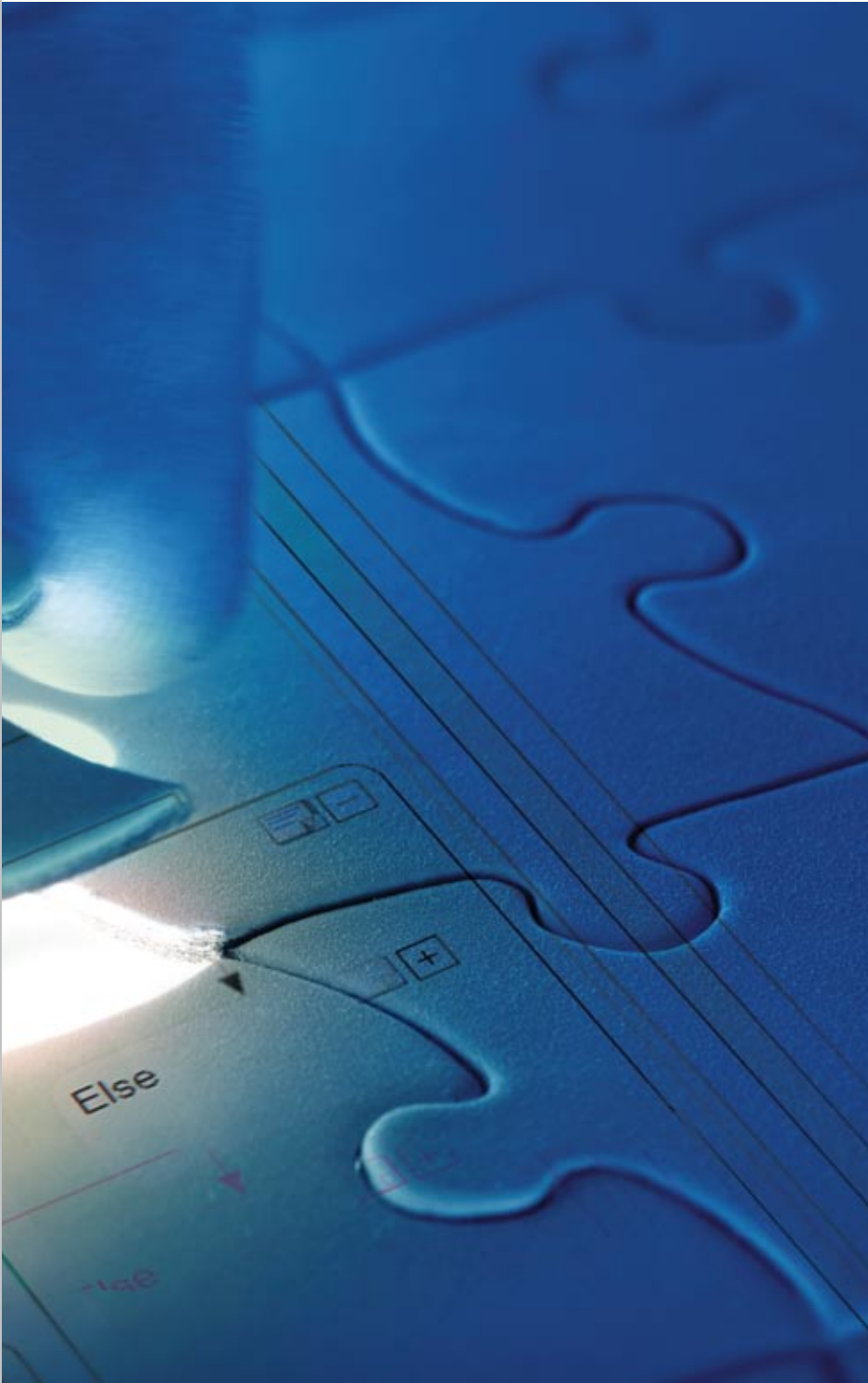


Test gut, alles gut.

