

Vector Software

WHITEPAPER

Mit VectorCAST die FDA-Anforderungen an Softwaretests erfüllen

Einführung

Die VectorCAST Produktfamilie automatisiert Testaktivitäten über den gesamten Software Entwicklungszyklus. In diesem White Paper geben wir einen umfassenden Überblick darüber, wie diese automatisierten Tests den Softwaretest-Anforderungen gemäß den „General Principles of Software Validation; Final Guidance for Industry and FDA Staff“ (Allgemeine Prinzipien der Software-Validierung, Leitfaden für Industrie und Mitarbeiter der FDA), Abschnitt 5.2.5 „Testing by the Software Developer“ (Vom Softwareentwickler durchzuführende Tests) genügen.

Diese FDA-Richtlinie bezieht sich insbesondere auf:

- > Modultests
- > Integrationstests
- > Systemtests

In Absatz 5.2.5 finden sich die folgenden Anforderungen:

Bei den Tests sollen Softwareprodukte unter bekannten Bedingungen mit definierten Eingaben und dokumentierten Ausgaben ablaufen.

Das erwartete Ergebnis ist ein wesentliches Testelement.

Die Testreihe beginnt mit einem Modultest und endet mit einem Systemtest.

Die Höhe der strukturellen Abdeckung und die möglichen Risiken stehen in einem angemessenen Verhältnis.

Mittels Regressionsanalyse und -tests wird nachgewiesen, dass eine Änderung kein Problem zur Folge hatte.

Softwareprodukttests können in Modul-, Integrations- und Systemtests unterteilt werden.

Mit VectorCAST die FDA-Anforderungen an Softwaretests erfüllen

VectorCAST unterstützt sämtliche oben aufgeführten Testaktivitäten, einschließlich der Generierung der für die FDA-Prüfung benötigten Testobjekte.

Die VectorCAST Produktfamilie umfasst vier sich ergänzende Technologien:

VectorCAST/C++	<i>Automatisiert den Testprozess von Quellcode-Modulen, die in C oder C++ entwickelt worden sind. Unterstützt Modul- und Integrationstests. Austausch von Daten mit VectorCAST/Cover für 100 %-ige Testabdeckung,</i>
VectorCAST/Cover	<i>Generiert Codeabdeckungsobjekte bei funktionalen Tests oder Systemtests.</i>
VectorCAST/RSP (Runtime Support Package)	<i>Erweiterung von VectorCAST/C++. Ermöglicht die Ausführung von Testfällen für Echtzeitanwendungen in einer eingebetteten Zielumgebung oder in einer.</i>
VectorCAST/Manage	<i>Dient zur Automatisierung der Regressionstest-Aktivitäten für alle Modul- und Integrationstests.</i>

