



© RTimages, Shutterstock.com

Fehlerfrei beim Going Live ist eher die Ausnahme

Zehn Regeln fürs SAP-Testmanagement

Testmanagement-Projekte gestalten sich oft schwierig: Die Software ist nicht rechtzeitig fertig, Testsysteme und Berechtigungen fehlen und Testfälle, falls vorhanden, sind von schlechter Qualität.

Von Dieter Koenen, Innobis

Nicht alle, aber viele Steine beim Testmanagement lassen sich mit den folgenden Regeln im Vorfeld aus dem Weg räumen.

Regel 1

Schreiben Sie ein Testkonzept: Warum sollte das Testmanagement-Team bei knappen Ressourcen und enger Terminierung den Aufwand betreiben, ein Testkonzept zu erstellen und abnehmen zu lassen? Die Kunst liegt darin, das Konzept möglichst kurz und knapp zu halten und komplexe Sachverhalte wie beispielsweise eine Testumgebung mit vielen Schnittstellen durch Grafiken darzustellen. Zu den wesentlichen Bestandteilen eines Testkonzepts gehören z. B. die Definition der Testobjekte und -ziele, der Abnahmekriterien und des Abnahmeverfahrens sowie die Darstellung der Testinfrastruktur (SAP-Systeme, Transportwege, Transporttermine). Lassen Sie sich das Konzept von der Entwicklung und der Fachseite abnehmen.

Regel 2

Investieren Sie genügend Zeit in die Qualität der Testfälle. Den Fachabteilungen fehlt häufig die Zeit, die Testfälle zu erstellen. Folglich werden diese aus alten

Vorhaben „recycelt“, sind zu allgemein, zu granular oder treffen nicht die zu testenden und abzunehmenden Funktionen. Das Resultat: Die Fachtester testen zu viel oder zu wenig, aber nicht die notwendigen Features. Der Fortschritt ist zu langsam, die Fehlerquote zu hoch und die Qualitätsverbesserungen sind zu gering. Die Fachabteilungen (und nicht die IT) sollten sich unbedingt die notwendige Zeit für die Erstellung der Testfälle nehmen.

Regel 3

Priorisieren Sie Ihre Testfälle. Oftmals möchte die Fachabteilung A alle neuen Features bis ins letzte Detail austesten und die bereits implementierte Software einem kompletten Integrationstest inklusive Batch-Test und End-2-End-Test unter Einbindung von Dritt- und weiteren Systemen unterziehen. Abteilung B hingegen wählt abhängig von der Termin- und Ressourcen-Situation einen pragmatischen Ansatz mit nur wenigen Testfällen. Wichtig ist, das richtige Maß für den Testumfang zu finden. Bewährte Praxis ist, die Testfälle zu priorisieren. Ein guter Ansatz ist das Risk-Based-Testing. Hier werden die Testfälle insbesondere hinsichtlich des Schadensmaßes bei Fehlfunktionen gewichtet.

Regel 4

Planen Sie mehrere Testzyklen. Der erste Testzyklus läuft üblicherweise sehr holprig an. Es gibt Blockaden, die Tester kennen die Anwendung noch nicht und es ergeben sich viele Fehler. Die hohe Zahl an Korrekturen kann zu Seiteneffekten führen, die den Retest bereits erfolgreich abgeschlossener Testfälle erfordern. Ehe man sich versieht, ist der Testzyklus vorbei. Legen Sie daher nach dem ersten Durchlauf eine Pause ein und geben Sie der Entwicklung die Chance, alle kritischen Fehler zu beheben. Starten Sie dann mit einem konsolidierten Software- und Datenstand einen weiteren Zyklus, in dem Sie zumindest alle hoch und mittel eingestuften Testfälle aus dem ersten Zyklus testen.

