# Test Automation mit VectorCAST während der gesamten Softwareentwicklung

## VectorCAST Produktfamilie

Die VectorCAST Produktfamilie automatisiert Testaktivitäten über den gesamten Software Entwicklungszyklus. Dieses Dokument soll Ihnen einen umfassenden Überblick darüber geben, was VectorCAST Produkte können, wie sie zusammenwirken und wie Sie diese Produkte optimal einsetzen können, damit sie Ihre Anforderungen an Softwaretests voll erfüllen.

Die VectorCAST Produktfamilie umfasst fünf sich ergänzende Werkzeuge:

| | |
|---|---|
| **VectorCAST/C++ VectorCAST/Ada** | *VectorCAST/C++ und VectorCAST/Ada zusammen automatisieren den Testprozess von Quellcode-Modulen, die in C/C++ oder Ada/Ada95 entwickelt worden sind. Sie unterstützen sowohl Modul- als auch Integrationstests und zeigen die zugehörige Codeabdeckung.* |
| **VectorCAST/Cover** | *Überprüft die Codeabdeckung während eines funktionalen Tests oder Systemtests. Die gewonnenen Abdeckungsdaten können auch von VectorCAST/C++ (VectorCAST/Ada) genutzt werden.* |
| **VectorCAST/RSP (Runtime Support Package)** | *Eine Erweiterung von VectorCAST/C++ (VectorCAST/Ada). Ermöglicht Ihnen die Ausführung von Testfällen für Echtzeitanwendungen in einer eingebetteten Ziel- oder Simulationsumgebung.* |
| **VectorCAST/Manage** | *Ermöglicht Ihnen den Import von bestehenden VectorCAST/C++ (VectorCAST/Ada) Testumgebungen in Regressions-Testreihen und stellt einen zentralen Kontrollpunkt für alle Modul- und Integrations-Regressionstest-Aktivitäten dar.* |
| **VectorCAST/Requirements Gateway** | *Erlaubt einen Datenaustausch zwischen einem Anforderungsmanagement-Werkzeug und den VectorCAST-Testwerkzeugen. Ein einfaches wie intuitives Interface ermöglicht das Anlegen von Assoziationen zwischen den Anforderungen und den Testfällen in VectorCAST.* |

*Abb. 1: Überblick über VectorCAST Produkte*

## VectorCAST/C++ und VectorCAST/Ada

VectorCAST/C++ (VectorCAST/Ada) automatisiert die wichtigsten Aktivitäten in Verbindung mit Modul- und Integrationstests. Diese Aktivitäten sind sonst oft sehr zeitraubend und teuer und werden dementsprechend häufig vernachlässigt.

### *Automatische Generierung eines ausführbaren Testrahmens auf der Basis einer oder mehrerer Quellcode-Dateien*

Wie in Abbildung 2 dargestellt, umfasst der von VectorCAST/C++ (VectorCAST/Ada) generierte ausführbare Testrahmen einen Testtreiber, die zu testende(n) Quelldatei(en), ggf. vom Anwender bestimmte Stubs für abhängige Funktionen sowie alle abhängigen Funktionen ohne Stubs. Der Testrahmen ist datengesteuert, was bedeutet, dass die Testdaten während der Ausführung vom Rahmen gelesen werden. Damit entfällt die Notwendigkeit, bei der Durchführung des gleichen Tests mit mehreren Testfällen jedes Mal einen neuen ausführbaren Rahmen zu kompilieren und zu verknüpfen.