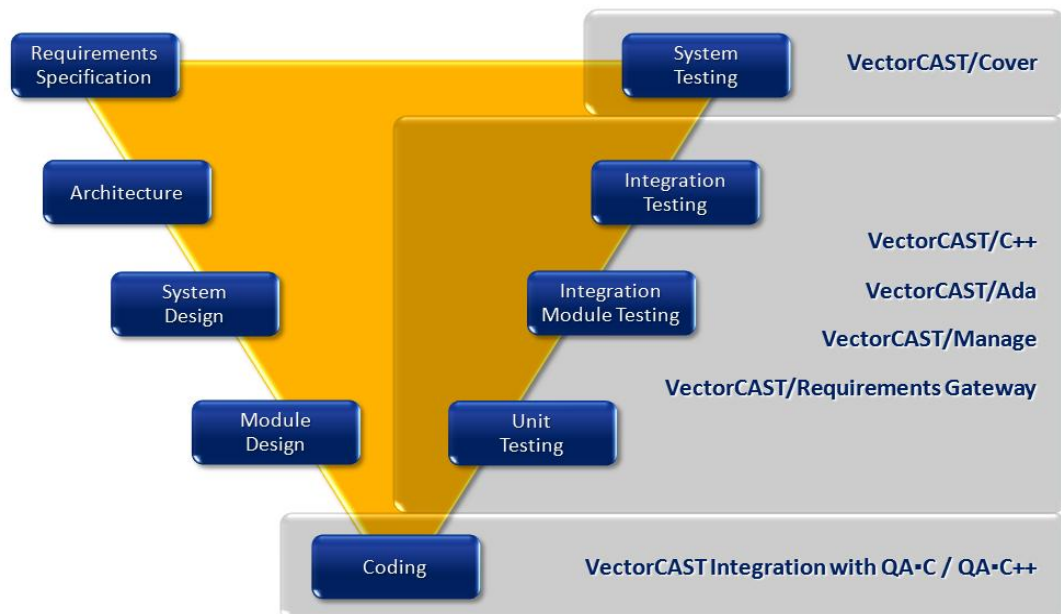


Abb. 1 – Einsatzmöglichkeiten von VectorCAST im Software-Entwicklungsprozess

VectorCAST/C++

VectorCAST/C++ automatisiert die wichtigsten Softwaretestaktivitäten. Nachfolgend einige Beispiele:

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

Wie in Abbildung 2 dargestellt, umfasst der von VectorCAST/C++ 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.

Datengesteuerter Testrahmen

Die Testdaten werden während der Ausführung vom Rahmen gelesen. Damit entfällt die Notwendigkeit, bei der Durchführung des gleichen Tests mit mehreren Testfällen jedes Mal einen neuen ausführbaren Rahmen zu kompilieren und zu verknüpfen.

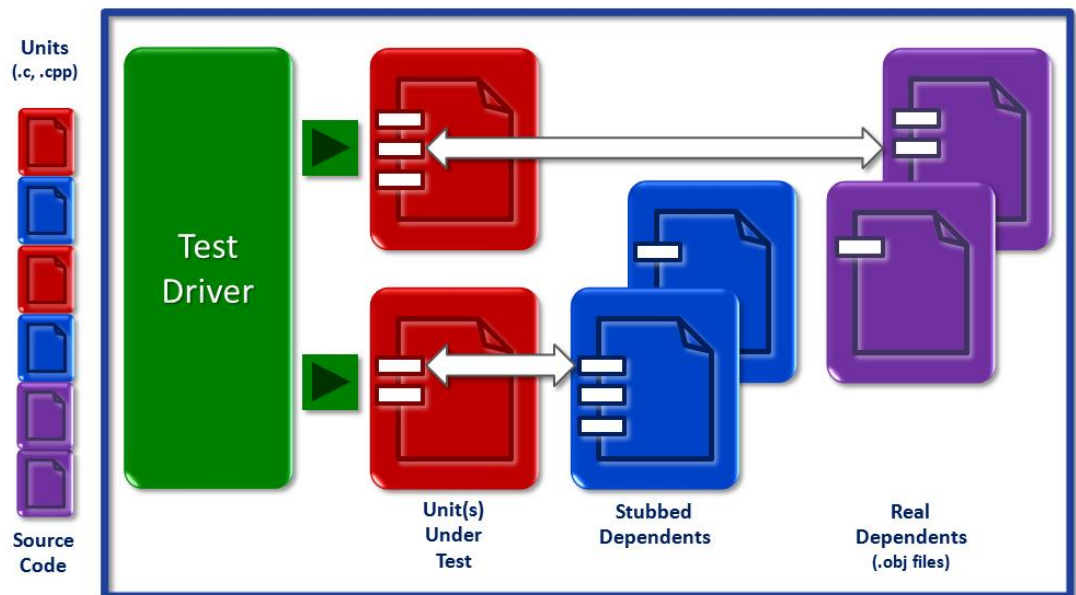


Abb. 2 - Automatisierte Generierung eines ausführbaren Testrahmens

Erstellung von Testfällen für die Aktivierung des zu testenden Quellcodes

VectorCAST/C++ generiert automatisch vollständige Testfälle. Alternativ hat der Anwender 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++ sind drei Arten von Testausführungen möglich:

1. Auf einem Host, der nativ auf einem PC oder einem UNIX-Rechner gestartet wird; und
2. mit einem Simulator oder Emulator (in Verbindung mit VectorCAST/RSP); oder
3. direkt in einem eingebetteten Ziel (in Verbindung mit VectorCAST/RSP)

Tests können entweder von einer 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++ unterstützt Statement-, Branch- und MC/DC-Coverage.

VectorCAST/C++ kann Abdeckungsdaten mit VectorCAST/Cover teilen (und umgekehrt), so dass eine hundertprozentige Abdeckung garantiert ist.