Mit strukturierter Testorganisation die Vorteile des Simulators vollständig nutzen



Ein Horrorszenario für den Testmanager: Die Hardware-in-the-Loop (HIL)-Simulatoren sind in Betrieb und bringen nicht die gewünschte Arbeitserleichterung. Was nutzt einem dann die ganze Test-Hardware? Hier schaffen eine gute Strategie und eine gute Organisation Abhilfe. dSPACE bietet den Anwendern neben maßgeschneiderten HIL-Simulatoren nicht nur die Testautomatisierungssoftware AutomationDesk, sondern auch Unterstützung beim Aufbau großer, komplexer Testumgebungen. Dies bildet die Grundlage für einen langfristig effizienten Einsatz der HIL-Simulatoren.

HIL-Simulation in der Automobilindustrie

Während vor einigen Jahren die Hardware-in-the-Loop (HIL)-Simulation lediglich eine weitere Testmethode für einzelne Steuergeräte oder Steuergeräteverbunde darstellte, ist sie heute integraler Bestandteil des Software-Freigabeprozesses. Ihr Einsatz reicht vom Einzelkomponententest bis hin zum Verbundtest großer und größter Systeme und umfasst alle Domänen, darunter den Motor-, Getriebe-, Fahrdynamik-, Chassis-, Infotainment- und Komfortbereich.

Zwar wird der Nutzen der HIL-Simulation heute nicht mehr in Frage gestellt, doch um die Testsysteme zeit- und kosteneffizient einzusetzen, bedarf es neben dem HIL-System selbst auch einer durchdachten Teststrategie und -organisation.

Den Simulator effizient nutzen

Die große Anzahl der neuen, zu testenden Funktion, die steigende Komplexität der Steuergeräte und

Testautomatisierung ist als langfristige Aufgabe anzusehen, die sogar eine längere Lebensdauer hat als der Simulator, auf dem sie ursprünglich implementiert wurde.

nicht zuletzt die fast unüberschaubare Variantenvielfalt führen zu einer Vielzahl von Testfällen. Um diese Testvielfalt beherrschbar zu machen, kommt die Testautomatisierung als Mittel der Wahl ins Spiel. Mit ihr können einmal definierte Testabläufe automatisiert über Nacht oder am Wochenende laufen. Aus erstellten Testfällen baut der Anwender nach und nach hochwertige Bibliotheken auf, aus denen immer neue Tests und Testvarianten erstellt werden. Ziel ist es dabei, diese Bibliothekselemente auch für nachfolgende Fahrzeuggenerationen möglichst unverändert weiter zu verwenden. Daher ist die Testautomatisierung von Anfang an als langfristige Aufgabe anzusehen, die sogar eine längere Lebensdauer hat als der Simulator, auf

dem sie ursprünglich implementiert wurde.

Ordnung von Anfang an

Eine Testautomatisierungssoftware unterstützt den Anwender bei der Testimplementierung und Testausführung. Die konkrete Testentwicklung kann sie nicht ersetzen, dies muss der Testentwickler weiterhin selber leisten. Sie kann allerdings den Anwender unterstützen, von Anfang an strukturiert zu arbeiten und die erstellten Basisfunktionsblöcke in einer Bibliothek zu verwalten. In AutomationDesk, der Testautomatisierungssoftware von dSPACE, werden Testfälle mit Hilfe von grafischen Elementen aufgebaut, so dass der Einstieg erleichtert wird. So erstellte Basisblöcke können sehr

