

Software-Tests



Effektives und effizientes SAP-Testing

Um die Innovationskraft in Unternehmen zu stärken, stellt SAP in immer enger getakteten Zeiträumen Updates bereit. Herkömmliche Testvorgehen müssen an diese neuen Release-Zyklen angepasst werden. Hierfür benötigen Unternehmen gezielte Change-Impact-Analysen sowie Testautomatisierung.

Neue Versionen von SAP sind inzwischen Cloud-fokussiert und konzentrieren sich vor allem darauf, Applikationen zu modernisieren, die Nutzerfreundlichkeit zu verbessern und Ende-zu-Ende-Prozesse zu bündeln. Um von Neuerungen zu profitieren, müssen Unternehmen Updates in immer kürzeren Zeiträumen einspielen. Hierfür aktualisieren rund



Alexander Ertl,

Group Product Manager SAP bei Tricentis

40 Prozent der Hybrid- und Cloud-Kunden ihr System monatlich, 44 Prozent sogar schon täglich. Die Herausforderung dabei: Traditionelle Software-Teststrategien, die sicherstellen sollen, dass Updates keine Probleme hervorrufen, sind diesen neuen Anforderungen nicht mehr gewachsen. Tatsächlich bieten ca. 67 Prozent der bisher funktional durchgeführten Tests keinen Mehrwert.

Um schnelles, intelligentes und automatisiertes Software-Testing zu ermöglichen, müssen bisherige Strategien, die auf Key-User-Tests, „Testing for Green“ und aufwendigen Hypercare-Phasen basieren, also adaptiert und modernisiert werden. Gemäß dem Pareto-Prinzip lassen sich 80 Prozent des Geschäftsrisikos mit nur 20 Prozent der Test-Cases

abdecken. Es gilt, diese 20 Prozent der Tests zu identifizieren.

Das gelingt mit Change-Impact-Analysen. Sie durchleuchten das gesamte SAP-Ökosystem, um die von einem Update betroffenen Objekte, Objektreferenzen und -abhängigkeiten zu erkennen. So lassen sich die tatsächlich nötigen Tests identifizieren, die zeigen, ob ein Update Probleme hervorruft – und die Anzahl der Tests sinkt signifikant. Im nächsten Schritt kann es zudem sinnvoll sein, Tests zu automatisieren. Dies ist vor allem bei Updates, die Kerngeschäftsprozesse beeinflussen, sinnvoll. Zudem sind kontinuierliches Testdaten-Management, Service-Virtualisierung, verteilte Ausführung und CI/ CD-Integration sowie Lasttests wichtig.