



Software-Absicherung mit Continuous Testing

Bei der Absicherung von Steuergeräte-Software durch kontinuierliche Systemtests ermöglicht Continuous Testing sehr kurze Feedback-Schleifen zum aktuellen Entwicklungsstand, weil die Testergebnisse sehr schnell (innerhalb von Stunden oder wenigen Tagen) nach der Weiterentwicklung zur Verfügung stehen. Dadurch lässt sich die Entwicklungsgeschwindigkeit deutlich erhöhen.

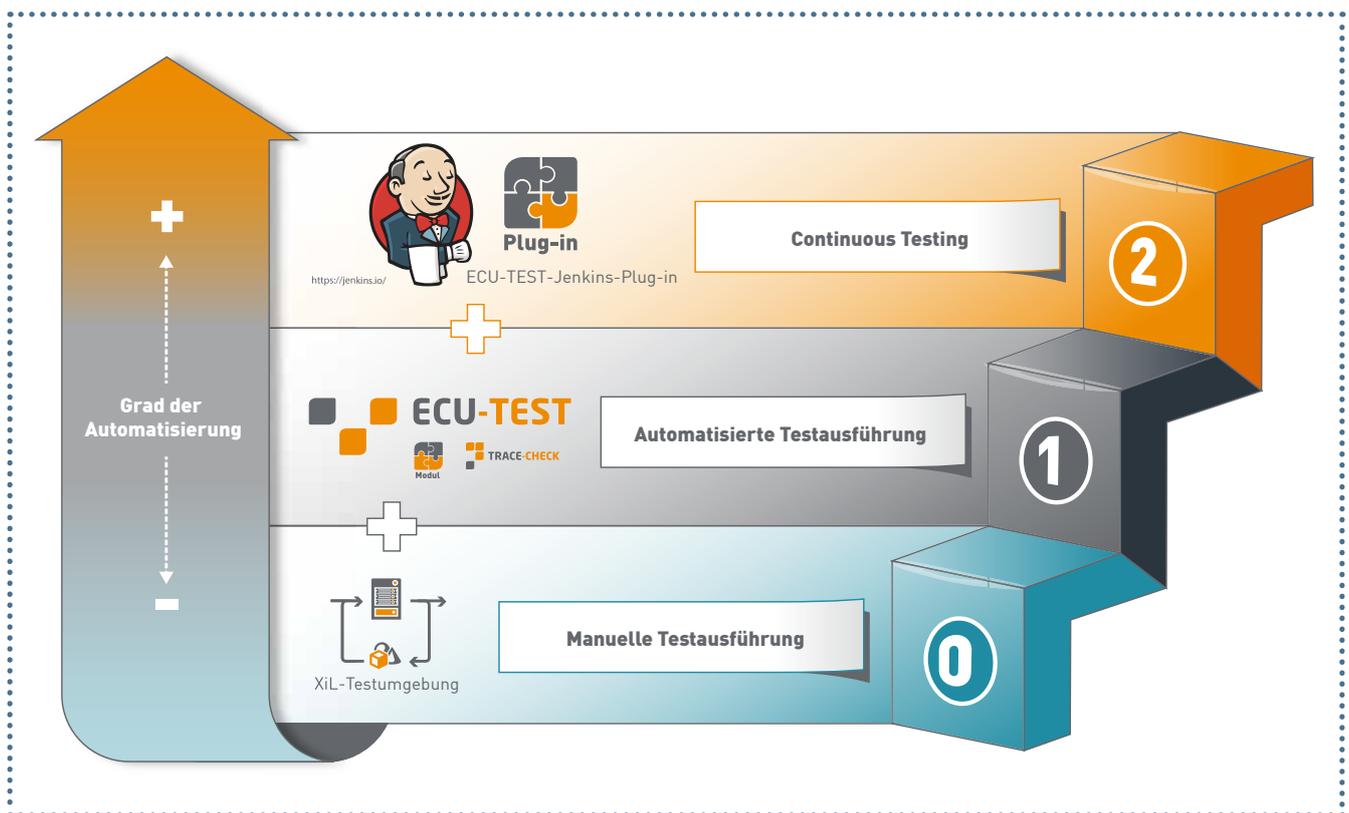


Bild 1: Die Stufen der Testautomatisierung und die eingesetzten Tools. (© TraceTronic)