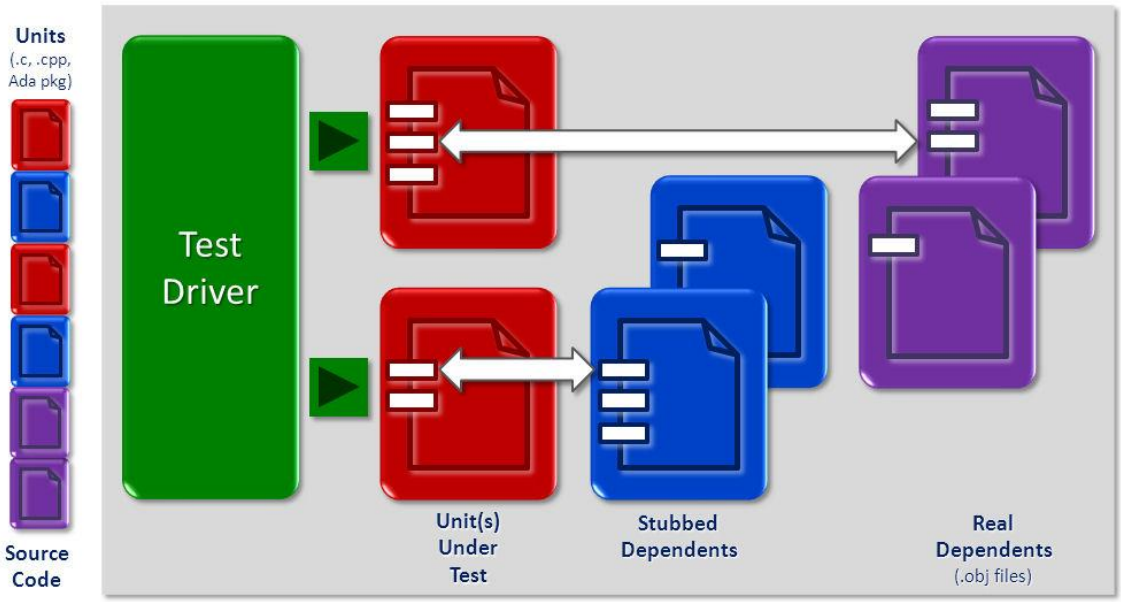


*Abb. 2: Automatisierte Generierung eines ausführbaren Testrahmens*

## *Erstellung von Testfällen für die Aktivierung des zu testenden Quellcodes*

VectorCAST/C++ (VectorCAST/Ada) kann Testfälle automatisch generieren. Alternativ hat der Anwender aber auch die Möglichkeit, für einen bestimmten Test Eingabedaten und erwartete Ergebnisse selbst anzugeben. Die Testfälle (Eingabedaten und erwartete Ergebnisse) werden für die anschließenden Regressionstests automatisch gespeichert.

## *Ausführung der Tests*

Mit VectorCAST/C++ (VectorCAST/Ada) können Tests auf mehrere Arten ausgeführt werden: auf einem Host, der nativ auf einem PC oder einem UNIX-Rechner gestartet wird; oder — jeweils in Verbindung mit VectorCAST/RSP — mit einem Simulator oder Emulator oder direkt in einem eingebetteten Ziel.
Tests können entweder von der grafischen Benutzeroberfläche (GUI) oder einer Befehlszeilenoberfläche (CLI) aus aktiviert werden.

## *„Pass/Fail"-Ergebnisse und Codeabdeckung*

VectorCAST/C++ (VectorCAST/Ada) generiert für die ausgeführten Tests Pass/Fail-Aussagen sowie Reports, die die zugehörige Codeabdeckung (coverage) angeben. VectorCAST/C++ (VectorCAST/Ada) unterstützt Statement-, Branch- und MC/DC-Coverage ebenso wie die DO-178B Standards A, B und C.

VectorCAST/C++ (VectorCAST/Ada) kann Abdeckungsdaten mit VectorCAST/Cover teilen (und umgekehrt), damit bei einer Kombination aus System- und Modul-/Integrationstests eine hundertprozentige Abdeckung garantiert ist.

## *Regressionstests leicht gemacht*

VectorCAST/C++ (VectorCAST/Ada) speichert alle im Laufe eines Entwicklungsprojekts für die automatisierten Regressionstests erforderlichen Informationen. Anders als bei vielen anderen Testframework-Tools muss kein Test-Code manuell geändert oder erneut erstellt werden.

## VectorCAST/RSP

Mittels VectorCAST/RSP können Sie VectorCAST/C++ (VectorCAST/Ada) zur Ausführung von Modul -und Integrationstests bei Echtzeitanwendungen nutzen, in der Zielumgebung selbst oder in einer Simulationsumgebung.

VectorCAST/RSP integriert sich in die Ziel-CLI, um den Cross-Compiler aufzurufen und eine E/A-Verbindung zwischen Zielumgebung und VectorCAST/C++ (VectorCAST/Ada) einzurichten. Die Testausführung ist für den Anwender transparent.

