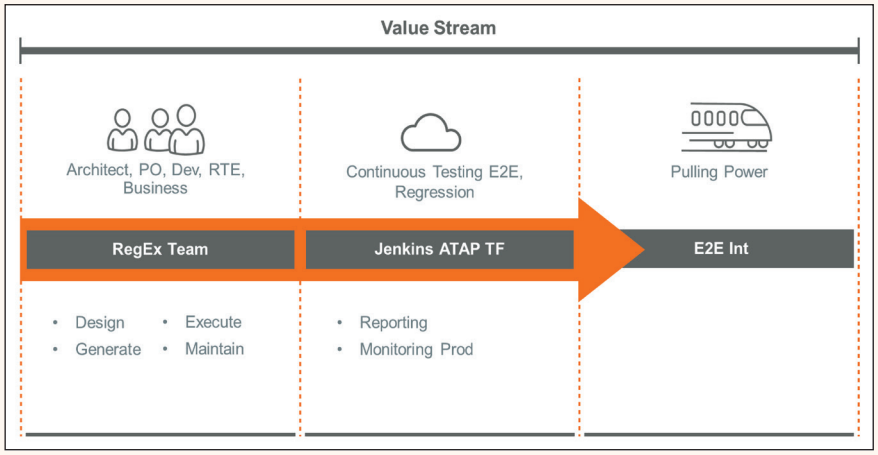
»Modernes Testen in einer technischen Welt«

In diesem Artikel wird beschrieben, was unter Testen von Hardware-in-the-Loop (HIL) in einem End-to-End Testszenario (E2E) mit Web, Middleware und Mobile Environment verstanden wird und wie dieses Vorgehen erfolgreich in eine Agile Tool Chain integriert wird, um automatische Testreihen für das Human Machine Interface (HMI) eines Fahrzeugs durchzuführen.



Der digitale Wandel wird das Mobilitätsverhalten der Menschen nachhaltig verändern. Auch die Automobilbranche muss mit den technologischen Möglichkeiten Schritt halten und ihre Produkte an die Wünsche und Bedürfnisse der Kunden anpassen. Dabei werden für die breite Masse von Kunden vor allem smarte und interaktive Services, die auf komplexen, vernetzten Softwareprogrammen basieren, immer wichtiger.

Neue Herausforderungen und Chancen für Automobilhersteller

Das zentrale Bedienelement für die Elektronik eines PKW ist das Human Machine Interface (HMI). In den meisten Modellen in der Mittelkonsole verbaut, erlaubt es dem Fahrer die Bedienung verschiedener Systeme vom Radio über die Navigation bis hin zum Reifendruck. Somit stellt das HMI eine Einheit im PKW dar, die mit fast allen elektronischen Steuerelementen (Electronic Control Unit, ECU) des Fahrzeugs verbunden ist. Durch die gestiegenen Erwartungen der Kunden an die Konnektivität ihres PKW werden über das HMI nicht mehr ausschließlich im Automobil verbaute elektronische Steuerelemente geleitet, sondern auch Informationen externer Plattformen, wie zum Beispiel Kurznachrichten und Instant Messages.

Diese hohe Komplexität und der Wunsch nach einem immer schnelleren Go-to-Market bedeuten, dass Software nicht mehr traditionell, das heißt, getrennt nach zuständigen Fachabteilungen, getestet werden kann, sondern einem Gesamtdurchlauf (End-to-End) standhalten muss. Denn mit der steigenden Anzahl der zu testenden Spezifikationen durch immer mehr elektronische Steuerelemente nimmt die Anzahl der implementierten, miteinander vernetzten Computerplattformen (Cloud, Mobil und Embedded Systems) zu. In der Vergangenheit war ein vollumfänglicher End-to-End-(E2E)-Test über alle Plattformen hinweg und unter Einbeziehung sämtlicher Steuerelemente im Anschluss an die Integrations-, Funktions- und Akzeptanztests nicht möglich.

Als eine weitere Konsequenz müssen sich die Fachabteilungen wie IT und Softwareentwicklung, die sich um die Backend-Systeme für Kundenservices kümmern, stärker mit den Fahrzeug-Entwicklungsabteilungen verzahnen und enger zusammenarbeiten.