Komplexe Systeme wie ein Steuergeräteverbund im Automobil erfordern eine strukturierte und solide Absicherung über den gesamten Entwicklungsprozess hinweg. Ein wichtiger Baustein dafür ist der Test einzelner Komponenten – bevor und nachdem diese integriert werden bzw. wurden. Dieser Test kann automatisiert werden, in der ersten Stufe durch den Einsatz ei-

ner entsprechenden Software. Eine solche Software unterstützt bei der intuitiven Testfallerstellung und führt die Testfälle automatisiert aus.

Die nächste Stufe der Automatisierung kann mit Continuous Testing erreicht werden. Hierbei werden entweder bei jeder Änderung der Software oder in definierten zeitlichen Abständen (z. B. jede Nacht) vollautomatisiert Tests

durchgeführt. Ein von der TraceTronic GmbH entwickeltes spezielles Plug-in versetzt den Automatisierungs-Server Jenkins in die Lage, die in der Testautomatisierungs-Software ECU-TEST implementierten Testfälle in der jeweiligen MiL-, SiL- oder HiL-Testumgebung koordiniert anzustoßen. So können komplexe Absicherungsprojekte auf Systemebene, bspw. in der Software

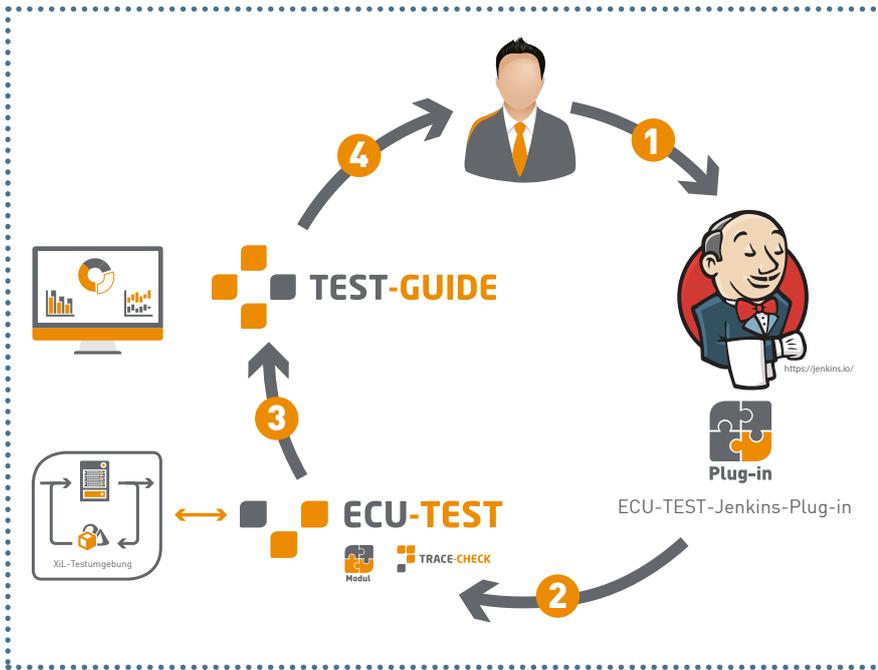


Bild 2: Der Continuous-Testing-Prozess mit TraceTronic-Tools. © TraceTronic

eines Motor- oder Getriebesteuergerätes, kontinuierlich ausgeführt werden.

Herausforderung

Eine umfangreiche Testautomatisierung ist bei der Absicherung von Steuergerätesoftware im Zuge der Automobil-Entwicklung heutzutage nicht mehr wegzu-denken. Der weitverbreitete Stand der Technik in der Automobilindustrie be-schränkt sich dabei jedoch meist auf die automatisierte Ausführung von Testfäl-len. Die Koordination der Testfälle so-wie die Vorbereitung der jeweiligen Testumgebung erfolgen oftmals noch manuell. Um der großen Vielfalt an Fahrzeugvarianten und den immer kür-zeren Entwicklungszyklen gerecht zu werden, ist eine weitergehende Auto-matisierung des Testprozesses jedoch unausweichlich. Wie könnte ein durch-gehend automatisierter Testprozess aussehen?