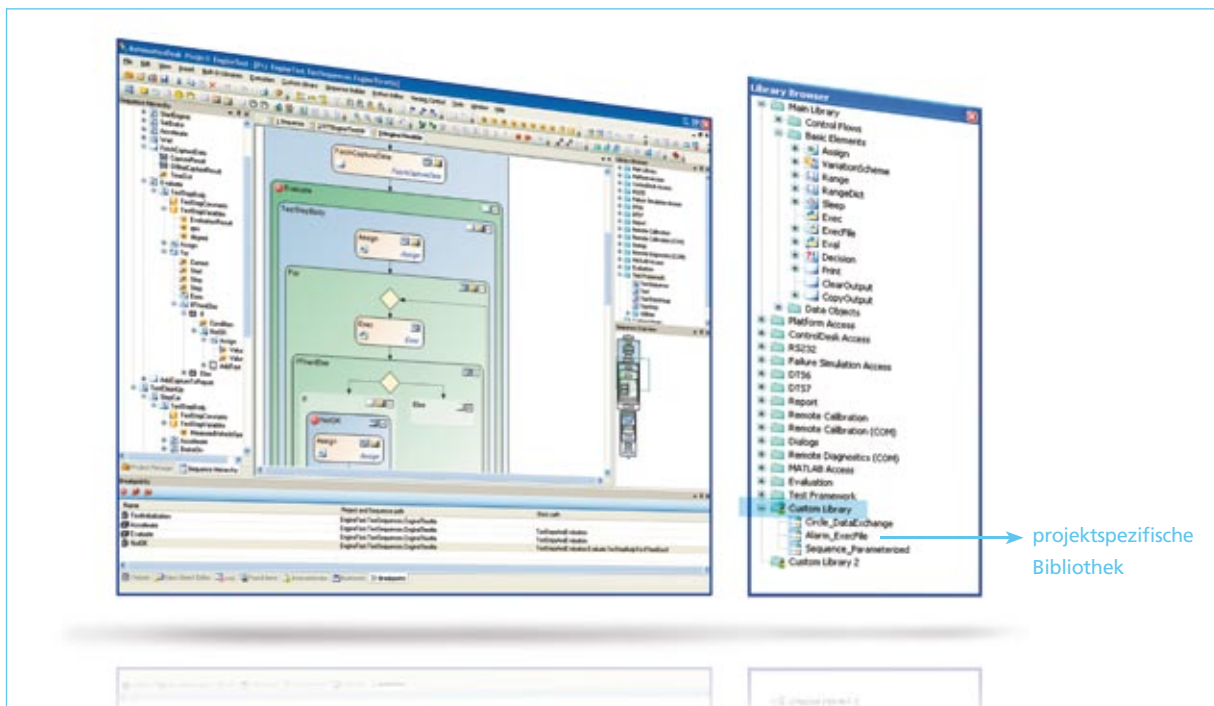


Abbildung 1: AutomationDesk unterstützt den Aufbau projektspezifischer Bibliotheken, beispielsweise mit so genannten CustomLibraries (links). Die einzelnen Elemente aus den Testfällen lassen sich per Drag&Drop der Bibliothek hinzufügen (rechts).

einfach in Bibliotheken gespeichert und wiederverwendet werden (Abbildung 1).

