich eine computergestützte Analyse des Testfallbestandes, etwa zur Bestimmung der Testabdeckung, sowie eine gezielte Selektion von Testfällen nach bestimmten Testkriterien.

### Test von Geschäftsvorfällen

Zur Erfassung von Geschäftsvorfällen, wie beispielsweise Policierung oder technische Änderung sind in LV-Bestandsführungssystemen üblicherweise entsprechende Eingaben über eine graphische Benutzeroberfläche (GUI) erforderlich. Für die Automatisierung solcher Eingaben gibt es im Wesent-

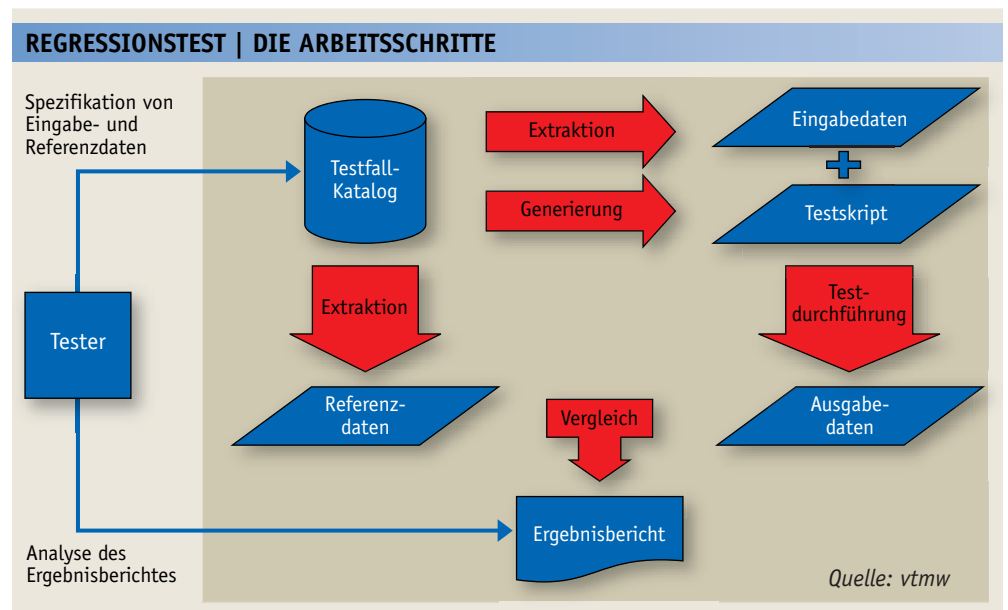


lichen zwei verschiedene Ansätze, die sich durch die Schlagworte „Capture-Replay“ und „Scripting“ charakterisieren lassen [2]. Beim Capture-Replay Verfahren werden manuell getätigte Benutzereingaben mit Hilfe eines Software-Werkzeugs aufgezeichnet und zur späteren Wiederholung gespeichert. Der Vorteil eines solchen Verfahrens liegt in der einfachen Bedienbarkeit, die Software-Tester in die Lage versetzt, ohne spezielles IT-Fachwissen Testfälle zu erstellen und anschließend automatisiert zu wiederholen. Klassische, positionsbasierte Capture-Replay Verfahren, welche die Benutzeraktionen lediglich in Form von Tastatureingaben, Mausebewegungen und zugehörigen Bildschirmkoordinaten aufzeichnen, sind jedoch sehr empfindlich gegenüber Änderungen des GUI-Layouts. Schon geringe Abweichungen der Position und Größe von GUI-Elementen oder eine geänderte Bildschirmauflösung können eine korrekte Wiederholung eines aufgezeichneten Testfalls verhindern. Capture-Replay Tools neuerer Ge-

neration umgehen zwar dieses Problem, indem sie direkt die Oberflächenelemente referenzieren, doch bieten sie kaum Möglichkeiten zur Strukturierung der Eingaben. Daher erweist sich dieser Ansatz bei häufiger Modifikation der Testfälle oder des GUI-Layouts in der Praxis als recht wartungsintensiv.

### Unabhängige Teilschritte auslagern

Ein alternativer Ansatz basiert auf der Programmierung von Testdurchläufen, wobei Abfolgen von Benutzereingaben mittels einer prozeduralen Skriptsprache formuliert werden, welche spezielle Anweisungen zur Bedienung von Standard-Oberflächenelementen und Eingaben in Textfelder bereitstellt. Unabhängige Teilschritte lassen sich dabei in Unterprozeduren auslagern und in verschiedenen Kontexten wiederverwenden. Dies hat außerdem den Vorteil, dass bei Änderungen an einer Eingabemaske nur die zuständige Unterprozedur angepasst werden muss. Da es sich



bei den Testskripten um Programme handelt, ist es naheliegend, Programmcode und Eingabedaten zu trennen, so dass das Testskript lediglich ein Gerüst darstellt – z.B. eine Navigation durch verschiedene Eingabemasken und Menüs – während die Daten, welche in Eingabefelder eingegeben werden, separat vorliegen, etwa in einer Datei. Auf diese Weise können Testfälle mit ähnlichem Ablauf, aber unterschiedlichen Eingabedaten durch ein einziges Testskript gesteuert werden. Der Nachteil des Scripting-Ansatzes ist die Tatsache, dass zu seiner Umsetzung in der Regel zusätzliches IT-Fachwissen, wie z.B. Programmierkenntnisse erforderlich sind.

