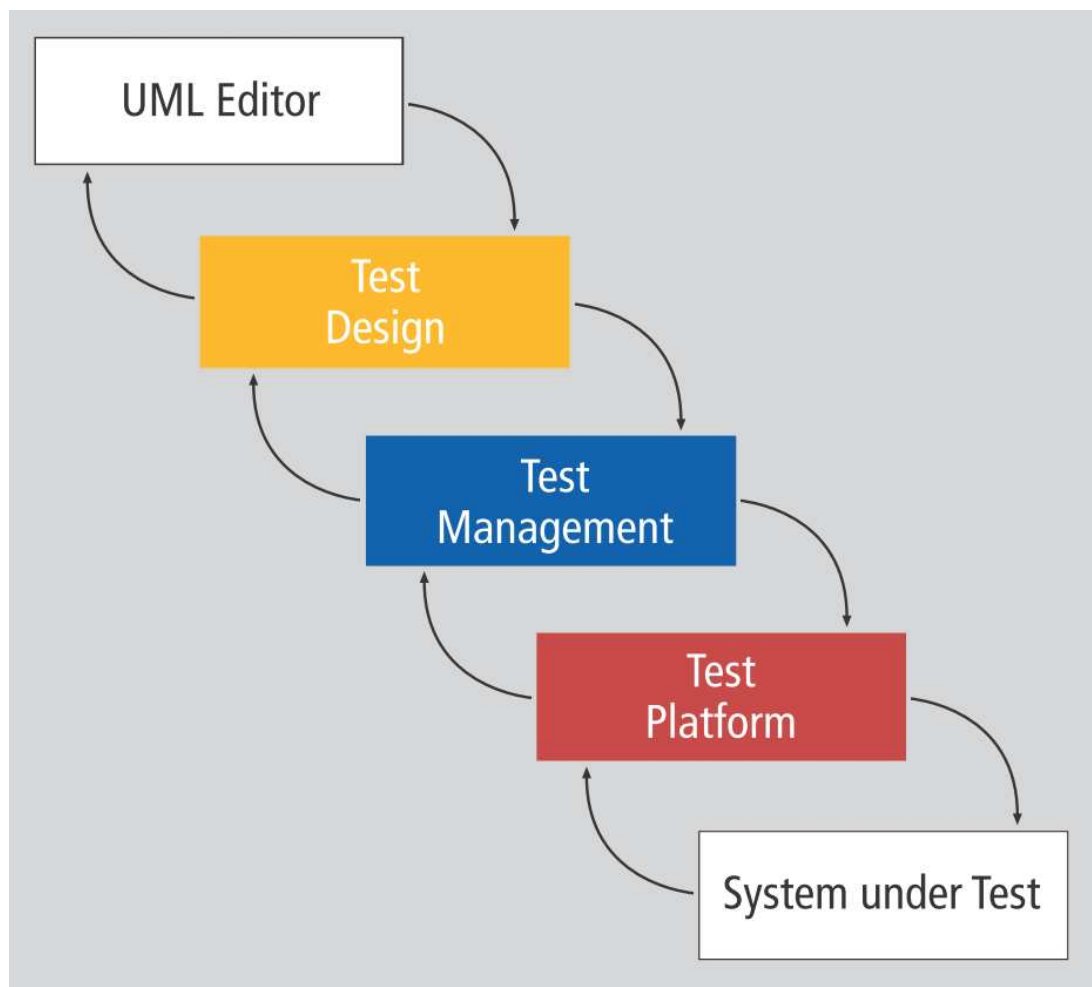


# CETES

## PRODUKT DATENBLATT Komponente Testdesign

Version 1.0



## 1. Technische Anforderungen

Systemvoraussetzung Testdesign

Betriebssysteme: Windows oder Linux Plattform mit Java Virtual Maschine  
Prozessor: x86-Pentium oder höher, mindestens 2000 MHz  
Arbeitsspeicher: Mind. 2 GB  
Festplattenplatz: 100 MB

## 2. Lizenz

Für die Komponente Testdesign benötigen Sie eine Lizenz deren Bereitstellung über die sepp.med GmbH erfolgt.