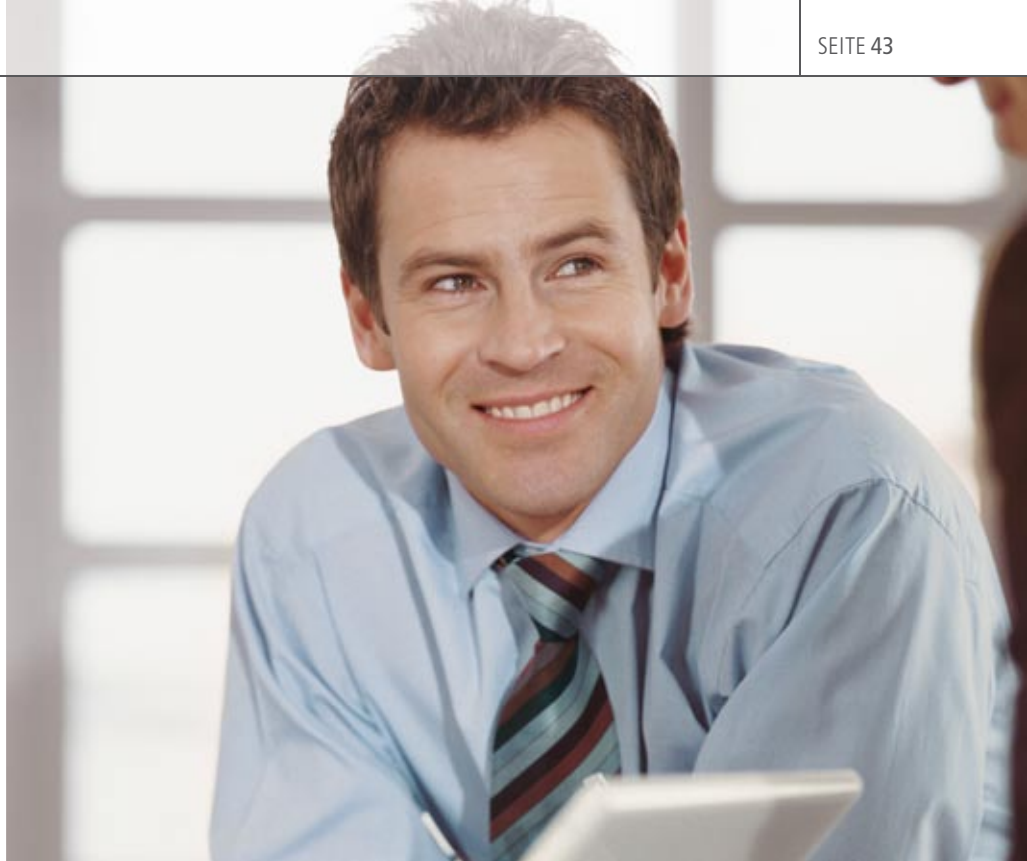
Veränderungen einplanen

Gerade beim ersten Einsatz einer Testautomatisierungssoftware gibt es Aspekte, die oftmals nicht ausreichend berücksichtigt werden. So sollten bei den Testfällen von Anfang an mögliche zukünftige Entwicklungen mit einbezogen werden, beispielsweise zusätzliche Fahrzeugreihen oder wachsende HIL-Systeme. Auch die Erweiterung des Testteam auf mehr Mitarbeiter sollte beim Aufbau der Testgruppenstruktur mit bedacht werden.

Wiederverwendung von Testfällen

Der Wunsch nach schnell verfügbaren Testergebnissen führt zu Beginn des Testens oftmals zur Vernachlässigung des Bibliothekaufbaus, was die Wiederverwendung der Tests stark erschwert. Dabei sind die meisten HIL-Tests so komplex und aufwendig, dass man sie nur mit einer klar definierten Organisation verwalten und wieder verwenden kann. Die Wiederverwendung eines implementierten Testablaufs beschränkt sich dabei nicht nur auf dessen wiederholte Verwendung in verschiedenen Testsets, zum Beispiel für Regressionstests. Diese Art der Wiederverwendung ist einfach zu realisieren und praktisch ohne Aufwand. Aufwändiger sind jedoch die weiteren Arten der Wiederverwendung, die andere ähnliche oder gar neue Komponenten mit einbeziehen:

1. Wiederverwendung eines generischen Tests für strukturell ähnliche Testfälle (zum Beispiel ein einziger generischer Fensterhebertest für alle Fenster, anstatt vier Tests für jedes einzelne Fenster)
2. Wiederverwendung von Basisblöcken auf demselben Simulator für verschiedene Testbereiche und -themen



3. Wiederverwendung von Basisblöcken und Tests auf unterschiedlichen Simulatoren, jedoch für das gleiche Fahrzeug
4. Wiederverwendung von Basisblöcken und Tests für ein neues Fahrzeug auf einem neuem oder demselben Simulator

Das Ziel dabei ist, durch Vermeidung von Redundanzen in der Bibliothek die Wartbarkeit zu verbessern und durch möglichst generische und allgemeingültige Basisblöcke deren Mehrfachnutzung überhaupt erst zu ermöglichen (Abbildung 2). Allgemeingültige und redundanzfreie