Regressionstests leicht gemacht

VectorCAST/C++ speichert alle im Laufe eines Entwicklungsprojekts für die automatisierten Regressionstests erforderlichen Informationen. Im Zusammenspiel mit VectorCAST/Manage automatisiert VectorCAST/C++ das gesamte Regressionstestverfahren.

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.

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. Dementsprechend generiert VectorCAST/Cover andere Abdeckungsdaten als VectorCAST/C++.

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

Da bei einem Systemtest normalerweise eine Applikation nicht in ihrer Gesamtheit getestet werden kann, ist für eine hundertprozentige Abdeckung eine Kombination aus Modul-, Integrations- und Systemtest erforderlich.

100 % Abdeckung

Abbildung 3 zeigt zwei Möglichkeiten, wie sich mit VectorCAST/Cover und VectorCAST/C++ eine hundertprozentige Codeabdeckung erzielen lässt.

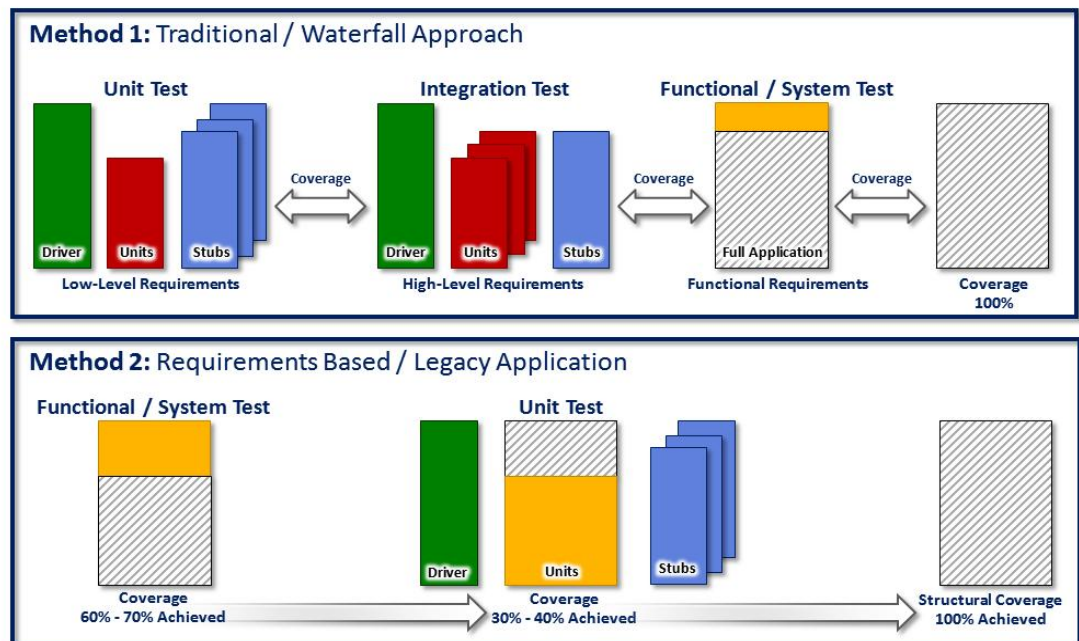


Abb. 3 – Zwei Methoden zur Sicherstellung einer 100 %-igen Abdeckung

Mit VectorCAST die FDA-Anforderungen an Softwaretests erfüllen

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

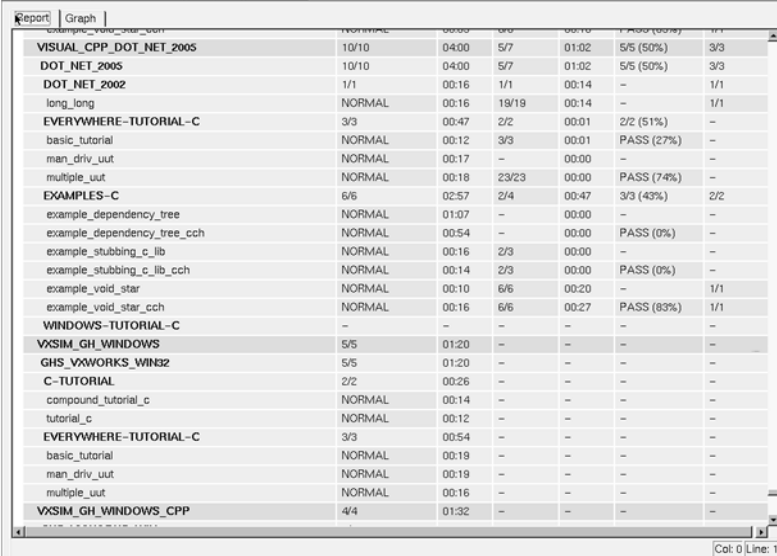
Methode 2 kommt bei anforderungsbasierten Tests zur Anwendung, bei denen zunächst High-Level-Systemanforderungen getestet werden und anschließend die systemnahen (Low-Level) Modul- oder Integrationstests erfolgen.