### Robustes, flexibles Fundament

Andererseits bieten Techniken der Codegenerierung die Möglichkeit, solche Testskripte automatisch zu erzeugen und dadurch individuelle Fehler bei ihrer Erstellung auszuschließen. Aus der abstrakten Testfallspezifikation in dem Testfall-Katalog kann das konkrete Testskript automatisch generiert werden, ohne dass sich der Tester ständig mit den technischen Details der automatisierten Testdurchführung befassen muss. Die entsprechenden Testdaten werden ebenfalls aus der Testfallspezifikation extrahiert. Somit bietet dieser Ansatz ein robustes und zugleich flexibles Fundament zur umfassenden Automatisierung des Regressionstest, dessen

typische Arbeitsschritte in der Grafik „Regressionstest“ dargestellt sind. Mit Ausnahme der Spezifikation der Eingabe- und Referenzdaten durch den Tester sowie der Analyse der Abweichungen sind alle Schritte automatisiert.

### Mathematischer Test

In heutigen Bestandsführungssystemen ist die actuarielle Berechnungslogik typischerweise in einem mathematischen Subsystem oder Rechenkern gekapselt. Die Kommunikation mit dem eigentlichen Bestandsführungssystem erfolgt über eine definierte Schnittstelle, was die Abgrenzung des mathematischen Tests scheinbar sehr einfach macht. Daher beschränkt man sich häufig darauf, nur die an die Schnittstelle zur Bestandsführung gelieferten Daten zu testen. Für den Regressionstest lassen sich die zum Testfallbestand gehörenden Daten der Eingabeschnittstelle des Rechenkerns nutzen, um mit der neuen Software-Version die Ausgabeschnittstelle zu bestücken.

### Alles korrekt weiterverarbeitet?

Durch Vergleich dieser neuen Ergebnisdaten in der Ausgabeschnittstelle mit den unter Verwendung einer früheren Software-Version erzeugten Referenzdaten lassen sich Veränderungen in der Funktionalität des Rechenkerns leicht ermitteln (siehe Grafik „Rechenkern“ Seite 9). Jedoch wird bei diesem iso-

lierten mathematischen Test nicht überprüft, ob die im Rechenkern berechneten Werte von der Bestandsführung und den Randsystemen tatsächlich korrekt weiterverarbeitet werden. Ein weiteres Problem ist, dass die Eingabeschnittstelle mit der Zeit „veraltet“. Somit werden Veränderungen beim Aufruf des mathematischen Subsystems durch die Bestandsführung im Regressionstest nicht erkannt.

### Fazit und Ausblick

Gerade im Bereich der Bestandsführungssysteme ist aufgrund der komplexen Software-Architektur eine möglichst umfassende Automatisierung des Regressionstests wünschenswert und dank moderner Werkzeuge auch praktikabel. In der Praxis werden diese jedoch vielfach nur zögernd und punktuell eingesetzt.

Separate Regressionstests in einzelnen Software-Komponenten reichen allerdings nicht aus. Ein erster Schritt in Richtung eines umfassenden Regressionstests ist die Pflege eines gemeinsamen Testbestands für alle Komponenten, wodurch eine Durchgängigkeit des Tests der versicherungstechnischen Werte für den Gesamtprozess erreicht werden kann. Über weitere Schritte zur Testautomatisierung mehrerer Komponenten im Verbund sowie Möglichkeiten zur Auslagerung von Testaktivitäten werden die Autoren in einer der folgenden Ausgaben berichten.

Bild: pressmaster -  
Fotolia.com



### Fachgespräche 2009:

**Testautomatisierung und Regressions-tests für LV-Bestandsführungssysteme**  
Praxisbeispiele und Erfahrungsaustausch bezüglich Testdatenbeschaffung, Werkzeugeinsatz, Skriptgenerierung, Ergebnisaufbereitung, Bestandskorrekturen, Testauslagerung.

– jeweils 16:00 – 19:00 Uhr –

München, 15. Juli 2009  
Köln, 23. September 2009  
Stuttgart, 14. Oktober 2009  
Hamburg, 28. Oktober 2009  
Wiesbaden, 11. November 2009  
Zürich, 25. November 2009

Wir bitten um rechtzeitige Anmeldung, da nur begrenzte Plätze zur Verfügung stehen.  
vtmw GmbH, Seumestraße 3,  
D-81379 München  
Tel. +49 (0)89 78020895  
eMail: ruth.kilian@vtmw.de  
www.vtmw.de