Regel 5

Kümmern Sie sich rechtzeitig um die Testumgebung. Eine typische Situation: Die Testfälle stehen bereit, die Fachbereichsmitarbeiter wollen beginnen, aber der Test kann nicht gestartet werden, da das SAP-System nicht zur Verfügung steht. Gründe dafür gibt es viele. Das System wurde nicht rechtzeitig reserviert und wird von anderen Projekten belegt. Oder die Entwicklung konnte die Soft-



Dieter Koenen, Manager Consulting Services bei Innobis, leitet seit über zehn Jahren große Testmanagement-Vorhaben im SAP-Banking-Umfeld. Er ist Experte für die Entwicklung und Praxiseinführung von Methoden und Verfahren in diesem Kontext.

ware nicht oder nicht vollständig auf die Testsysteme ausrollen. Oft fehlen auch wichtige und notwendige Schnittstellen oder geeignete User-IDs und Berechtigungen. Planen Sie daher frühzeitig Ihre Testumgebung und überwachen Sie die vollständige und korrekte Bereitstellung. Den ersten Pfeiler setzen Sie durch die Definition von Testeingangskriterien im Testkonzept. Sind wesentliche Bedingungen nicht erfüllt, starten Sie erst gar nicht. Definieren Sie im Konzept oder in einem Testumgebungsplan genau, welche SAP-Systeme mit welchem Datenbestand wann benötigt werden, welche Schnittstellen über welche Drittsysteme angebunden sein müssen, welche Benutzer-IDs mit welchen Berechtigungen bereitzustellen sind sowie wann Software-Korrekturen transportiert werden.

Regel 6

Nutzen Sie ein integriertes Test- und Defect-Management-Tool. Das am häufigsten für das Test- und Defect-Management eingesetzte Werkzeug ist Excel. Es funktioniert bis zu einer Anzahl von 100 bis 150 Testfällen bei geringem Defect-Aufkommen ganz gut. Sobald das Mengenvolumen steigt, wird es zunehmend problematischer, mit eigenen Excel-Lösungen zurechtzukommen. Insbesondere fehlt häufig die Übersicht, welche Defects welche Testfälle betreffen. Auch das Erzeugen des aktuellen Gesamtstatus oder des Testfortschritts ggf. mit Forecasts gestaltet sich schwierig. Oft können die Excel-Tabellen (und besonders die Makros) nur noch von wenigen oder im ungünstigsten Fall von einem Experten gewartet werden. Der Einsatz eines integrierten Test- und

Defect-Management-Tools ist die bessere Wahl. Für SAP-Anwender bieten sich die SAP-Solution-Manager-Komponenten Test Workbench und Help Desk an. Sie verursachen keine Lizenzgebühren, der Einführungsaufwand ist mit drei bis fünf Personentagen überschaubar und die Anwender sind mit den SAP-Oberflächen vertraut.

Regel 7

Führen Sie einen Test-Kick-off durch. Den Aufwand, den Sie in einen gut vorbereiteten Kick-off investieren, fahren Sie im täglichen Testbetrieb wieder ein. Auf der Agenda sollten Themen wie Ziele, Vorgehen, Termine, Quality Gates, Testeinschränkungen, Test- und Defect-Reporting sowie Regelmeetings und Telefonkonferenzen stehen. Auch wenn es viele Teilnehmer werden, laden Sie alle Beteiligten wie Fachtester, das Testmanagement-Team, die Entwicklung, ggf. die Batch-Steuerung und die Projektleitung zum Kick-off ein. Das erleichtert die spätere Kommunikation und hilft, viele unnötige Rückfragen zu vermeiden. Bewährt hat sich, die Kick-off-Unterlagen und Statusreports, aber auch Konzepte in einem einfach zugänglichen Medium wie MS SharePoint zur Verfügung zu stellen.