Stufe 0 der Testautoma-tisierung: Manuelles Testen

Für das Testen der Software von komplexen Systemen werden Testumgebungen eingesetzt, die die Simulation aller für den Test benötigten Kompo-nenten realisieren und notwendige Schnittstellen bereitstellen. In eine sol-che Testumgebung ist das Testobjekt eingebunden. Ein zur Testumgebung zugehöriges Programm stellt eine Be-

nutzeroberfläche bereit, mit deren Hilfe der Testingenieur seine Tests manuell ausführen kann.

Stufe 1: Einsatz einer Testautomatisierungs-Software

Die erste Stufe der Automatisierung des Absicherungsprozesses ist der Ein-satz einer Testautomatisierungs-Soft-

i ECU-TEST

Als integriertes Testautomatisierungs-Werkzeug zur Spezifikati-on, Implementierung, Ausführung und Dokumentation von Testfäl-len dient ECU-TEST der Absiche-rung komplexer technischer Pro-dukte durch Regressionstests. Die Software unterstützt eine breite Palette von Testwerkzeu-gen und kann in jeglichen Test-umgebungen (MiL/SiL/HiL/Fahr-zeug) angewendet werden. ECU-TEST zeichnet sich neben seiner intuitiven grafischen Bedienober-fläche durch die hohe Wieder-verwendbarkeit der Testfälle aus: Durch die Parametrierung, Struk-turierung und generische Testbe-schreibung sind die erzeugten Testfälle weitestgehend unabhän-gig von der spezifischen Testum-ggebungshard- und -software und somit in hohem Maße wiederver-wendbar.

ware, beispielsweise ECU-TEST. Die Software dient neben der Spezifikation, Implementierung, Dokumentation und Auswertung von Testfällen insbesonde-re der automatisierten Ausführung der-selben. Hierfür ist die Testumgebung über die jeweilige Konfiguration an die Testautomatisierungs-Software ange-bunden. Dabei steuert und beobachtet die Software das Testobjekt (beispiel-weise die Steuergeräte-Software) über die jeweiligen Programme der Testum-ggebung.

Die Testfälle in ECU-TEST bestehen aus einer Sequenz von Aktionen bzw. Testschritten, bspw. dem Lesen oder Bewerten einer Zustandsgröße des Testobjekts oder dem Manipulieren der Testumgebung. Ein Testfall wird als so-genannte Package-Datei gespeichert. Mehrere Packages können im Rahmen eines Projekts aneinandergereiht sowie für eine Ausführung in verschiedenen Konfigurationen parametrieret werden.

ECU-TEST führt die Testfälle mittels einer Ablauf-Maschine aus, die die Test-schritte abarbeitet und die Ergeb-nisse mit den spezifizierten Erwartungshal-tungen vergleicht. Daraufhin erhalten die Testschritte unabhängig voneinan-der eine Bewertung. Die Kombination aller Bewertungen ergibt das Tester-gebnis für den gesamten Testfall. Nach der Testausführung kann sich der An-wender die Daten anzeigen lassen oder Report-Dokumente (z. B. im HTML-For-mat) daraus generieren. Um den Test-prozess zu automatisieren, ist die An-wendung einer solchen Software die erste Stufe. Mit Continuous Testing lässt sich der Automatisierungsgrad weiter steigern.

