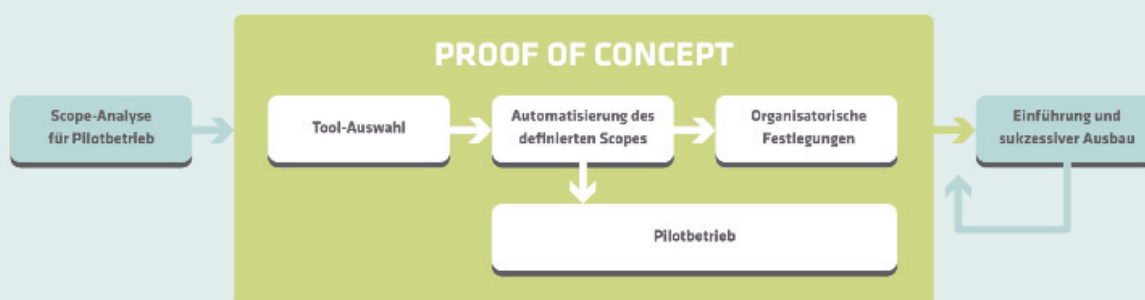


Testautomatisierung von Bankenanwendungen

Ein Ziel der Softwaretests ist, dass es gilt, Downtimes von wichtigen Anwendungen oder Plattformen unbedingt zu vermeiden.

Vorgehensmodell für die Einführung von Testautomatisierung



Tester-Ressourcen sind knapp und teuer. Insbesondere zum Ende des Testzyklus fehlt die Zeit, um ausreichend zu erproben und zu kontrollieren. Eine Folge kann sein, dass die Software in ungenügender Qualität in die Produktion geht. Es kommt zu Fehlern und Störungen im betrieblichen Ablauf. Noch schlimmer sind negative Außenwirkungen auf die Kunden durch falsche oder nicht rechtzeitig bereitgestellte Daten.

Von Dieter Koenen* und Marco Nowak**

Die Ursachen für die Diskrepanz zwischen den Testanforderungen und den tatsächlich durchgeführten Tests sind vielfältig:

- Der Markt fordert, neue oder veränderte Produkte und Prozesse kurzfristig zu platzieren.
- Die Verzahnung von Frontend- und Backend-Anwendungen erhöht die Komplexität der Software. Zusätzlich zu den Backend-Prozessen über SAP GUI gilt es, über Systemgrenzen hinweg Weboberflächen integriert zu testen.
- Das Angebot von Produkten und Services im Internet setzt neue Maßstäbe für die Sicherheit der Software. Der Umfang der Tests und die Anforderungen

an die Tester sowie die Testumgebung wachsen.

- Das agile Vorgehen verstärkt die Anforderungen an den Test: Das Ende eines Sprints erfordert zusätzlich zum Test der neu entwickelten Komponenten einen Regressionstest der kompletten Software aus vorherigen Sprints.
- Die Frequenzen für Release-Auslieferungen erhöhen sich. Dadurch verkürzen sich die Zeitfenster für das Testen dramatisch. Oder zu Ende gedacht: „Continuous Delivery“ fordert „Continuous Testing“.
- Die Personaldecke in den Fachbereichen dünnt sich aus verschiedensten Gründen immer weiter aus. Die Prozess- und Anwendungsspezialisten zerreißen sich zwischen der Mitwirkung in eng getakteten Projekten und dringend notwendigen fachlichen Tests.
- Die regulatorischen Anforderungen an die Entwicklung und den Betrieb und damit auch an den Test von Software verschärfen sich.
- Und da gibt es in der SAP-Welt noch die ungeliebten Release-Wechsel und Enhancement-Package-Upgrades ver-

bunden mit dem unvermeidbaren Regressionstest über alle Prozesse, Funktionen und Schnittstellen.

Dreistufige Vorgehensweise zur Automatisierung von Tests

Die Liste der möglichen Maßnahmen ist lang. Sie beinhaltet organisatorische und prozessuale Anpassungen im Testmanagement, Erweiterung der Tester-Ressourcen, Outsourcing von Tests, risikobasierte Testansätze und eben auch die Automatisierung von Tests. Aufgrund der Erfahrungen aus mehreren Testautomatisierungs-Projekten bei mittleren und großen „SAP for Banking“-Anwendern hat sich eine dreistufige Vorgehensweise bewährt.

Scope-Analyse

In der Scope-Analyse definieren Fachbereich und IT die Testszenarien oder Testfälle, die sich besonders gut für die Automatisierung eignen. Kriterien wie eine hohe Wiederverwendbarkeit oder eher seltene Änderungen der Oberflächen spielen für die Definition des Scopes eine entscheidende Rolle. ▶

*Dieter Koenen ist Manager des Teams Consulting & Application Services bei der innobis AG.

**Marco Nowak arbeitet als Senior Consultant im Team Consulting & Application Services bei der innobis AG.

Wenig überraschend stoßen die Beteiligten hier schnell auf Regressionstestfälle für SAP-Release-Wechsel oder regelmäßige Release-Auslieferungen für ausgewählte Prozesse wie die Partnerpflege oder die Vertragsverwaltung.

