


GUI-Testtool für Java, Web, Android & native Windows Anwendungen

 ✓		
Das Tool für professionelle GUI Testautomatisierung		
Features		
GUI Technologie der zu testenden Anwendung	Java Anwendungen: Swing, JavaFX, AWT, SWT, Eclipse Plug-Ins, RCP, Applets, JavaWebStart, RIA, ULC, CaptainCasa, JavaFX SubScene Komponenten, JIDE Common Layer Web Anwendungen: Browser: Chrome, Firefox, Opera, Safari, Edge (Chromium basiert), Microsoft Edge Legacy, Internet Explorer; Headless Browser Versionen für Chrome, Firefox und Edge (Chromium basiert) HTML 5, AJAX: QF-Test unterstützt umfassend Frameworks wie Angular, React und Vue.js aber auch viele konkrete UI-Toolkits wie GWT, Smart GWT, ExtGWT, ExtJS, ICEfaces, jQueryUI, jQueryEasyUI, KendoUI, PrimeFaces, Qooxdoo, RAP, RichFaces, Vaadin und ZK. Weitere Toolkits können bei Bedarf mit geringem Aufwand integriert werden, z.B. SAP UI5, Siebel Open UI und Salesforce. <u>Testen von Electron und Wehswing Anwendungen wird unterstützt</u>	
	Android Anwendungen: Native Android Apps, mobile Web Apps und hybride Apps auf Android ab Android 7 auf echten Geräten und mit dem Emulator von Android Studio	
	Windows Anwendungen: klassisches Win32, .NET (oft mit C# entwickelt), Windows Presentation Foundation (WPF), Windows Forms, Windows Apps / Universal Windows Platform (UWP) mit XAML Steuerelementen, moderne C++ Anwendungen (z.B. mit Qt)	
	Hybride Systeme mit der Kombination mehrerer GUI Technologien, auch eingebettete Browser-Komponenten (JavaFX WebView, JXBrowser, SWT-Browser)	
	PDF Dokumente können wie eine normale Anwendung getestet werden (textuelle und grafische Checks für einzelne Elemente)	
	PDF Dokumente können wie eine normale Anwendung getestet werden (textuelle und grafische Checks für einzelne Elemente)	
GUI Unterstützung abhängig vom Betriebssystem	Java Anwendungen: Swing und JavaFX: Windows, Linux, Unix, macOS SWT: Windows, Linux-GTK; Solaris-GTK auf Anfrage Web Anwendungen: Windows, Linux, macOS Windows Anwendungen: Windows	
Testprinzip	Aufnahmefunktion (Capture/Replay) für direkte und effiziente Erzeugung von Testsequenzen zur Weiterverarbeitung in komplexere Testfälle mit Ablaufsteuerung, Parametrisierung, Modularisierung und erweiterten Skripting-Möglichkeiten. Alles kann angepasst werden.	
Teststrukturierung	Klar und übersichtlich dank graphischer Darstellung der Testfälle und Aktionsknoten in Baumstruktur. Projekte können durch Testsuiten und Bibliotheken modular aufgebaut werden.	
Komponentenerkennung, Robustheit der Tests	Stabile Komponentenerkennung unabhängig von geometrischen Eigenschaften, auch von komplexen Elementen wie dynamischen Bäumen und Tabellen. Tests sind robust und tolerant gegenüber Änderungen am GUI.	
Wiederverwendbarkeit, Wartungsaufwand	Hohe Wiederverwendbarkeit von Testbausteinen durch modularen Aufbau z.B. durch Prozeduren, Kapselung von Komponentenzugriffen etc.	