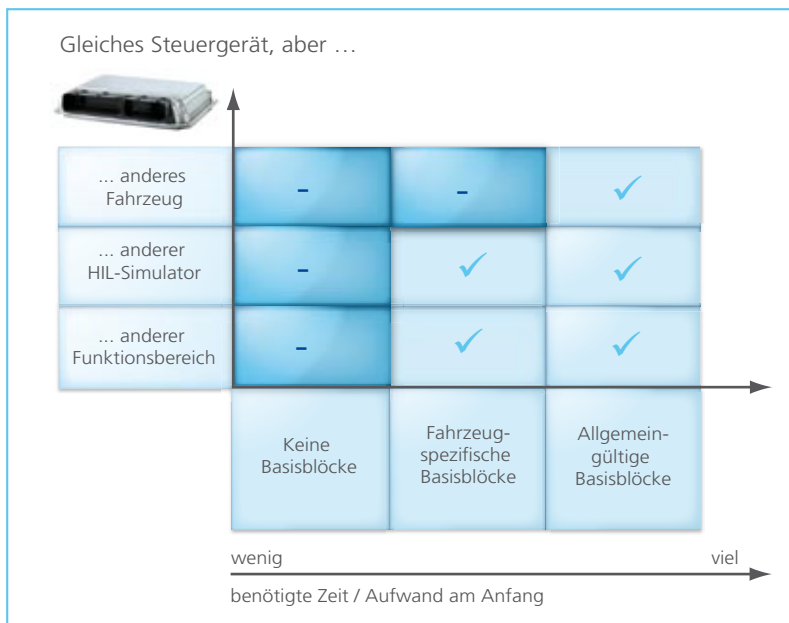


Abbildung 2: Der höhere Aufwand zu Beginn für die Erstellung von allgemeingültigen Basisblöcken vereinfacht im späteren Prozess die Testwiederverwendung.



Ein gut ausgebildetes Testteam, das sich gut mit dem Testsystem auskennt, ist eine wichtige Voraussetzung für einen erfolgreichen zukünftigen Einsatz.

Basisblöcke verbunden mit einem geeigneten Bibliothekskonzept stellen die Voraussetzung für eine Wiederverwendung von Testfällen dar, da in diesem Fall nötige Anpassungen, beispielsweise an geänderten Sensoren, nur an sehr wenigen Stellen Veränderungen nach sich ziehen. Hieraus abgeleitete Testfälle sind danach beinahe unmittelbar lauffähig. Werden die Tests ohne Bibliothekskonzept erstellt, müssen bei jeder Veränderung, zum Beispiel beim Einsatz eines neuen HIL-Simulators, sämtliche Tests überarbeitet werden (Abbildung 4). Auch wenn die Einführung einer tragfähigen Teststrategie und -organisation einen relativ hohen anfänglichen Aufwand bedeutet, stellen die entwickelten Bibliotheken das Fundament dar, um die HIL-Simulatoren langfristig effizient nutzen zu können.

Qualität als Basis ...

Selbst mit Software-Unterstützung ist die Erstellung, Verwaltung und

Organisation der Testfälle keinesfalls eine Aufgabe, die nebenher erledigt werden kann. Testerstellung ist vergleichbar mit Software-Ent-

wicklung – auch hier spielt die Architektur und die Versionierung eine große Rolle. Die Qualität der Arbeit, die am Anfang geleistet wird, beeinflusst im großen Maße, wie gut und mit wie viel Aufwand sich die Tests später wieder verwenden lassen. Ein gut ausgebildetes Testteam, das sich gut mit dem Testsystem auskennt, ist daher eine wichtige Voraussetzung für einen erfolgreichen zukünftigen Einsatz.