Bei der ersten Methode führt VectorCAST/C++ Modul- und Integrationstests an Komponenten und Teilsystemen durch. In Zusammenhang mit diesen Tests würden Sie mithilfe der integrierten Abdeckungs-Utility die Quellcodeabdeckung ermitteln und berichten. Bei funktionalen Tests/Systemtests tauscht das VectorCAST/Cover Werkzeug Abdeckungsdaten mit VectorCAST/C++ aus, womit eine hundertprozentige Abdeckung sichergestellt ist.

Bei der zweiten Methode, angewandt in Zusammenhang mit Systemtests, stellen Sie mit VectorCAST/Cover eventuell vorhandene ‚Löcher‘ in der Quellcodeabdeckung fest. Dann erhöhen Sie mithilfe von VectorCAST/C++ die Abdeckung, indem Sie an den Quellcodedateien mit den ‚Löchern‘ Testfälle ablaufen lassen. Auch bei dieser Methode tauscht die Abdeckungsanalyse-Utility von VectorCAST/C++ Daten mit VectorCAST/Cover aus, um eine hundertprozentige Abdeckung sicherzustellen.

VectorCAST/Manage

VectorCAST/Manage ist eine Erweiterung der VectorCAST Produktfamilie von Modul- und Integrationstestwerkzeugen. VectorCAST/Manage ermöglicht Ihnen den Import von bestehenden VectorCAST/C++ Testumgebungen in Regressions-Testreihen und stellt einen zentralen Kontrollpunkt für alle Modul- und Integrations-Regressionstest-Aktivitäten dar. Übersichtliche Protokolle, Zusammenfassungen (Summenberichte) und farbig gekennzeichnete Pass/Fail-Kriterien zeigen den Status jedes Tests innerhalb der Regressionstestreihe.



Test Case Name	Status	Pass	Fail	Total	Pass %	Fail %	
VISUAL_CPP_DOT_NET_2005	NORMAL	10/10	04:00	5/7	01:02	5/5 (50%)	3/3
DOT_NET_2005	NORMAL	10/10	04:00	5/7	01:02	5/5 (50%)	3/3
DOT_NET_2002	NORMAL	1/1	00:16	1/1	00:14	-	1/1
long_long	NORMAL	00:16	19/19	00:14	-	-	1/1
EVERYWHERE-TUTORIAL-C	NORMAL	3/3	00:47	2/2	00:01	2/2 (51%)	-
basic_tutorial	NORMAL	00:12	3/3	00:01	PASS (27%)	-	-
man_driv_uut	NORMAL	00:17	-	00:00	-	-	-
multiple_uut	NORMAL	00:18	23/23	00:00	PASS (74%)	-	-
EXAMPLES-C	NORMAL	6/6	02:57	2/4	00:47	3/3 (43%)	2/2
example_dependency_tree	NORMAL	01:07	-	00:00	-	-	-
example_dependency_tree_cch	NORMAL	00:54	-	00:00	PASS (0%)	-	-
example_stubbing_c_lib	NORMAL	00:16	2/3	00:00	-	-	-
example_stubbing_c_lib_cch	NORMAL	00:14	2/3	00:00	PASS (0%)	-	-
example_void_star	NORMAL	00:10	6/6	00:20	-	-	1/1
example_void_star_cch	NORMAL	00:16	6/6	00:27	PASS (83%)	-	1/1
WINDOWS-TUTORIAL-C	-	-	-	-	-	-	-
VXSIM_GH_WINDOWS	NORMAL	5/5	01:20	-	-	-	-
GHS_VXWORKS_WIN32	NORMAL	5/5	01:20	-	-	-	-
C-TUTORIAL	NORMAL	2/2	00:26	-	-	-	-
compound_tutorial_c	NORMAL	00:14	-	-	-	-	-
tutorial_c	NORMAL	00:12	-	-	-	-	-
EVERYWHERE-TUTORIAL-C	NORMAL	3/3	00:54	-	-	-	-
basic_tutorial	NORMAL	00:19	-	-	-	-	-
man_driv_uut	NORMAL	00:19	-	-	-	-	-
multiple_uut	NORMAL	00:16	-	-	-	-	-
VXSIM_GH_WINDOWS_CPP	NORMAL	4/4	01:32	-	-	-	-