Stufe 2: Continuous Testing

Ein Jenkins-Automatisierungs-Server hebt die Automatisierung auf die nächs-te Stufe. Jenkins ist ein Open-Source-Software-System, das im Master-Slave-Prinzip mit den ange-bundenen Res-sourcen kommuniziert und ursprüng-lich für die Automatisierung der Soft-waregenerierung aus Programmcode entwickelt wurde. Über Plug-ins kann der Funktionsumfang jedoch beliebig erweitert und Jenkins so auch für ande-re Aufgaben befähigt werden. Ange- »

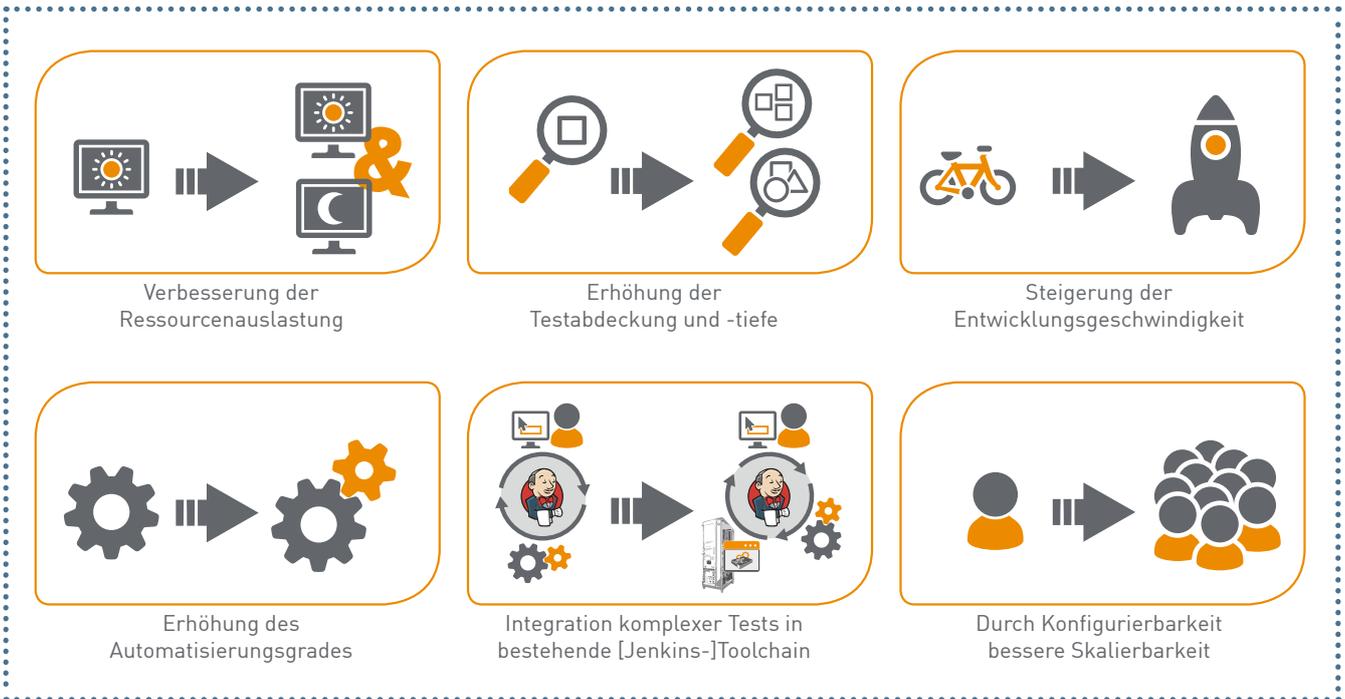


Bild 3: Nutzen und Vorteile von Continuous Testing. (© TraceTronic)

bundene Ressourcen sind dabei z. B. die Steuerrechner einer Hardware-in-the-Loop-Testumgebung, auf denen über Jenkins in ECU-TEST entsprechende Testfälle ausgeführt werden.

Die Verknüpfung von ECU-TEST und Jenkins erfolgt über ein von TraceTronic entwickeltes Plug-in, das auf dem Jenkins-Server installiert wird und ECU-TEST ansteuert. Es ermöglicht die Ausführung von ECU-TEST-Testfällen mit der zugehörigen Konfiguration und gewährt eine erste Übersicht über die Testergebnisse. In Bild1 sind die drei genannten Stufen der Testautomatisierung noch einmal zusammenfassend veranschaulicht.