Fachabteilungen werden zu neuen Teams

Anstelle von autonom agierenden Abteilungen entlang der Wertschöpfungskette der Forschung und Entwicklung sowie der Qualitätssicherung neuer PKW werden funktionsübergreifende Teams mit durchschnittlich sieben Mitgliedern gebildet. Neben dem Product Owner (PO) und den Architekten sind die Entwickler ebenso Teil dieser Teams wie die E2E-Tester.

Die Zusammenarbeit erfolgt dabei nach dem heutigen Standard der Softwareentwicklung auf Basis agiler Methoden. Am Anfang der Entwicklung steht die Definition eines angenommenen Nutzens für den Fahrer, der im Anschluss in einem iterativen Prozess innerhalb eines vorgegebenen Zeitrahmens entwickelt, gebaut und getestet wird, dem sogenannten Sprint. Die Kommunikation und der Informationsaustausch erfolgen in Scrum of Scrum Meetings und die Teams verwenden das Scaled Agile Framework for Enterprise Model (SAFE). Diese Neuausrichtung der Teams und ihre Zusammenarbeit garantieren kürzere Entwicklungszyklen für einzelne Elemente bei gleichzeitiger Steigerung der Validität der Ergebnisse durch permanente Tests.

Virtuelle Testumgebungen, zum Beispiel mithilfe eines Hardware-in-the-Loop-Simulators (HILS), reduzieren dabei Komplexität und damit Kosten und Sicherheitsrisiken. HMI werden daher nicht mehr ausschließlich in Testfahrzeugen verbaut und manuell getestet, sondern kommen auch beim HIL-Verfahren zum Einsatz. Dieses Verfahren wird bei der Entwicklung und dem Testen komplizierter eingebetteter Systeme angewendet.

Im vorliegenden Fall wird ein HMI physisch in den HIL-Simulator integriert. Damit stellt das HMI das Embedded System Under Test (SUT) dar. Der HILS stellt sämtliche Netzwerksignale, wie zum Beispiel elektrische Signale, bereit. Somit simuliert der HILS die Interaktion sämtlicher elektronischer Steuerelemente mit dem HMI, indem er diesem suggeriert, dass er an ein echtes Steuergerät angeschlossen ist. Dies ermöglicht das Testen beliebig vieler elektronischer Steuerelemente, die im Echtbetrieb elektrische Signale an das HMI senden. Sämtliche Antworten werden vom HMI aufgezeichnet und anhand der Spezifikation getestet. Der grundsätzliche Aufbau dieser Tests ist in **Abbildung 1** dargestellt.

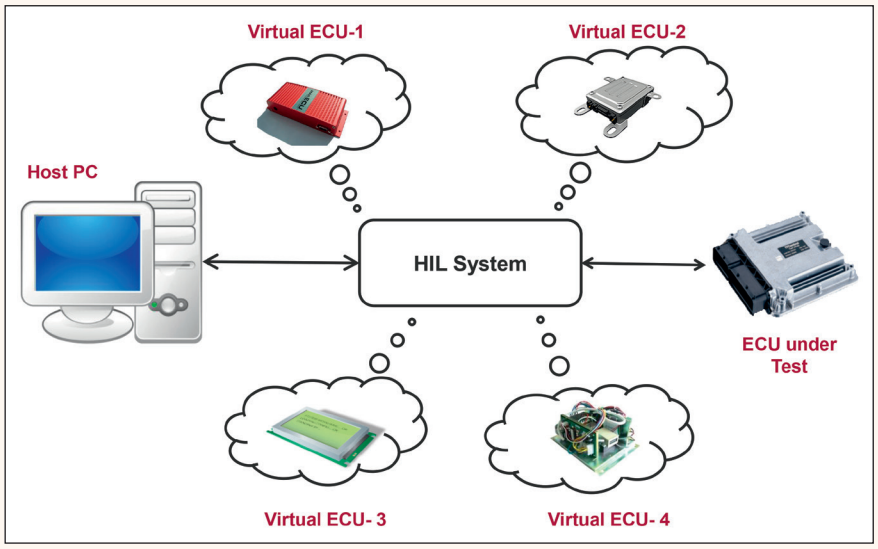


Abb. 1: Darstellung HILS

Beispiel der Automatisierung von Testreihen bei einem deutschen Automobilhersteller

CGI bietet zur Automatisierung von Testreihen eine HILS für Steuergeräte, die auf der Plattform des Kunden ausgeführt wird. Diese Software kombiniert alle Simulationen, Diagnosen und Umgebungen der elektronischen Steuergeräte. Einmalige Sequenzen sind reproduzierbar und werden für die nächsten Entwicklungsschritte, zum Beispiel für Regressionstests, weiterverwendet. Hinzu kommt die Möglichkeit, die Endbenutzerfunktionalität durch intuitiv zu bedienende, grafische Benutzeroberflächen für Diagnosetests zu testen.