Beispiel für einen VectorCAST/Manage Report

VectorCAST für Tests in eingebetteten Umgebungen

VectorCAST/C++ in Zusammenarbeit mit dem VectorCAST/Runtime Support Package (RSP) ermöglicht nahtlose Modul- und Integrationstests bei Echtzeitanwendungen, 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++ einzurichten. Die Testausführung ist für den Anwender transparent.

Bei einer Konfiguration wie in Abbildung 4 erstellt VectorCAST/C++ (auf einer Host-Plattform mit aktiviertem RSP) einen Testrahmen für eine oder mehrere Anwendungscodeeinheiten. Der RSP stellt die korrekte Kompilierung sicher und lädt automatisch den ausführbaren Rahmen in die Zielumgebung herunter. VectorCAST/C++ führt Testfälle (auf dem Hostplattform-Rahmen) an den Anwendungsmodulen aus, die sich in der Zielumgebung befinden.

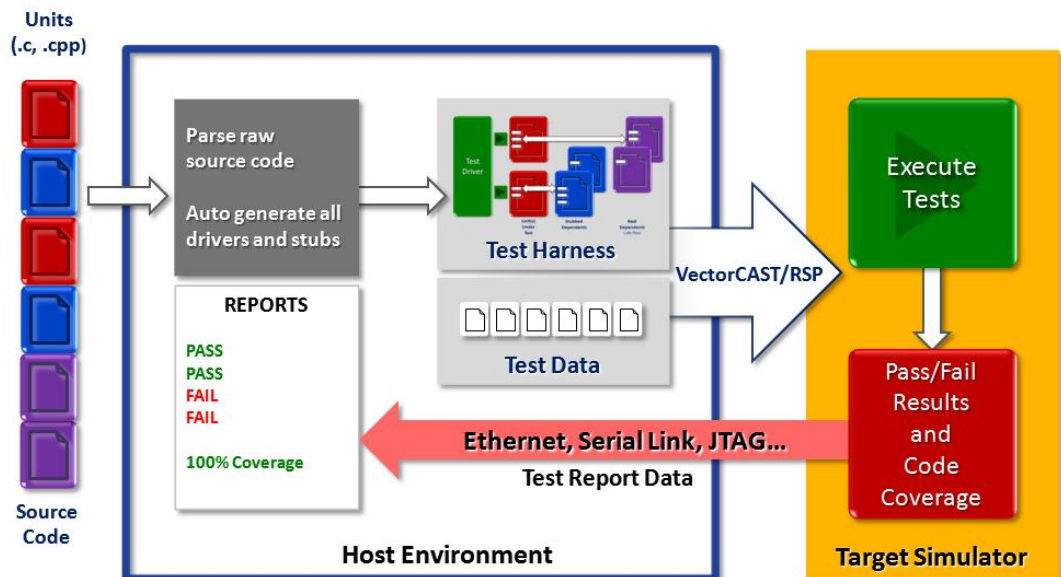


Abb. 4 – VectorCAST Architektur für eingebettete Tests

Validierung kommerzieller Testwerkzeuge

In den Dokument „*General Principles of Software Validation; Final Guidance for Industry and FDA Staff*“ steht in Abschnitt 5.2.5 Folgendes (deutsche Übersetzung):

„Software-Testwerkzeuge werden häufig eingesetzt, wenn bei den Tests solcher Software-Produkte Übereinstimmung, Gründlichkeit und Effizienz sichergestellt werden soll, und um die Anforderungen der geplanten Testaktivitäten zu erfüllen. Solche Werkzeuge können selbst entwickelte, unterstützende Software umfassen, die die Modultests und anschließenden Integrationstests (z. B. Treiber und Stubs) ermöglichen, ebenso wie kommerzielle Software-Test-Tools. Das Qualitätsniveau der Werkzeuge darf nicht unter dem des Softwareprodukts liegen, bei dessen Entwicklung sie zur Anwendung kommen. Die Validierung der Softwarewerkzeuge für ihren vorgesehenen Bestimmungszweck ist mittels entsprechender Unterlagen nachzuweisen.“

Bei VectorCAST erfolgt die Validierung über die Zulassungsdokumente *Tool Operational Requirements* (TOR) und *Tool Qualification Data* (TQD).