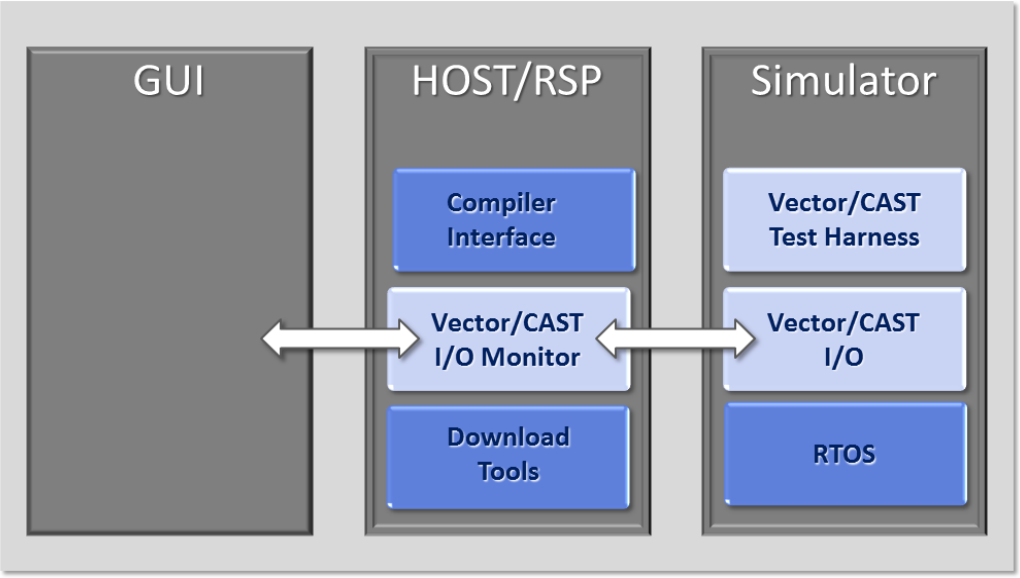


*Abb. 3: VectorCAST/RSP Architektur*

Bei einer Konfiguration wie in Abbildung 3 würden Sie VectorCAST/C++ (VectorCAST/Ada) auf einer Host-Plattform (mit aktiviertem RSP) zum Aufbau eines Testrahmens für den Test einer oder mehrerer Anwendungscodeeinheiten verwenden. Der RSP würde die korrekte Kompilierung sicherstellen und automatisch den ausführbaren Rahmen in die Zielumgebung herunterladen.

Dann würden Sie mittels VectorCAST/C++ (VectorCAST/Ada) auf der Host-Plattform Testfälle an den Anwendungsmodulen ausführen, die sich in der Ziel- oder einer Simulationsumgebung befinden. Den Bericht über die Abdeckung Ihrer Tests würden Sie ebenfalls mithilfe von VectorCAST/C++ (VectorCAST/Ada) erstellen.

## VectorCAST/Cover

Sie können die Wirksamkeit von Systemtests beurteilen, indem Sie mit VectorCAST/Cover feststellen, welche Bereiche einer Applikation angewendet (oder abgedeckt) wurden. Die Abdeckungsanalyse können Sie für die gesamte Applikation oder für einen Teil davon durchführen.

VectorCAST/Cover generiert andere Abdeckungsdaten als VectorCAST/C++ (VectorCAST/Ada). Die Abdeckungsdaten von VectorCAST/Cover werden aus System- oder Funktionstests gewonnen, die in der Regel anwendungsspezifisch und für die Tests von High-Level-Anforderungen ausgelegt sind.

Die Abdeckungsdaten von VectorCAST/C++ (VectorCAST/Ada) werden aus Modul- oder Integrationstests gewonnen. Meistens dienen solche Tests dazu, festzustellen, ob Algorithmen sich erwartungsgemäß verhalten, ob abweichende Bedingungen vorliegen und ob alle Pfade getestet werden.

Bei einem Systemtest kann normalerweise keine ganze Applikation getestet werden. Für eine hundertprozentige Abdeckung ist eine Kombination aus Modul-, Integrations- und Systemtest erforderlich.

### *100 %-ige Abdeckung*

Die Abbildungen 4 und 5 zeigen zwei Möglichkeiten, wie sich mit VectorCAST/Cover und VectorCAST/C++ (VectorCAST/Ada) eine hundertprozentige Codeabdeckung erzielen lässt.

Methode 1 (Abb. 4) entspricht dem üblichen Testverfahren, bei dem zuerst einzelne Komponenten, dann Teilsysteme und schließlich das ganze System getestet werden.

