

Gegenüberstellung QF-Test vs. Tosca

Einleitung

Bei der Gegenüberstellung lag der Fokus auf der Handhabung, so wie der benötigten Einarbeitungszeit und natürlich auch der unterschiedlichen Einsatzmöglichkeiten. Im Bezug auf die Einsatzmöglichkeiten ähneln sich die beiden Lösungen sehr, verfolgen aber unterschiedliche Ansätze. Für die unterschiedlichen Tests kamen verschiedene Anwendungen zum Einsatz, wobei der Fokus auf Desktopanwendungen lag. Die mobilen Anwendungen für Android und iOS wurden hier allerdings nicht betrachtet, da wir im Haus dazu keine Anwendungsmöglichkeiten haben. Zudem wurden die Wartbarkeit und Erweiterungsmöglichkeiten geprüft. Der Vergleich wurde auf einem Laptop mit i7 CPU durchgeführt. Tests innerhalb von VM (virtuellen Maschinen) gab es ebenfalls, diese fließen aber nicht zu großem Maße in die Bewertung ein.

Voraussetzungen/Installation

Die Installation von QF-Test ist, im Vergleich zur Installation von Tosca, sehr einfach und intuitiv. Bei Tosca hingegen muss im Vorfeld die Infrastruktur, bei der On-Premise Lösung, gestellt und eingerichtet werden. Hierbei wird ein Server benötigt, auf dem der "Tosca Server" installiert werden kann, der für die Verteilung der Agents und den Upload der Repositories zuständig ist. Hinzu kommen noch eine Datenbank für die Repositories sowie der Lizenzserver, welcher sich bei uns auf derselben VM wie der Tosca Server befand. Die Installation von Tosca Commander, der eigentlichen Automatisierungsanwendung, läuft dann relativ unkompliziert ab. Allerdings müssen die Anbindungen an die verschiedenen Server bzw. Teile der Tosca Infrastruktur manuell hinterlegt werden, an verschiedenen Stellen im Programm oder den Konfigurationsdateien. Außerdem benötigt Tosca höhere Hardwareanforderungen, auch auf Grund der Infrastruktur.

Für beide Tools ist es grundsätzlich nötig, die Begrifflichkeiten zu kennen und etwas Test- bzw. Entwicklerhintergrund mitzubringen. Laut Tricentis ist das für Tosca nicht nötig, aber interne Erfahrungen haben gezeigt, dass ohne vorherige Kenntnisse, der Ablauf und das Vorgehen falsch verstanden werden können. Zur Erläuterung hierzu: Es wurden 3–4 Personen ohne Programmierhintergrund über den Einsatzzeitraum von Tosca an das Tool herangeführt. Leider konnte keiner der eingesetzten Personen Tosca komplett bedienen und diese lehnten auch den weiteren Einsatz ab, da Tosca zu komplex aufgebaut ist.

Einsatzmöglichkeiten

Beide Tools unterstützen viele Technologien, was den Einsatz bei unterschiedlichen Projekten erleichtert. Das Hauptprojekt, welches den Vergleich zugrunde legt, war eine Java SWING Anwendung mit Browserteil für den Login. Mit beiden Tools konnten die Anwendungsfälle automatisiert/realisiert werden.

Zu erwähnen ist noch, dass Tosca nicht in eine Pipeline integriert werden kann, jedenfalls nicht ohne größeren Aufwand zu betreiben. Hier ist die Infrastruktur des Tools zu komplex aufgebaut und die Anforderungen an die Hardware des Clients zu hoch, um eine schnelle und einfache Integration zu ermöglichen. Die automatisierte Ausführung erfolgt über einen Aufruf des Servers, der wiederum den Agent anweist, die Ausführung zu starten. Dies funktioniert zwar, ist aber insgesamt eine Doppelung innerhalb der Pipeline und im modernen Dev-Ops Umfeld keine agile Lösung, da Freischaltungen u.Ä. erforderlich sind.