Um die Funktion manuell zu bedienender Steuerelemente, wie zum Beispiel Knöpfe oder Schalter, zusätzlich zu überprüfen, wird eine externe, bereits vorhandene Robotersteuerung derart integriert, dass Steuerbefehle direkt übermittelt werden können. Sobald der Zugriff auf den Prüfstand sichergestellt ist, führt die Robotersteuerung die Schaltelemente manuell aus, indem sie den entsprechenden Knopf oder Schalter betätigt. Damit werden manuelle Testschritte, wie sie üblicherweise ein menschlicher Tester ausführt, imitiert.

Anhand eines komplexen Anwendungsszenarios, bei dem mehrere Computerplattformen miteinander vernetzt sind, kann jetzt ein Gesamtdurchlauf gestartet werden. Im folgenden Testszenario setzt sich die Architektur aus einer mobilen Umgebung (Smartphone), einem Backend-System (Webservices) und einer Middleware (Fahrzeug-Software) zusammen.

Anwendungsfall: Kurz vor Arbeitsschluss sucht der Fahrer eines PKW mit seinem Mobiltelefon nach einem Restaurant, in dem er zu Abend essen will. Nachdem er ein Restaurant gefunden hat, sendet er die Adresse des Restaurants an sein Auto, um beim Einsteigen direkt die Navigation zum Restaurant starten zu können. Dieses Szenario wird weiter in zwei Varianten unterteilt, das heißt, es werden die beiden gängigsten mobilen Betriebssysteme Android und iOS getestet. Neben den beiden Betriebssystemen interagiert das HMI mit einer in der Cloud bereitgestellten Vehicle-to-X-Anwendung (V2X). **Abbildung 2** zeigt eine Übersicht der in diesem Test integrierten Plattformen.

Für diesen Test wird ein Prüfstand mit verbauten Steuergeräten, die sogenannte Test Bench, eingesetzt. Mithilfe der Test Bench wird das Infotainment-System des PKW simuliert. Gleichzeitig ermöglicht dieser Aufbau die Überprüfung und Validierung der Webservices. Nachdem der Fahrer seine Suche erfolgreich abgeschlossen hat, beginnt schließlich der automatisierte E2E-Test mit den folgenden Prozessschritten:

- Fahrer meldet sich im Webportal des Automobilherstellers mit seinem Account an.
- Fahrer wählt sein Fahrzeug aus.
- Fahrer navigiert zu „Nachrichten".
- Fahrer überträgt die Adresse des Restaurants.
- Fahrer übermittelt die Nachricht.
- Test Bench empfängt die Nachricht und zeigt sie an.

Abbildung 3 zeigt eine schematische Darstellung des Szenarios.

Eine Agile Tool Chain macht den Unterschied

Dem automatisierten Testen bei der Softwareentwicklung kommt eine bedeutende Rolle zu. Die Herausforderung ist, Tests wiederholbar und effizient zu gestalten. Erst