... schafft Vertrauen

Beim Testen mit dem HIL-Simulator muss sichergestellt sein, dass die während eines Testlaufs gefundenen Fehler sich auch wirklich im Steuergeräte-Code befinden und nicht etwa in den Tests selbst. Daher werden zu Beginn zunächst die Basisblöcke spezifiziert, implementiert und ausführlich getestet. Bei der Basisblockentwicklung gelten Qualitätsstandards, wie bei der Software-Entwicklung, beispielsweise eine klare Funktionsspezifikation, Reviews und offizielle Freigaben. Aus diesen Grundlagen entstehen die ersten Testfälle, die nach und

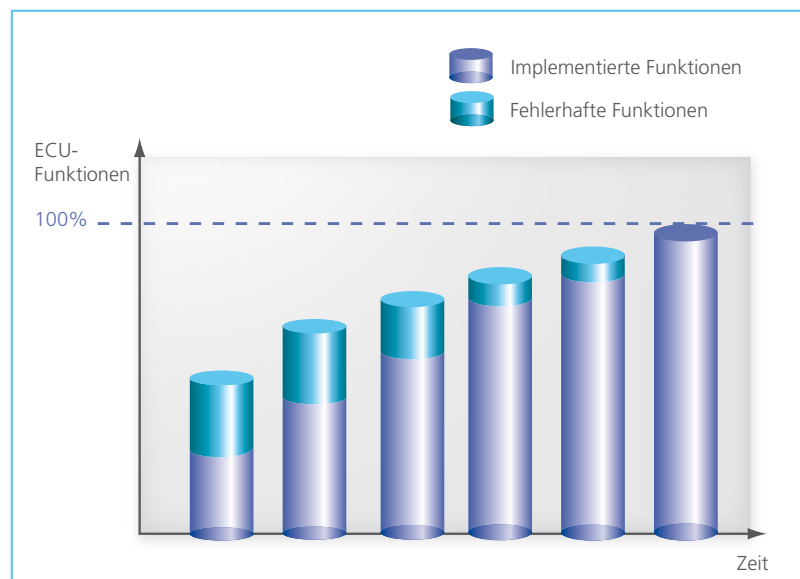


Abbildung 3: Durch den Einsatz von HIL-Simulation und Testautomatisierung steigert sich die Qualität der Steuergeräte-Funktionen kontinuierlich.