Regel 8

Starten Sie nicht zu früh mit den Tests. Das übliche Szenario: Die Test- und Abnahmetermine sind fest terminiert. Die Fachtester sind eingeplant. Die Entscheider wollen den Projekterfolg einfahren. Und das Entwicklungsteam kämpft noch mit schweren und/oder vielen Defiziten in der Software. Der Entwicklungsleiter möchte den Test verschieben; die Projektsteuerung möchte den Termin um jeden Preis halten. Oft wird dem Wunsch der Verantwortlichen nachgegeben und der Test mit Kenntnis der Defizite und den damit verbundenen Risiken gestartet. Was passiert? Die ersten Annahmetests verlaufen unter Umständen noch positiv. Aber dann finden sich viele Fehler, es kommt zu Testblockaden. Die Tester sind frustriert. Und vor allem kommt dies negativ in den Projektgremien, bei den involvierten Fachbereichen und unter Umständen auch beim Vorstand an. Daher die Empfehlung: Auch wenn Sie unter hohem Druck stehen, den geplanten Starttermin für die Fachtests einzuhalten, verschieben Sie die Tests, wenn die Software noch nicht testbereit ist. Sie sparen sich und den Testern jede Menge Ärger und vermeiden vor allem einen hohen Imageschaden für Ihr Projekt.

Was kann man präventiv tun? Bewährt haben sich die Definition und das Überprüfen von Quality Gates. Definieren Sie im Testkonzept, welche Vorbedingungen für den Start der Tests erfüllt sein müssen. Dazu gehören zum Beispiel der Nachweis über dokumentierte Entwicklertests, die Erstellung von Anwender-Dokumentatio-

nen, das Bereitstellen der Testinfrastruktur sowie die Auflistung von nicht gelösten Fehlern mit einem Korrekturtermin. Wird das Quality Gate nicht bzw. in einem nicht ausreichenden Maß erfüllt, gibt es kein Go für den Test.

Regel 9

Unterbrechen Sie die Tests bei Testblockaden. Das Fehlen von wichtigen Features, eine hohe Anzahl von Fehlern oder Einschränkungen in der Testinfrastruktur wie nicht performante Systeme können zu Testblockaden führen. Die Tester können aufgrund der aktuellen Gesamtsituation nicht starten. Parallel gibt es hohen Druck, Routinearbeiten am Schreibtisch zu erledigen. Natürlich macht es keinen Spaß, Tester in eng getakteten Fenstern nach Hause zu schicken. Aber die negativen Konsequenzen wie unzufriedene Tester und Panik in den Fachabteilungen oder eventuell beim Vorstand aufgrund der mangelhaften Softwarequalität sind im Zweifelsfall höher zu bewerten. Sie können die Unterbrechung dafür nutzen, die Software zu konsolidieren.

Regel 10

Kommunizieren Sie mit den Testern. Ein wesentlicher Erfolgsfaktor für Ihr Vorhaben ist die regelmäßige Kommunikation mit den Testern. Wichtig sind Transparenz über die aktuelle Situation, den Fehlerstand, Einschränkungen usw. Führen Sie regelmäßig Status- und Fehlermeetings mit den Testern und der Entwicklung durch. Bewährt haben sich beispielsweise tägliche Stand-up-Meetings. Versuchen Sie, täglich mit jedem direkt zu sprechen, um mehr über konkrete Test- und vermeintliche Fehlersituationen zu erfahren. Bewährt hat sich auch, bei komplexen Sachverhalten Entwickler zu Tests oder für den Retest von Fehlern hinzuzuziehen.

Fazit

Jedes Testmanagement-Vorhaben im SAP-Umfeld verläuft anders und den perfekten Projektablauf wird es wohl nie geben. Aber vieles wird mit den oben genannten Grundregeln einfacher. Letztendlich geht es darum, in einem von allen Beteiligten akzeptierten Prozess die Qualität der Software auf ein für den Produktivbetrieb notwendiges Niveau zu bringen. Dass dabei nicht alle Fehler vor dem Going Live behoben werden, ist eher die Regel und nicht die Ausnahme.



Bitte beachten Sie auch den Community-Info-Eintrag ab Seite 99