Der Continuous-Testing-Prozess

Wenn alle technischen Voraussetzungen erfüllt und die vorgestellten Software-Werkzeuge installiert und eingerichtet sind, kann das kontinuierliche Testen beginnen. Wie aber sieht der Testprozess genau aus? Prinzipiell ist dieser immer vom konkreten Anwender abhängig und der Prozess muss an die jeweiligen Gegebenheiten angepasst werden. Dennoch soll hier ein möglicher Ansatzpunkt gezeigt werden, der jedoch keinesfalls verbindlich oder alleingültig ist. Bild2 zeigt den Ansatz des Continuous-Testing-Prozesses von TraceTronic.

Der Testingenieur legt in einem Jenkins-Job fest, zu welcher Zeit oder zu welchem Ereignis ein jeweiliger Test ausgeführt werden soll. Je nach Konfiguration kann daraufhin über den Jenkins-Automatisierungs-Server automatisiert die Testumgebung mit dem aktuellen Programmstand bzw. dem Testobjekt eingerichtet werden. Anschließend stößt Jenkins zur vorgegebenen Zeit oder zum definierten Ereignis den Test an, lädt die Konfiguration und lässt ECU-TEST die Testschritte abarbeiten. Nach der Testdurchführung und der Bewertung der Testergebnisse durch ECU-TEST gibt das Plug-in Rückmeldung zum Beispiel in Form

- eines verlinkten ECU-TEST-Reports,
- einer Trend-Darstellung der Testergebnisse über JUnit-Reportgenerierung,
- einer Anbindung an TEST-GUIDE zur Reportverwaltung oder
- eines Parsings der ECU-TEST-Logs nach Warn- und Fehlermeldungen.

Während des Testdurchlaufs kommuniziert die Testautomatisierungs-Software ECU-TEST stetig mit dem Automatisierungs-Server Jenkins. Dadurch ist es möglich, für sonst manuell auszuführende Schritte Automatismen zu implementieren oder automatisiert in definierte Rückfall-Ebenen zu springen, falls es während der Testausführung zu Problemen kommt.

Jenkins erlaubt darüber hinaus die Koordination der Trace-Analyse. TRACE-

CHECK ist ein leistungsstarkes Werkzeug zur automatischen Auswertung und Validierung aufgezeichneter Messdaten und verfügt über eine nahtlose Anbindung an ECU-TEST. Es dient der Auswertung großer Datenbestände verschiedenster Aufzeichnungsformate. Dabei werden Trace-Analysen modular aus einzelnen, parametrierbaren Bausteinen kombiniert. Durch eine generische Analyse-Beschreibung sind die erzeugten Analysen unabhängig von der spezifischen Hard- und Software zur Messaufzeichnung und somit in hohem Maß wiederverwendbar.

Das Jenkins-Plug-in ermöglicht es, aus Jenkins heraus ECU-TEST zu starten und zu beenden, über ECU-TEST Testfälle (als Packages oder Projekte) auszuführen und Reports in das Standardformat ASAM ATX zu konvertieren sowie diese optional an TEST-GUIDE zu übertragen. TEST-GUIDE ist eine in verschiedene Entwicklungs- und Test-Prozesslandschaften flexibel integrierbare, zentrale Datenbankanwendung. Da sie speziell für den Einsatz im Automotive-Bereich mit einer ASAM-ATX-Standardkonformen Schnittstelle entwickelt wurde, ist es mit dieser Software möglich, Testdurchführungen und deren Ergebnisse intuitiv per Browser zu verwalten, darzustellen und zu analysieren. TEST-GUIDE stellt somit einen möglichen Rückkanal für den Testingenieur dar (4).