Die Lizenz kann vom Typ Floating oder Node-locked sein.

### 3. Leistungsbeschreibung

#### Automatisches Generieren

Aus einer kompakten UML-Beschreibung in Form von Aktivitäts- oder Zustandsdiagrammen erstellt **.getmore** vollständige, durchführbare Testfälle. Alle testrelevanten Informationen – inklusive der Test- und Testmanagementdaten – sind dabei in diesen Diagrammen hinterlegt.

Die automatische Generierung liefert einen Satz von Testfällen, der entweder vom Anwender manuell abgearbeitet wird oder automatisiert ablaufen kann. Zugleich kann die Testfallbeschreibung in beliebige Dokumentformate generiert werden.

#### Abarbeitungsstrategien

Zur Generierung von Testfällen stellt **.getmore** folgende Strategien bereit:

- **Path Coverage** – Erzeugt in einem Diagramm alle Pfade, die von einem Startpunkt bis zum Endpunkt möglich sind.
- **Edge / Node Coverage** – (c0 / c1) Erzeugt Testfälle, in denen jeder Zweig / Anweisung mindestens einmal vorkommt.
- **Random** – Erzeugt aus einem Diagramm zufällig Testfälle, bis die definierte Anzahl erreicht ist.
- **Named Path** – Legt einen Pfad fest, auf dem das Diagramm durchlaufen werden muss.
- **Guided Path** – Mit Hilfe einer Tag-Systematik werden zu generierende Testfälle definiert

#### Managementinformationen

Die Abarbeitung lässt sich mit **.getmore** durch verschiedene Steuerungsmechanismen gezielt beeinflussen und optimieren:

- **Testfalllänge** – Erzeugt nur Testfälle bis zu einer definierten Anzahl von Testschritten.
- **Beschränkung der Schleifendurchläufe** – Durchläuft Schleifen im Diagramm nur bis zur definierten Höchstzahl.
- **Guard Conditions** – An beliebigen Stellen im Diagramm können Variablen gesetzt und während der Testfallgenerierung durch Guard Conditions ausgewertet werden. Letztere steuern den weiteren Fortgang der Pfadabarbeitung. Die Variablen, auf denen die Überprüfung beruht, werden lokal (nur im aktuellen Diagramm) oder global (über Diagrammgrenzen hinweg) definiert. Sie können sich dynamisch während der Abarbeitung ändern.
- **Priorisierung** – **.getmore** ermöglicht die Priorisierung von Übergängen, um Testfälle mit niedriger Priorität bei Bedarf von der Testfallerzeugung auszuschließen.
- **Tags** – mit Hilfe von im Modell angegebenen Tags ist es möglich, nach verschiedensten Testfällen zu suchen
- **Requirements** – mit Hilfe von im Modell angegebenen Requirements ist es möglich, gezielt zu testen.
- **Best-N Test Cases** - Bietet die Möglichkeit die generierten Testfälle nach verschiedensten Metriken und Merkmalen zu priorisieren

## Hierarchische Strukturen

**Komplexe Systeme** lassen sich durch Verwendung von Hierarchien strukturiert und übersichtlich beschreiben. Modelle sollten daher hierarchisch aufgebaut sein. Damit ist gewährleistet, funktional zusammengehörende Einheiten in eigenen Diagrammen zu bündeln und mit den optimalen Strategien zu bearbeiten.

## Metriken

**.getmore** stellt umfangreiche Statistiken und Kennzahlen zur Verfügung, mit deren Hilfe man die Ergebnisse einer Generierungen bewerten kann. Unter anderem hat man Zugriff auf: **Anzahl Test Cases, Test Case Länge, Generierungsdauer, Abdeckungsmaße, etc.**

## Kostenfunktion

**.getmore** bietet die Möglichkeit eine Auswertung über die im Testmodell angegebenen Kosten zu erstellen. Man kann zum Beispiel die „teuersten“ als auch die „kostengünstigsten“ Testfälle ermitteln, die die Abdeckungsanforderungen erfüllen oder eine eigene Kostenfunktion definieren.