Erster Start, Handhabung und Übersicht

Der erste Start bei Tosca gestaltet sich als sehr umfangreich, da viele Parameter gesetzt werden müssen, wie zum Beispiel Erstellung des Repositories und deren Anbindungen an die Server. Bei QF-Test hingegen kann man direkt im Tool das Tutorial starten und die Demo Suites anschauen, ohne großen Aufwand für weitere Einstellungen. Die Oberfläche von Tosca Commander ist relativ unübersichtlich und vollgepackt mit mehreren Funktionen über ein altes Ribbon, was die Anwendung selbst sehr unzugänglich erscheinen lässt. Hierbei ist natürlich zu beachten, dass Tosca noch weitere Teile für den Softwaretest im Tosca Commander bereitstellt. Zum Beispiel können Anforderungen direkt im Tool erstellt oder komplexere Datentabellen über TestDesign realisiert werden.

Die Tutorials gestalten sich ebenfalls sehr unterschiedlich. Die Tutorialserien von Tricentis kosten insgesamt ca. 20 bis 40 Stunden. Diese sind notwendig, um überhaupt verstehen zu können, wie das Tool arbeitet. Leider sind die Tutorials nur auf Englisch vorbereitet und teilweise fehlerhaft, zum Beispiel bei der Interpunktion. Für das Tosca-Tutorial muss zudem ein Account auf der Supportseite eingerichtet werden und das Handbuch bietet leider keine gute Grundlage, um mit dem Tool zu starten. Im Vergleich zum Tutorial von QF-Test ist der Zeitaufwand wesentlich geringer und es wird keine zusätzliche Anmeldung benötigt. Außerdem ist das Handbuch sehr gut aufbereitet und in der neueren Version nochmals übersichtlicher. Man kann mit QF-Test auch direkt durch "Learning-By-Doing" starten und wird dabei durch "Step-By-Step" Anleitungen direkt im Tool unterstützt.

Wartbarkeit

Die Wartung der Komponenten gestaltet sich in Tosca als sehr aufwendig, da diese bei Nicht-Erkennung komplett neu aufgenommen werden müssen. Die Hierarchie der Erkennung bei Tosca lautet: IDs, Eigenschaften (Kombination), Path, Pixel (Positions X/Y). Die Erkennung ist zu ca. 80% in Ordnung, erfordert aber hohen manuellen Aufwand. In QF-Test können die Elemente auf mehrere qfs-Dateien verteilt werden, was die Erkennung bzw. das Finden vereinfacht. Je nach Anwendung kann der manuelle Aufwand natürlich auch hier ansteigen. Die Komponentenerkennung ist aber deutlich höher als bei Tosca.

Bei größeren Testszenarien, wie zum Beispiel Prozessketten, kann die Komplexität enorm ansteigen. Vor allem bei Tosca sind die Testfälle dann sehr schlecht les- und nachvollziehbar, da die Komponenten nicht direkt angesteuert werden können, sondern meistens geschachtelt in einer "Hülle" liegen. Die automatische Bezeichnung der "Komponentenhülle" hat meistens wenig Bezug zur Fachanwendung und muss manuell beschrieben werden.

Tosca bietet zusätzlich eine AI Komponentenerkennung namens "Ara". Diese wurde mit einem Testzugang ausprobiert, brachte aber wenig bis keinen Mehrwert. Außerdem wird hier eine zusätzliche Anmeldung benötigt und die "Ara" kann dann nur von bestimmten Personen genutzt werden. Extrakosten für die Verwendung fallen an.

Tosca Testsuite	QF-Test
Technologieunterstützung (Testmöglichkeiten)	
API	Java GUI
Web	Web
Windows	Windows
Mobile	Android
Datenbank	PDF
Infrastruktur	
Hardwareanforderungen relativ hoch für lokale Ausführung Hardwareanforderungen an Server hoch	Hardwareanforderungen moderat
Tosca Server/Lizenzserver (Auftragsverteilung, Lizenzverteilung, Repository Bereitstellung)	Lizenz als Datei
Datenbankserver	
Agents für die Testausführung	
CI/CD Integration	
Schwierige Integration, da Portfreigaben zum Server/Agents erforderlich	Batch-Modus vorhanden
Ausführung nur auf Tosca Agents möglich (Tosca Testsuite muss installiert sein)	Über Jenkins realisierbar Azure ebenfalls möglich
Lokal über TC-Shell (in Tosca enthalten) möglich, Aufwand aber relativ hoch	Lokal über Aufgabenplanung (Windows) ebenfalls möglich
Erweiterbarkeit / Plugins	
Nicht erweiterbar	Nicht erweiterbar
Weitere Tricentis Tool kompatibel, wie NeoLoad oder qTest	
Voraussetzungen	
Basiswissen Softwaretest (Begriffe)	
Basiskurs Tosca Testsuite, Pfad 1 + 2 (Supporthub Tricentis), Aufwand ca. 20–40 h	Tutorial Kurs (Entwicklerseite) Aufwand ca. 6–8 h
Installation	
Nur Windows	Windows Mac OS Linux
Testskripterstellung	
Scannen der GUI insgesamt möglich	Schnellstarthilfe für Anwendungsstart
Drag&Drop Testfallerstellung	Scan&Replay, mit Anpassungen
einfach, aber durch Drag&Drop unnötig komplex und unübersichtlich	Drag&Drop
Kein 'echtes' Skripting möglich	Skripting für komplexere Aktionen/Prüfungen möglich
Repositories	
Testskripte in Datenbank gespeichert	Git, local
Versionierung möglich, Rücksprung aber fehleranfällig	Versionierung bei Git vorhanden
Benutzerfreundlichkeit	
Mittlere Benutzerfreundlichkeit: unübersichtlich; Relativ einfache Bedienung nach längerer Einarbeitung	Hohe Benutzerfreundlichkeit: intuitive Bedienung; übersichtlich

