

Einführung eines neuen Werkzeugs erfordert Integrationsleistungen

Modellbasierter Ansatz klappt auch in Riesenprojekten

T-Systems hat in einem der größten Systemintegrationsprojekte Europas, das Offshore-Elemente und mehr als 400 Mitarbeiter umfasste, einen modellbasierten Ansatz eingeführt. Hierzu wurde ein Innovationsprojekt aufgesetzt, um sicherzustellen, dass praktische und betriebliche Faktoren im Blickfeld stehen.

Die drei Hauptphasen des Innovationsprojekts waren: Die Einführung modellbasierten Testens in einen bestehenden Testprozess, die Durchführung eines Pilotprojekts und schließlich die praktische Realisierung des Ganzen. Der bestehende Testprozess beruhte auf dem Standard des International Software Testing Qualifications Board. Die wesentlichen Leistungen des Zielprojekts waren funktionale und betriebliche Abnahmetests des gesamten System-Stack des Kunden. Die Testeinheit erhielt von den Business-Ownern für jedes Release textbasierte Lösungsdokumente, die als Testgrundlage dienen.

Teil des Ansatzes war ein Forschungsprojekt in enger Zusammenarbeit mit der TU München. Der Projektplan umfasste drei Phasen von jeweils sechsmonatiger Dauer: Modellierung, Integration, Automation.

Das Pilotprojekt sollte durch eine große Zahl von Anwendungen und Schnittstellen technisch komplex sein. Mehrere Anwendungen stehen dabei miteinander in Verbindung, um Geschäftsprozesse abzuwickeln. Und die Organisation war auf Deutschland, Indien und Ungarn verteilt.

In der Modellierungsphase wurde ein Konzept für die Einführung von Modellierungskonzepten in den bestehenden Testprozess erstellt. UML 2.0 wurde für die Modellierung der Testgrundlage verwendet. Die Geschäftsanforderungen aus den Lösungsdokumenten des Kunden wurden in Anwendungsfall- und Aktivitätsdiagrammen erfasst. Das Resultat war ein Modell über das zu testende System und jeden Testfall. Es diente als Grundlage für die manuelle Testdurchführung.

Das Konzept musste auch das Offshoring berücksichtigen. Der textbasierte Ansatz, für den Testfälle ins Englische übersetzt werden, ist nicht sehr effizient, was einer der Haupt-

gründe dafür war, Modellierung in Betracht zu ziehen.

Das Modellieren der Anforderungen in UML war ein notwendiger Schritt, um andere Vorteile für das Testen zu ermöglichen. Denn der Kommunikationsaufwand war hoch aufgrund von Nachfragen bezüglich der Testspezifikationen, Übersetzungen erwiesen sich oft als Hauptfehlerquelle und implizites Wissen erschwerte das Verständnis von Bedeutung und Inhalt textlicher Testfälle.

Die Modellierung war zeitaufwändiger als erwartet, trotz Schulungen und der Verfügbarkeit von Experten. Der Einsatz von Enterprise Architect (EA) als Modellierungs-Tool erwies sich als zweischneidig.

Der Aufwand für die Kommunikation war im Projekt hoch

EA ist einfach in der Handhabung, leicht zu erlernen und lässt dem Modellierer völlige Freiheit bei der Erfassung der Anforderungen als Modelle. Diese Freiheit kann Probleme verursachen, selbst wenn Richtlinien existieren. Denn zum einen muss sich der oder die Modellierende stets an die Vorgaben halten, zum anderen muss es Kontrollmechanismen geben, die sicherstellen, dass die Richtlinien korrekt angewendet werden.

Während der Modellierungsphase zeigte sich, dass der Business-Eigner seine Anforderungen und Systemarchitektur so modellieren muss, dass sie von Entwicklungsabteilungen verwendet werden können. Er entschied sich nach einem Evaluierungsprozess für das Tool Innovator von MID, das eine integrierte Komponente für modellbasiertes Testen enthält. Innovator sollte nicht nur als Brücke zwischen

dem Kunden und der Testeinheit dienen, sondern auch eine viel genauere Definition der Modellierungsprozesse ermöglichen als im ersten Ansatz mit Enterprise Architect.

Mit der Integration von Innovator in den Testprozess entstand eine modellbasierte Schnittstelle zum Kunden. Die Testeinheit musste nicht mehr die textbasierten Anforderungen modellieren und konnte allein dadurch große Produktivitätssteigerungen erreichen. Dazu kam die Möglichkeit, aus den Modellen automatisch Testfälle zu erzeugen.

Die Einführung des Modellierungs-Tools in die bestehende Testlandschaft erforderte Integration:

- Für die Integration der testspezifische Aspekte in den Anforderungsspezifikationsprozess wurde eine Domain Specific Language (DSL) eingeführt. Die Testorganisation nahm an der Entwicklung der DSL teil, um den eigenen Standardtestprozess mit einzubeziehen.
- Für die Integration in das Toolset der Testeinheit bedurfte es einer XML-basierten Schnittstelle zwischen Innovator und dem eigenen Standardtestmanagement-Tool.
- Für die Integration in den bestehenden Testprozess mussten einige Prozessschritte verändert werden, inklusive Testplanung und -spezifikation.

Als nächste Phase des Innovationsprojekts folgte die Automatisierung. Sobald eine klar definierte DSL zur Verfügung steht, besteht die Möglichkeit automatisch durchführbare Testfälle zu erzeugen. Ziel ist es dabei, Modellierungsstrukturen in einem ausreichend hohen Abstraktionsgrad zu definieren, um die Vorteile für weitere Projekte und Kunden zu nutzen.

Graham Bath, Lilith Al-Jadiri,
Test Factory, T-Systems/mv