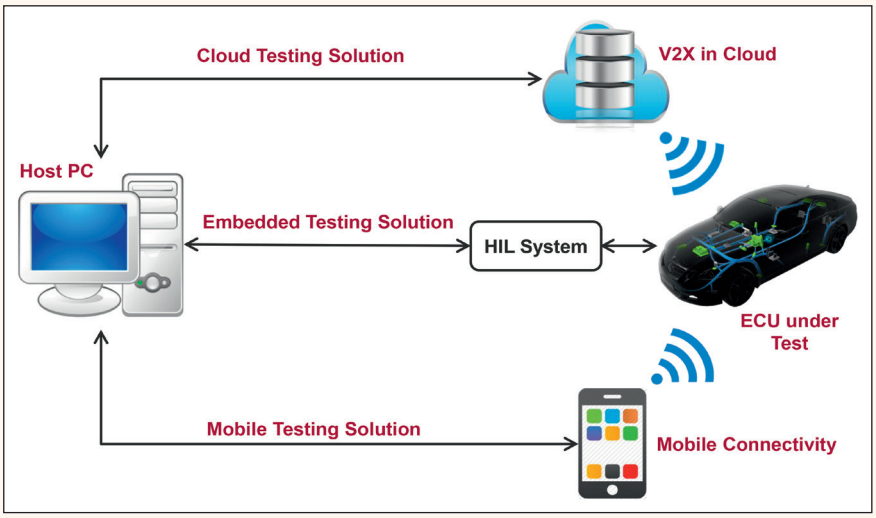


Abb. 2: Übersicht der integrierten Plattformen

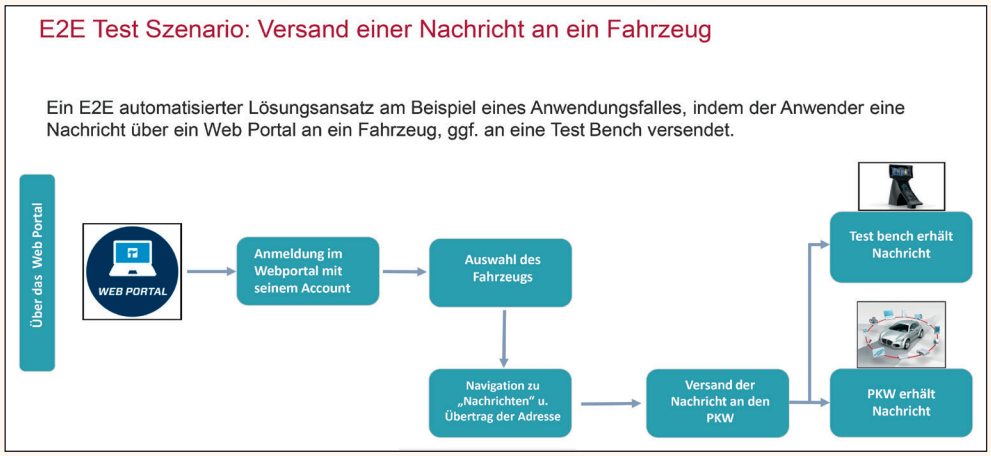


Abb. 3: Schematische Darstellung des Testprozesses

mithilfe einer Agile Tool Chain und den entsprechenden Komponenten kann ein Gesamtdurchlauf erfolgreich simuliert werden.

Zur automatischen Ausführung von Testfällen wird Jenkins in einer Master-Slave-Konfiguration verwendet, da die Test-Skripte von den Hardware-Treibern der elektronischen Steuerelemente auf den lokalen Maschinen abhängig sind. Die Konfiguration erfolgt auf dem Master. **Abbildung 4** zeigt den vollständigen E2E-Testfluss.

Nachfolgend wird die Integration des Testprozesses in eine Agile Tool Chain unter Angabe der jeweiligen Programmiersprache in Klammern beschrieben:

- Jenkins holt sich von Bitbucket den Code und die Jenkins-Datei.
- Die Jenkins-Datei initiiert die Ausführung einer Batch-Datei auf der lokalen Maschine, die an die Test Bench angeschlossen ist.
- Maven startet automatisch Tests auf dem Webportal mit Selenium (Java) im Testframework cucumber (Gherkin) und HILS-Testautomatisierung (Python). Die Bearbeitung der Feature-Dateien für cucumber erfolgt in Xray, einem Plug-in für Jira zur Umsetzung von Testaufgaben.
- Testautomatisierung ermöglicht den Zugriff auf die Elemente des Prüfstandes.
- Die Ergebnisse werden verglichen.
- Der Report wird in Jenkins angezeigt.

Abbildung 5 zeigt eine vollständige Übersicht der Architektur und aller Komponenten zur Unterstützung des E2E-Testszenarios.

Die drei größten Herausforderungen im Überblick

Die Einbindung von Jenkins Master Slave

Die E2E-Testkette technisch mit dem CI-Server Jenkins aufzusetzen und dabei die Testfälle auf einem Server durchzuführen, stellte sich als eine der größten Herausforderungen heraus. Im hier beschriebenen Anwendungsfall war wegen der Abhängigkeiten der vorhandenen Hardware eine standarisierte Anwendung nicht möglich. Nach eingehenden Prüfungen stellte sich die Implementierung von Jenkins als Master/Slave als die am besten geeignete Herangehensweise heraus. Nach intensiven Tests konnten alle Anforderungen des Kunden erfüllt werden.