Vorteil

Bei der Absicherung von Steuergeräte-Software durch kontinuierliche Systemtests ermöglicht Continuous Testing sehr kurze Feedback-Schleifen zum aktuellen Entwicklungsstand, weil die Testergebnisse sehr schnell (innerhalb von Stunden oder wenigen Tagen) nach der Weiterentwicklung zur Verfügung stehen. Dadurch lässt sich die Entwicklungsgeschwindigkeit deutlich erhöhen: Der höhere Automatisierungsgrad – von der automatisierten Testdurchführung hin zum, mithilfe eines Jenkins-Servers umfassend koordinierten, automatisierten Test – ermöglicht eine deutlich verbesserte Ressourcen-Auslastung. Außerdem erfolgt die optimale Nutzung der Prüfplätze, da diese bei Nichtbenutzung immer mit automatisierten Tests ausgelastet werden können. Continuous Testing kann die Testabdeckung und Testtiefe erheblich erweitern, da z. B. über Nacht zusätzliche Testfälle ausgeführt oder weitere Varianten getestet werden können (Bild 3).

Sofern bereits eine Jenkins-Toolchain in einem Continuous-Integration-Entwicklungsprozess etabliert ist, kann diese mit Hilfe des Plug-ins und von ECU-TEST für komplexe Systemtests erweitert werden. Ein intelligentes Rechte-Rollenmodell und eine entsprechende Konfiguration machen es möglich, dass auch Anwender, die nicht über fundiertes Fachwissen im Bereich der Testautomatisierung verfügen, die Testdurchführung starten können.

Anwendungsbeispiele

Das Zusammenspiel von ECU-TEST mit Jenkins ist bei Automobilherstellern und -Zulieferern mit verschiedenen Einsatzschwerpunkten bereits erfolgreich etabliert. Hier einige konkrete Anwendungsbeispiele:

- Integration von HiL-Tests in eine Jenkins-CI-Umgebung: Dadurch konnten die Feedback-Schleifen bei der Softwareentwicklung eines Antriebsteuergeräts erheblich verkürzt und die HiL-Prüfstandsauslastung durch die Nutzung von Nightly-Tests maßgeblich verbessert werden.

- Einsatz von Jenkins zum Test mehrerer Softwarestände über Nacht: Dadurch konnte der Konfigurationsaufwand für den Nutzer deutlich reduziert werden. Die HiL-Prüfstandsauslastung wurde optimiert und Fehlerquellen wurden dank der Vermeidung manueller Schritte verringert.
- Ergänzung einer bestehenden Jenkins-Toolkette um dynamische HiL-Tests auf Systemebene für den Software-Build und statische Codeanalysen: Dadurch wurde sowohl schnelles Feedback als auch ein einfaches Konfigurieren und Skalieren der Tests ermöglicht.

TraceTronic sieht erhebliches Potenzial für eine weitere Steigerung des Automatisierungsgrads mit Continuous Testing. Einerseits lässt sich die Auslastung der Prüfstände durch eine zeitliche und örtliche Entkopplung der Trace-Analyse von der Testausführung signifikant erhöhen. Andererseits kann eine noch engere Verzahnung eines Testmanagementwerkzeugs mit einem Testausführungswerkzeug über den Continuous-Testing-Ansatz den Automatisierungsgrad deutlich verbessern.

Im Bereich automatisierter Softwaretests für Kfz-Steuergeräte wird durch Continuous Testing die nächste Stufe der Automatisierung erreicht. Neben der automatisierten Durchführung der Testfälle ist nun auch deren automatisierte und verbesserte Koordination möglich. ■ (oe)

» www.tracetronic.de

» www.hanser-automotive.de/4059751

Hier finden Sie die Download-Version des Beitrags.



Dipl.-Ing. Frederik Schrader

arbeitet als Testingenieur bei der TraceTronic GmbH. Seit einigen Jahren ist er Spezialist im Bereich Continuous Testing mit HiL-Systemen und leitet ein Industriekundenprojekt in diesem Bereich.



Christian Pönisch M.Sc.

arbeitet als Software-Entwickler bei der TraceTronic GmbH in Dresden. Mit seiner langjährigen Erfahrung im Bereich Continuous Integration, insbesondere Jenkins, unterstützt er mehrere automobilmotivierten Kundenprojekte zur Steigerung des Testautomatisierungsgrads.