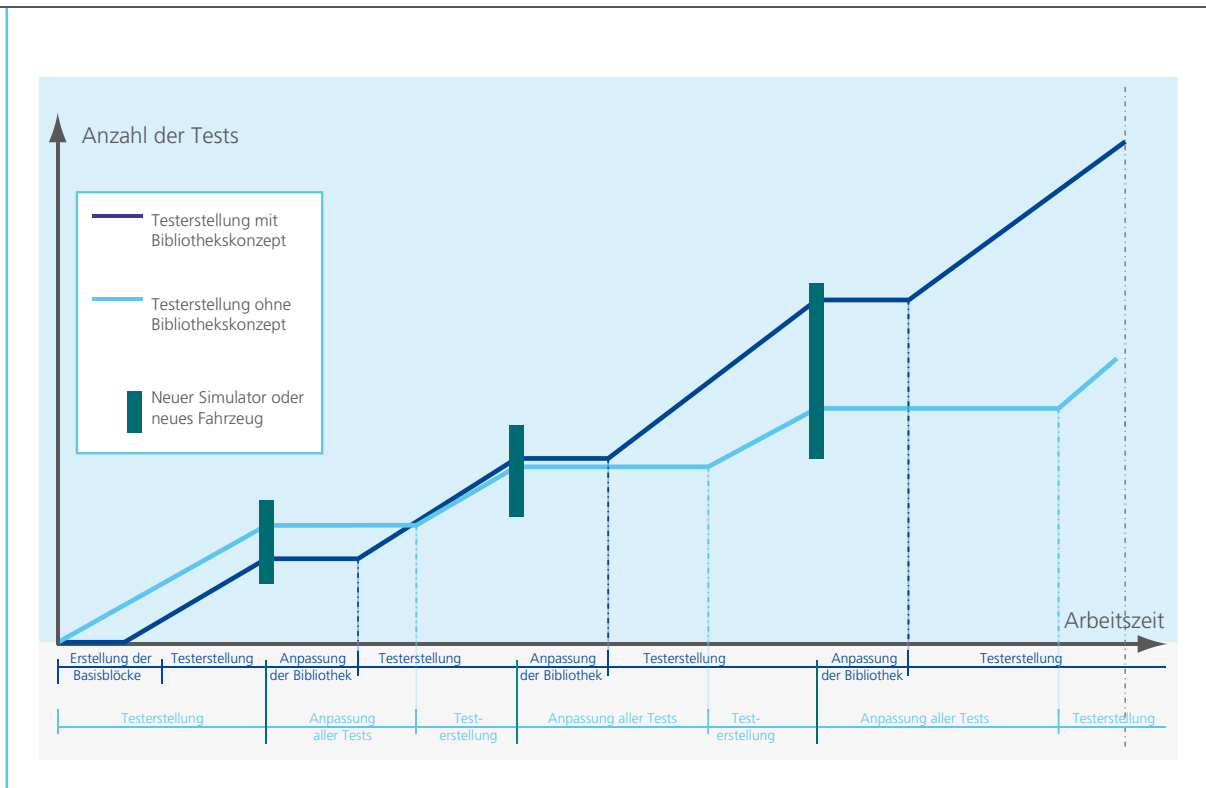


Abbildung 4: Mit Bibliothekskonzept müssen bei der Einführung eines neuen Simulators oder eines neuen Fahrzeugs lediglich die Basisbibliotheksblöcke angepasst werden.

nach erweitert und ebenfalls abgenommen und freigegeben werden. Somit kann man sicher sein, dass Fehler in den Steuergeräte-Funktionen selbst gefunden werden und damit die verantwortlichen Steuergeräteentwickler die HIL-Simulationsergebnisse als vertrauenswürdig einstufen (Abbildung 3).

dSPACE Schulungen und Engineering

Bei ihrer Anschaffung bedeuten der HIL-Simulator und die dazugehörige Software eine hohe Anfangsinvestition. Doch mit der Zeit überwiegt der Kostenanteil, der für ein Testteam auch ohne Simulator anfällt. Umso wichtiger ist die umfassende Einarbeitung der Testentwickler und -verwalter in die Testsoftware, so dass sie effizient genutzt werden kann. dSPACE bietet spezielle Schulungen für AutomationDesk und die Arbeit mit HIL-Simulatoren. Die Schulungen finden auf Wunsch auch direkt beim Kunden statt und gehen auf dessen konkrete Anforderungen und Bedürfnisse ein. Seit mehreren Jahren bietet dSPACE

seinen Kunden Unterstützung bei der Realisierung von Testaufgaben. Während des Testautomatisierungs-Start-Up-Engineerings profitiert der Anwender von den Projekterfahrungen und dem Hardware- und Software-Wissen der dSPACE-Ingenieure. Dazu gehören:

- Testprozesswissen
- Testwissen
- Projektdurchführung
- Resident-Ingenieure vor Ort
- Erstellung des Testtemplates
- Erstellung des Bibliothekskonzeptes
- Beispiel-Testimplementierung
- Daten- und Ergebnisverwaltung
- Einbindung von Drittanbieter-Software
- Anbindung der Testsoftware an die vorhandenen Werkzeuge

Das Wissen und die Erfahrung aus Engineeringprojekten lässt dSPACE in die Weiterentwicklung seiner Produkte einfließen, um sie den Anwenderbedürfnissen weiter anzupassen. Wir sind mit unseren Produkten nah am Markt und ge-

stalten Trends, wie zum Beispiel Standardisierungsformate, aktiv mit. Gleichzeitig profitiert der Anwender von den umfangreichen Erfahrungen, wenn es um den Aufbau seiner Testumgebung geht. ■

Fazit

- Testautomatisierung ist für die HIL-Simulation unentbehrlich.
- Strukturiertes Bibliothekskonzept vereinfacht die Testwiederverwendung.
- Anfangsaufwände zahlen sich langfristig aus.