Proof-of-Concept

Der Proof-of-Concept startet mit der Auswahl eines geeigneten Testautomatisierungs-Werkzeugs. Zu den führenden Tools für Endanwender gehören Micro Focus Unified Functional Testing und Tosca von Tricentis. Beide Werkzeuge können die gängigen Oberflächen wie SAP GUI, SAP Fiori, Web Dynpro, Webbrowser aufzeichnen und wiedergeben. Sie arbeiten jedoch mit einem unterschiedlichen Ansatz:

- UFT von Micro Focus erstellt Visual-Basic-Scripts, die nachträglich editiert werden können. Die Anwendung erfordert ein Grundverständnis für IT und Visual-Basic-Scripting.
- Tosca von Tricentis arbeitet mit einem modellbasierten Ansatz und eignet sich auch für IT-affine Mitarbeiter aus dem Fachbereich.

Die Tosca-Testsuite beinhaltet im Gegensatz zu Micro Focus UFT ein Testfallverwaltungs-Modul, allerdings ohne ein Defect-Management wie im HP-Quality-Center. Beide Tools lassen sich in die SAP-Solution-Manager-Test-Workbench integrieren. Die Lizenzmodelle und Lizenzkosten sind unterschiedlich. Für Micro Focus UFT liegt die Einstiegsschwelle mit einer für den Start durchaus ausreichenden Concurrent-User-Lizenz niedrig. Im nächsten Schritt werden die fünf bis zehn in der Scope-Analyse definierten Testfälle mit dem ausgewählten Werkzeug automatisiert. In der Regel funktioniert das gut; allerdings gibt es Grenzen.

So eignet sich die Testautomatisierung weniger für Batchabläufe oder für Tests mit inhaltlichen Zwischenprüfungen von Ergebnissen. Andere Hürden wie die Abbildung des Vier-Augen-Prinzips oder die Reaktion auf Pop-ups können durch technische Kniffe wie die Nutzung von mehreren Usern oder die Implementierung von Warteschleifen überwunden werden. Die Fachbereichskollegen lernen schnell, dass sich die Scripts auch hervorragend zur Bereitstellung von Testdatenkonstellationen wie die Anlage von Geschäftspartnern oder die Bereitstellung von Vertragskonstellationen eignen.

Weiterhin gilt es, sich im Proof-of-Concept mit grundsätzlichen organisatorischen Fragen zu beschäftigen:

- Wer übernimmt die Ownership über die Testautomatisierungs-Scripts?
- Wann, von wem und wie werden die Scripts ausgeführt und gepflegt?
- Wie soll die Bibliothek mit Testautomatisierungs-Scripts strukturiert werden?
- Kann oder soll das Tool in die vorhandene Testinfrastruktur integriert werden?
- Wie ist das Defect-Management für automatisierte Tests zu organisieren?

Die Präsentation und die Diskussion der Ergebnisse des Proof-of-Concepts in einem gut vorbereiteten Workshop mit allen Prozessbeteiligten führen zur Aufnahme eines Pilotbetriebs für einen reduzierten Test-Scope.

Einführung und Ausbau

Die Begleitung des Pilotbetriebs geht einher mit der letzten Phase „Einführung und Ausbau“. Die Aufgaben sind unter anderem:

- Die Verankerung der Testautomatisierung im Testmanagement-Vorgehensmodell,

- die Einbindung in die Testinfrastruktur wie beispielsweise die Integration in den SAP Solution Manager,
- die Bereitstellung einer Dokumentation und die Schulung der Anwender und vor allem
- unter Berücksichtigung der Erfahrungen aus dem Pilotbetrieb, der sukzessive Ausbau des Scopes.

Im Idealfall erfolgt die Automatisierung von Tests obligatorisch im Rahmen des Softwareentwicklungsprozesses und ist obligatorisch bei der Budgetierung und Planung zu berücksichtigen.

Erfahrungen aus der Praxis

Was ist aus vergangenen Testautomatisierungs-Projekten zu lernen? Testautomatisierung eignet sich nicht für alle Testfälle. Aber eine Automatisierungsquote von weit über 60 Prozent des Testfallportfolios ist zu erreichen. Nicht außer Acht lassen darf man den Aufwand für die Pflege der Automatisierungs-Scripts nach dem Rollout von Releases. Ein Erfahrungswert ist dabei, mit rund zehn Prozent des ursprünglichen Erstellungsaufwands zu rechnen. Weiterhin freuen sich die Tester über den enormen Zusatznutzen durch die beliebig wiederholbare und effiziente Bereitstellung von Testdaten. Die Automatisierung genießt eine hohe Akzeptanz in den Fachbereichen. Der Aufwand für die Abarbeitung der Tests lässt sich gegenüber der manuellen Ausführung um mindestens 70 Prozent reduzieren. Der Nutzen ist mittelfristig signifikant höher als die Einführungskosten. Die Startkosten sind niedrig, das Risiko zu scheitern ist gering. Für alle, die von den eingangs skizzierten Veränderungen im Testmanagement betroffen sind, ist der Einstieg in die Testautomatisierung eine unverzichtbare Investition in die Zukunft. (ch) @