Datengetriebenes Testen	Direktes Einlesen von CSV oder Excel Dateien, Nutzung von SQL Datenbank-Abfragen, XML Dateien. Über Skriptenerweiterung können beliebige andere Quellen eingebunden werden.	
Schlüsselwort getriebenes Testen/ Behavior-driven Testing	Nutzung von Schlüsselworten (Keywords) zur Implementierung und Steuerung von Testfällen, auch mittels externen Testspezifikations-Dokumenten oder -Tools (z.B. Excel oder Testmanagementtools)	
Last- und Performancetests	Last- und Performancetests durch synchronisierte, parallele Ausführung, auch auf mehreren Maschinen. Für Web in Kombination mit Tools wie JMeter oder NeoLoad.	
Docker Container	QF-Test unterstützt das Testen in Docker Containern	
Protokolle, Testdokumentation, Reports	Übersichtliche, detaillierte Protokolle inkl. Screenshots der Fehlersituation werden immer generiert. Konfigurierbare Reports in verschiedenen Formaten (HTML, XML, JUnit), Test- sowie Prozedurdokumentation auf Knopfdruck oder automatisiert erzeugbar.	
JIRA /REST	JIRA und JIRA Plugins wie TestRail, Zephyr, X-Ray, TM4J mittels REST integrierbar, oft auch über CI Tools wie Jenkins.	
Erweiterbarkeit durch Skripting	Freie Funktionserweiterungen und kundenspezifische Prüfungen/Aktionen durch eingebettetes Skripting (Jython, Groovy und JavaScript). Über Scripting-API voller Zugriff auf alle Objekte der Anwendung (SUT) und Ausführung eigenes Codes in der Anwendung bzw. im Browser.	
Testmanagement	Basisfunktionalitäten für kleinere Projekte in QF-Test enthalten. Integration verfügbar bzw. möglich für: u.a. ALM/QualityCenter von MicroFocus/HP, TestBench von Imbus, QMetry, Klaros von Verit, TestLink, IBM Rational Quality Manager, Jira und Jira Plugins wie TestRail, Zephyr, X-Ray, TM4J.	
Continuous Integration	Flexible Integration in Build-Tools: u.a. Jenkins, GitLab CI/CD, Bamboo, Travis CI, Circle CI, TeamCity, Ant, Maven, CruiseControl	
Virtuelle Desktops	u.a. Citrix, VMware, VirtualBox	
Versionsverwaltung	Gute Versionierbarkeit z.B. mittels Git, SVN/Subversion, CVS, Mercurial durch XML Format der relevanten Dateien.	
Fehlerverfolgung	Kann mittels offener Schnittstellen und RESTangbunden werden, z.B. Jira, MantisBT, Bugzilla	
Robot Framework	Robot Framework Keywords können einfach als QF-Test Prozeduren implementiert und ausgeführt werden.	
Testdurchführung		
Testvorbereitung	Ein Schnellstart-Assistent unterstützt bei der Erzeugung einer passenden Startsequenz für die zu testende Anwendung, abhängig von der zugrunde liegenden GUI Technologie.	
Testfallvorbereitung	Abhängigkeitsmanagement für Testvor- und Nachbereitung für unabhängig lauffähige Testfälle, inklusive automatischer Fehlerbehandlung.	
Verifikationspunkte	Direktes Aufnehmen von Standard-Checks, kundenspezifische Verifikationen variabel über Skripting implementierbar.	
Bildvergleiche	Direktes Aufnehmen von Image Checks möglich. Zahlreiche Algorithmen auch für unscharfe Bildvergleiche - komfortabler Diff-View zur Kontrolle bei Abweichungen.	
Object mapping	Komponenteninformationen werden zentral während der Aufnahme in einem eigenen Bereich in der Testsuite gespeichert, jederzeit editierbar. Mechanismen für Referenzsuche und automatisches Update.	
Intelligente Objekt-Suche	Mit SmartIDs können Komponenten direkt anhand von charakteristischen Eigenschaften insbesondere dem zugehörigen Label angesprochen werden. Das Setzen eines Geltungsbereichs (Scope) erlaubt das Einschränken der Objekt-Suche z.B. bei mehrfach vorkommenden Elementen.	

Generische Komponenten	Die Abbildung GUI spezifischer Komponenten in generische Komponenten (Buttons, Textfelder...) erlaubt die Wiederverwendung von Tests über Technologien hinweg sowie den Einsatz von allgemein gültigen Aktionen ohne jede einzelne Komponente erfassen zu müssen.	
Testausführung per Kommandozeile	Ausführung im Batch-Modus möglich mit umfangreichen Konfigurationsoptionen per Kommandozeilenparameter, auch zur Integration in Build-Umgebungen.	
Entfernte Ausführung	Testausführung auch auf entfernten Rechnern im Daemon-Modus.	
Fehlerbehandlung	Automatische Fehlerbehandlung garantiert das Fortführen des Gesamttestlaufs ohne Abbruch. Fehler werden für spätere Analyse protokolliert.	
Test-Debugger	Volle Debuggerfunktionalität inklusive Breakpoints und Variablenanalyse.	
Benutzerfreundlichkeit		
Bedienkomfort	Einfach und intuitiv zu bedienen mit klar strukturierter Baumansicht für bequemes Bearbeiten der Testfälle, z.B. Copy/Paste und Drag&Drop; Capture/Replay für einen schnellen Start.	
Benötigte Vorkenntnisse	Für Standardnutzung keine Programmierkenntnisse erforderlich Für Skripterstellung Kenntnis der Standardskriptsprachen Jython, Groovy und JavaScript hilfreich.	
Teamarbeit	Modularer Aufbau von Testsuiten und importierten Bibliotheken in Verbindung mit Versionsverwaltung erlaubt effiziente Teamarbeit.	
Lizenzen und Preise		
Produktvarianten	QF-Test ist in verschiedenen Produktvarianten variabel konfigurierbar für die unterstützten GUI Technologien erhältlich (Swing, JavaFX, SWT, Web, Android und Windows) - Kaufen Sie nur das, was Sie brauchen.	
Lizenztypen	Entwicklerlizenz - zum Erstellen (und Ausführen) von Testfällen und Runtime-Lizenzen - zum Ausführen von (nächtlichen) Tests.	
Lizenzmechanismus	Alle Lizenzen sind floating (d.h. nicht an eine bestimmte Person gebunden). Standard-Lizenzen funktionieren innerhalb eines (lokalen) Netzwerks, für netzwerkübergreifende Nutzung bieten wir einen Lizenzserver.	
Kaufen oder Mieten	Wir bieten Kaufoption und Miete auf Jahresbasis, für Lasttestlizenzen auch kürzere Laufzeiten.	
Support und Service		
Download und gratis Testen	Freie Demoversion anonym zum Download. Demoversion ohne Lizenzregistrierung lauffähig. Gratis Testlizenz für 4 Wochen inkl. freiem Support	
Installation	Einfache Installation unter Windows, macOS und Linux mit wenigen Clicks.	
Support	Unterstützung direkt von den Entwicklern und Testern von QF-Test auf Deutsch, Englisch und auch Französisch.	
Wartungsvertrag	Pflegevertrag (Support + Upgrades) auf jährlicher Basis beinhaltet Unterstützung per E-Mail und Telefon	
Schulung, Beratung	Individuelle Schulungen oder Beratung bei Ihnen vor Ort. Regelmäßige offene Standard-Schulungen bei QFS. Alle Angebote auch online möglich.	
Dokumentation	Umfangreiches Handbuch, Selbststarter-Tutorial, Videos, Blog, Online Suchfunktion, Online-Hilfe per Rechtsklick in QF-Test, allgemeine und technische FAQ.	