Die Zusammenführung von unterschiedlichen Programmierumgebungen

Eine weitere Herausforderung des Projektes war die Zusammenführung der Testframeworks mit den unterschiedlichen Programmiersprachen. Das Testframework Selenium verwendet Java, während das Framework für die Test Bench auf der Basis von Python arbeitet. Die optimale Lösung war es in diesem Fall, in der Java-Umgebung Python als Prozess aufzurufen. Dadurch konnte eine Echtzeit-Kommunikation in den beiden Sprachen ermöglicht werden.

Die Kooperation von allen Stakeholder

Neben der Technologie ist bei der Transformation hin zu neuen Methoden und agilen Vorgehensweisen der menschliche Faktor ein entscheidender. Anders als in sichtbaren, technischen Evolutionen, wie beispielsweise der Einführung der Glühbirne von Edison, sind die Auswirkungen und Vorteile von veränderten Prozessen und Methoden nicht immer sofort erkennbar. Altbewährtes wird ungern infrage gestellt. Workshops und gezielte Trainings haben geholfen, Synergieeffekte aus den zurückhaltenden Skeptikern und begeisterten „First Movers" zu bilden und eine kooperative Zusammenarbeit zu fördern.

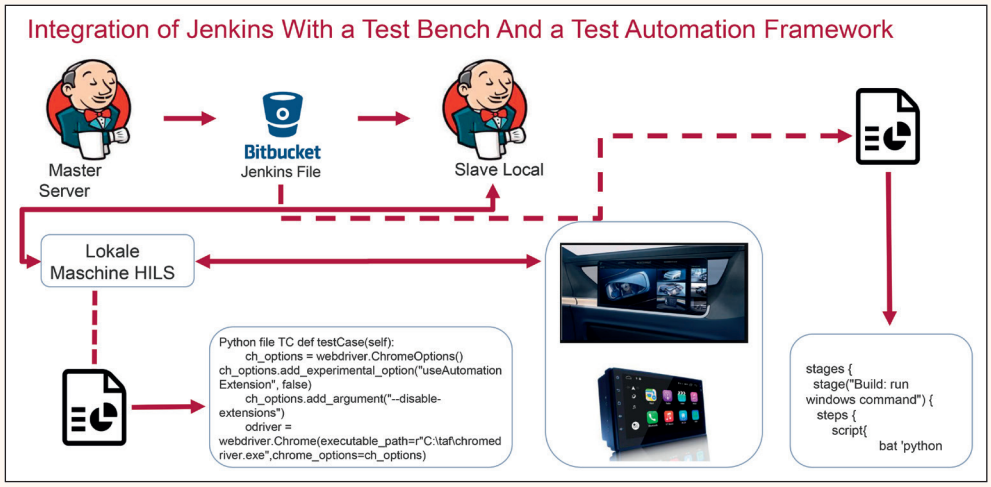


Abb. 4: E2E-Testfluss

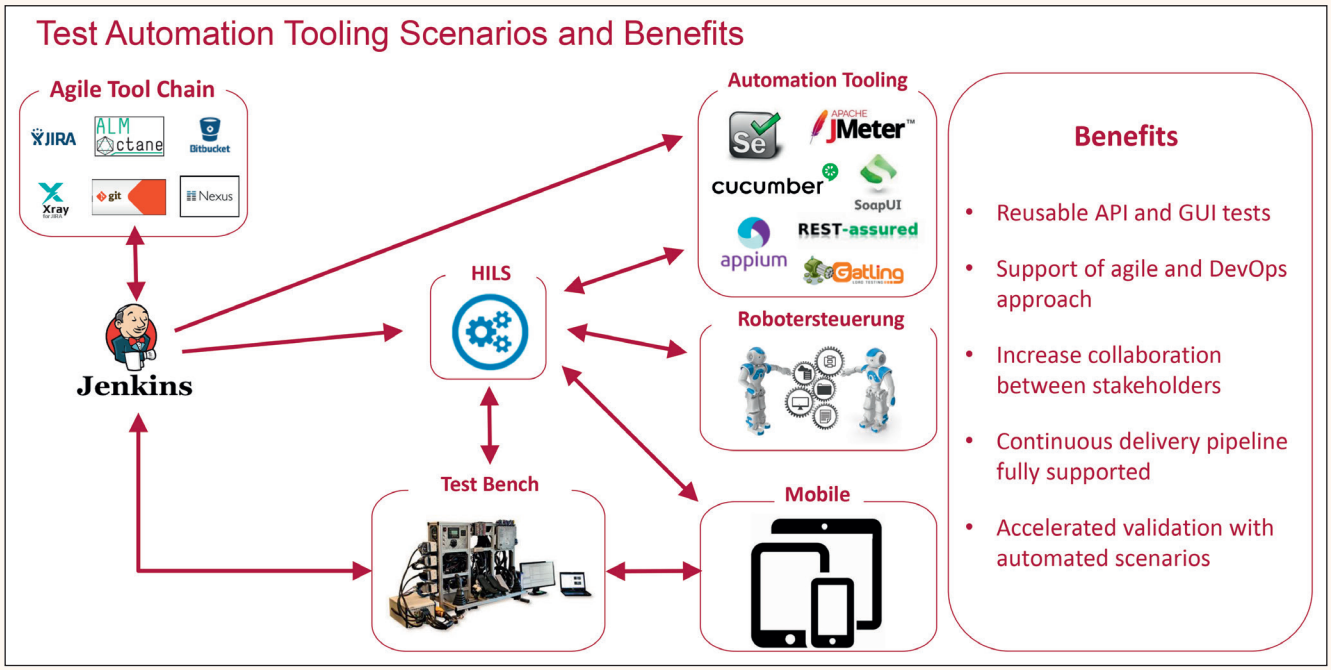


Abb. 5: Architektur & Komponenten für E2E-Testszenario

Projektfazit

Der Einsatz des HILS mit einem in den Prüfstand eingebauten HMI ermöglicht mit der Integration der Testautomatisierung und der Agile Tool Chain erfolgreiche, automatische E2E-Testläufe. Bisher waren Testläufe weder automatisiert noch plattformübergreifend möglich. Nur mit automatisiertem End-to-End-Testen können jedoch nennenswerte Kosteneinsparungen erzielt und die Frequenz von Tests erhöht werden. Eine größere Anzahl von Testläufen und Varianten erhöht die Aussagefähigkeit der Testergebnisse. Diese können, einmal gespeichert, problemlos für eine fortlaufende Pipeline wiederverwendet werden.