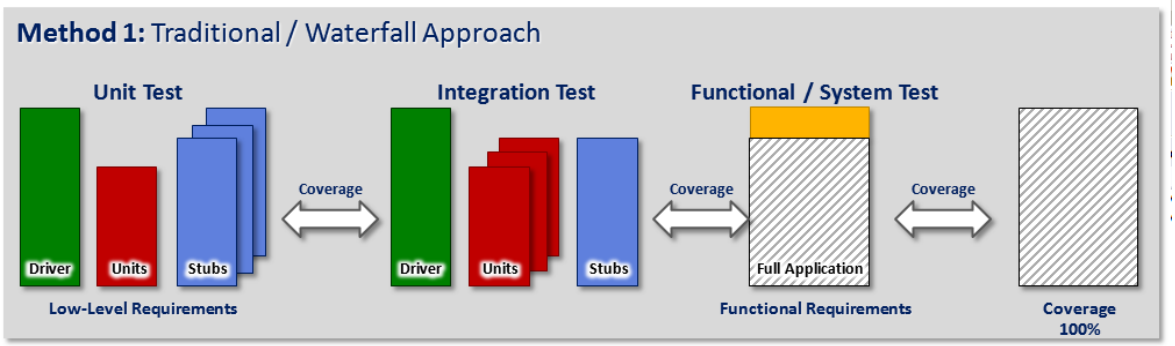


*Abb. 4: Methode 1 – Das oft angewendete Wasserfall-Prinzip*

Methode 2 (Abb. 5) kommt meistens bei anforderungsbasierten Tests zur Anwendung, bei denen üblicherweise zunächst High-Level-Systemanforderungen getestet werden und anschließend die systemnahen (Low-Level) Modul- oder Integrationstests erfolgen. Eine hundertprozentige Abdeckung ist das Ziel beider Methoden.

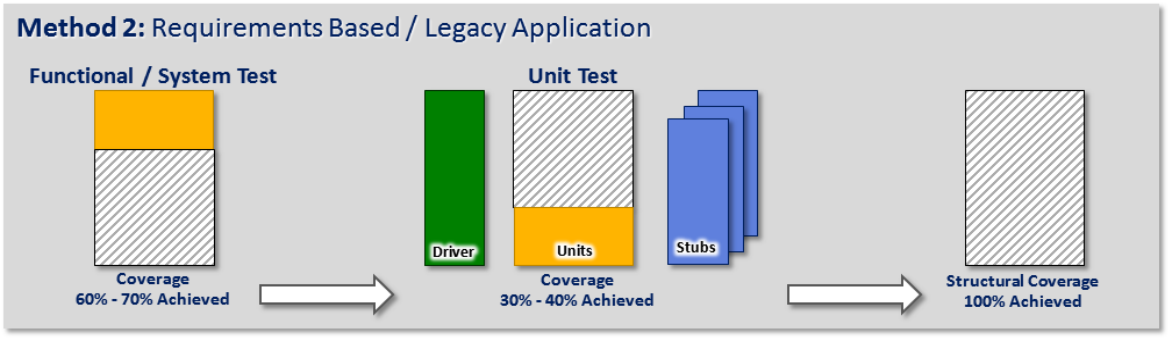


*Abb. 5: Methode 2 – Anforderungsbasierte / Legacy-Anwendung*

Bei der ersten Methode würden Sie mit VectorCAST/C++ (VectorCAST/Ada) Modul- und Integrationstests an Komponenten und Teilsystemen durchführen. In Zusammenhang mit diesen Tests würden Sie mithilfe der integrierten Abdeckungs-Utility die Quellcodeabdeckung ermitteln und berichten. Bei funktionalen Tests/Systemtests würden Sie mit dem VectorCAST/Cover Werkzeug und den Abdeckungsdaten von VectorCAST/C++ (VectorCAST/Ada) eine hundertprozentige Abdeckung sicherstellen.

Bei der zweiten Methode, angewandt in Zusammenhang mit Systemtests, würden Sie mit VectorCAST/Cover eventuell vorhandene ‚Löcher' in der Quellcodeabdeckung feststellen. Dann würden Sie mit Hilfe von VectorCAST/C++ (VectorCAST/Ada) die Abdeckung erhöhen, indem Sie an den Quellcodedateien mit den ‚Löchern' Testfälle ablaufen lassen. Zusätzlich würden Sie die VectorCAST/C++ (VectorCAST/Ada) Abdeckungsanalyse heranziehen und damit zusammen mit den Abdeckungsdaten von VectorCAST/Cover eine hundertprozentige Abdeckung sicherstellen.

***Hinweis:*** *VectorCAST/Cover unterstützt die McCabe-Metrik (zyklomatische Komplexität)*

**Über Vector Software**
Vector Software, Inc., ist der führende unabhängige Anbieter von automatisierten Testwerkzeugen für Entwickler von sicherheitskritischen eingebetteten Applikationen. Unsere Produktfamilie „VectorCAST" automatisiert und standardisiert die komplexen Aufgaben in Zusammenhang mit Modul-, Integrations- und Systemtests. VectorCAST Produkte unterstützen die Programmiersprachen C, C++ und Ada.

**Vector Software, Inc.**
1351 South County Trail, Suite 310
East Greenwich, RI 02818
USA
Telefon: 001.401.398.7185
Fax: 001.401.398.7186
E-Mail: info@vectorcast.com
www.vectorcast.com

**Vector Software**
Vorster Straße 80
47906 Kempen
Germany
Telefon: +49 2152 8088808
Fax: +49 2152 8088888
E-Mail: info@vectorcast.com
www.vectorcast.com