Einarbeitung/Recherche	
Hohe Lernkurve	Gute Lernkurve
Gute (englische) Workshops direkt von Tricentis, leider nicht angepasst für den deutschen Markt (Rundungen, Uhrzeiten, etc.)	Lernen durch Ausprobieren möglich
Keine weiteren (kostenfreien) Möglichkeiten vorhanden	Handbuch (in Deutsch) vorhanden
Support	
Lange Reaktionsdauer	Innerhalb von 12 h
Nur ein englischer Community Hub, geringer Nutzen	
Relativ hoher Aufwand für die Fehlerdokumentation	
Wartung	
Aufwendig, da Scans erneut durchgeführt werden müssen	Aktuell nicht abschätzbar und je nach Projektgrad sehr unterschiedlich (kann mit aufgeteilten Libraries vereinfacht werden)
Drag&Drop erschwert die Wartung zusätzlich	
Für Anfänger ist die Wartung teilweise nicht möglich	
Referenzen bzw. Modulverweise müssen teils manuell angepasst werden	
Tosca Server sowie Agents müssen wöchentlich gesichtet und ggf. gewartet werden	
Sonstiges	
Versionsupdates können die Repositories unbrauchbar machen, wenn Technologien wegfallen	Mit Version 9.0 auch Barrierefreitest möglich

Fazit

Die Handhabung und das Erstellen der Testskripte sowie der Installationsprozess und die Wartung sind bei QF-Test insgesamt besser gelöst, als bei Tricentis Tosca. Außerdem ist die Flexibilität durch das zusätzliche Skripting, mit JavaScript oder Jython, in QF-Test nochmal um einiges angenehmer, als die strikte Drag&Drop Skripterstellung in Tosca. Tosca deckt zwar insgesamt einen größeren Bereich ab, durch die zusätzlichen Tools wie qTest oder NeoLoad, allerdings sind diese sehr kostenintensiv und bringen nur geringfügigen Mehrwert für agile Projekte. Des Weiteren sind die Kosten für Tosca insgesamt um knapp das 5–6 fache höher als bei QF-Test, selbst wenn man 5 Technologien für die Lizenz einkauft, bei QF-Test. Die Komponentenerkennung ist ebenfalls bei QF-Test ca. 20–30 % höher bzw. auch einfacher in der Nachvollziehbarkeit, was die Wartung begünstigt.

Insgesamt ist QF-Test das besser Automatisierungstool für Desktopanwendungen, für einzelne und mehrere Projekte. Für die Automatisierung des Projektes wird dem Kunden vorgeschlagen, QF-Test zu verwenden.

	QF-Test	Tricentis Tosca
Installation	5	3
Infrastruktur	5	2
Erstellung Testskripte	4	3
Komponentenerkennung	5	3
Tutorials / Dokumentation	5	2
Lernkurve	4	4
Zeitaufwand Einarbeitung	4	2
Wartbarkeit	4	2
Erweiterbarkeit	2	5
Support	5	1
Insgesamt	43/50	27/50

Punkteverteilung

Stufe	Punkte
Sehr gut	5
Gut	4
Mittel	3
Ausreichend	2
Ungenügend	1