Das Dokument ***Tool Operational Requirements*** umfasst folgende Informationen:

- > *VectorCAST-Funktionalität in überprüfbaren Anforderungen*
- > *Projekteinsatzumgebung*
- > *Konfigurationsverwaltungsprozess*
- > *Methode zur Erlangung des Nachweises, dass VectorCAST bezüglich vorgegebener Anforderungen erfolgreich getestet wurde*

Das Dokument ***Tool Qualification*** ist vorgesehen für:

- > *Toolqualifizierungs-Testdaten und Testergebnisse für jede vorgegebene Anforderung*

Zum Zulassungsprozess gehören normalerweise Interaktionen mit dem bewertenden Anwender und gegebenenfalls mit der entsprechenden Zertifizierungsstelle. Muster der genannten Dokumente sind auf Anfrage erhältlich.

Fazit

Gemäß den „*General Principles of Software Validation; Final Guidance for Industry and FDA Staff*“ werden die Entwicklungstests in die in Abbildung 5 angeführten Ebenen unterteilt, damit eine gründliche und strenge Prüfung eines Softwareprodukts gewährleistet ist. Mithilfe der VectorCAST-Produktfamilie ist es inzwischen vielen Herstellern medizintechnischer Geräte gelungen, diese Testanforderungen an den vorgeschriebenen Testgegenständen zu erfüllen.

FDA Entwicklungstestanforderung	Erfüllbar mit VectorCAST-Produkt
Tests auf Modul- (oder Komponenten-) Ebene: Eine frühzeitige Prüfung der Unterprogrammfunktionen, mit der sichergestellt wird, dass die Tests sich auch auf Funktionen erstrecken, die auf Systemebene nicht sichtbar sind. Modultests gewährleisten, dass das Endprodukt qualitativ hochwertige Softwaremodule umfasst.	<ul style="list-style-type: none">•VectorCAST/C++ (Modultest)•VectorCAST/RSP (Zieltest)•VectorCAST/Manage (Regressionstest)•VectorCAST/Requirements Gateway (Anforderungsverfolgung)
Test auf Integrationsebene: Prüfung der Übertragung von Daten und Steuerinformationen über die internen und externen Schnittstellen eines Programms. Externe Schnittstellen sind die Schnittstellen zu anderer Software (einschließlich Betriebssystem-Software), Systemhardware und den Anwendern und fungieren sozusagen als Datenübermittlungsabschnitte.	<ul style="list-style-type: none">•VectorCAST/C++ (Modultest)•VectorCAST/RSP (Zieltest)•VectorCAST/Manage (Regressionstest)•VectorCAST/ Requirements Gateway (Anforderungsverfolgung)
Tests auf Systemebene: Hiermit wird nachgewiesen, dass angegebene Funktionen wirklich vorhanden sind und dass das Softwareprodukt zuverlässig ist. Es erfolgt eine Überprüfung der Programmfunktionalität und -leistung hinsichtlich der Anforderungen an das Softwareprodukt in der/den angegebenen Arbeitsumgebung(en) .	<ul style="list-style-type: none">•VectorCAST/Cover•VectorCAST/Requirements Gateway

Abb. 5 – FDA Entwicklungstestebenen und VectorCAST

Weitere Ressourcen

Dieses White Paper soll als Einführung in die Produktliteratur dienen, die auf www.vectorcast.com unter dem Menüpunkt **Ressourcen** verfügbar ist.

VectorCAST/C++	<i>Modul und Integrationstests für C/C++</i>
-----------------------	--

VectorCAST/Cover	<i>Mehrsprachiges Codeabdeckungs-Tool</i>
-------------------------	---

VectorCAST/Manage	<i>Regressionstest-Manager</i>
--------------------------	--------------------------------

VectorCAST/RSP	<i>Echtzeittests eingebetteter Systeme</i>
-----------------------	--

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