Die Nutzung des HILS mit der Integration der Robotersteuerung macht zusätzliche manuelle Tests überflüssig. So werden nicht nur Testfahrzeuge eingespart, sondern auch menschliche Ressourcen. Automatisierte Tests können über Nacht durchgeführt werden und sparen so produktive Arbeitszeit für das Abspielen der Tests. Diese eingesparten Ressourcen können für andere Mehrwert bringende Aufgaben eingesetzt werden.

Ein weiterer Vorteil, der sich durch automatisiertes Testen ergibt, ist die neue Form der Zusammenarbeit innerhalb der funktions- und abteilungsübergreifenden Teams. Im Ergebnis sind nicht nur kürzere Entwicklungszyklen zu verzeichnen, sondern auch ein besserer Austausch von Fachwissen.

Vijesh Kumar Raju Bhupathi Raju
vijeshkumarraju.bhupathiraju@cgi.com
ist Experte für Test Automation. Ihn zeichnet seine weitreichende Erfahrung im „Hardware in the Loop Testing" im Automotive-IT-Umfeld aus.

Jürgen Schröppel
juergen.schroeppel@cgi.com
ist Diplom-Informatiker, Diplom-Mathematiker bei CGI. Sein Betätigungsfeld ist das Testmanagement, agiles Testvorgehen und die Testautomatisierung in den Branchen Automotive und Telekommunikation.

Anastasios Mousios
anastasios.mousios@cgi.com
ist Test-Ingenieur und arbeitet seit 2012 für CGI. Als erfahrener Testexperte zeichnet ihn seine weitreichende Erfahrung in der Entwicklung, Analyse und dem Testing im Automotive-IT-Umfeld aus.

Geetha Godala
geetha.godala@cgi.com
ist Expertin für Test Automation und Java. Sie verfügt über große Erfahrung im End-To-End-Testing im Einzelhandel und in der Logistikbranche.