## Timing-Aspekte

**.getmore** bietet die Möglichkeit zeitliches Verhalten im Modell zu berücksichtigen. Es können Durchlaufzeiten und Timer angegeben werden und in der Testfallerstellung berücksichtigt werden. Die Zeitangaben finden außerdem in den angebotenen Metriken Wiederverwendung.

## Persistenz

Bei **.getmore** ist es möglich auf bereits importierte Modelle wieder zuzugreifen und eine neue Strategie darauf anzuwenden. Dadurch wird das Arbeiten mit **.getmore** effizienter und einfacher. Ebenso können die erzeugten Test Cases separat abgespeichert und geschützt werden.

## Anforderungsnachverfolgung

Die Komponente **Testdesign** ist in der Lage Requirements z.B. aus einer DOORS Datenbank zu importieren und in **.getmore** zu verfolgen. Dadurch ist gewährleistet, dass Anforderungen vom Modell bis in die einzelnen Testschritte und in die **Testmanagement** Komponente von **CETES** durchgereicht werden.

### Fehlerrückverfolgung

Wenn bei der Testdurchführung auf der **Testplattform** Komponente Fehler in einem Testschritt auftreten, so kann dieser zusammen mit dem dazugehörigen Pfad direkt im Modell bzw. in **.getmore** visualisiert werden. Dies erleichtert die Planung von Regressionstests.

### Wiederverwendung

Durch die Option, Diagramme, Subdiagramme und Sequenzen als Bibliotheken zusammenzufassen und wiederzuverwenden, lassen sich effizient und wartungsfreundlich neue Modelle erzeugen.

### CETES Plattformintegration

Die Komponente **Testdesign** von **CETES** vereint zwei Technologien. Die Methodik des modellzentrierten Tests (.mzT) und das Testfallgenerierungswerkzeug **.getmore**. **.getmore** basiert auf dem Eclipse Framework und ist als eigenständige RCP-Anwendung, wie auch als direktes PlugIn für die **CETES Testmanagement** Komponente verfügbar.

### Die Vorteile von CETES mit .mzT und .getmore:

- intuitive Handhabung
- viel Freiheit bei der Erstellung von Testmodellen
- flexible Anpassung an die unterschiedlichsten Einsatzgebiete
- .getmore verfügt über Anbindungen für die gängigsten UML- und Testmanagementwerkzeuge.
- weitere Schnittstellen lassen sich leicht realisieren.
- nahtlose Integration innerhalb von **CETES** für einen optimalen Workflow



## Weitere Werkzeugintegrationen

.getmore bietet auch standardmäßig Schnittstellen zu folgenden Standard-Werkzeugen<sup>1</sup> an:

- IBM Telelogic DOORS
- Sparx Systems Enterprise Architect
- ARTISAN Studio
- MID Innovator
- .modeller (sepp.med Modellierungswerkzeug auf Basis von ArgoUML)
- **CETES** - Testmanagementkomponente
- HP TestDirector/ Quality Center
- IBM Rational TestManager
- beliebige Script-/Programmiersprachen wie: Borland SilkTest, C/C++/C#, Java, Perl, Python, etc.
- beliebige Textformate, z.B. für Microsoft Word/Excel oder Telelogic DOORS

Es besteht die Möglichkeit, automatische Tests für verschiedene Standard- oder proprietäre Testautomatisierungswerkzeuge, wie z. B. Borland Segue SilkTest zu generieren. Die offene Architektur von .getmore erlaubt die Anbindung weiterer Werkzeuge.



<sup>1</sup> Alle verwendeten Markennamen sind geschützte Begriffe der jeweiligen Hersteller.

## 4. Über sepp.med

Mit mehr als 28 Jahren Erfahrung im **IT-Sektor** ist sepp.med ein verlässlicher und bewährter Partner bei der **Entwicklung, Integration und Qualitätssicherung von komplexen und sicherheitskritischen Systemen.**

Die besondere Expertise in der Medizintechnik und weiteren Branchen wie der Pharmazie oder der Automobiltechnik haben das in Röttenbach bei Erlangen im Medical Valley ansässige Unternehmen weit über die Grenzen der Region hinaus bekannt gemacht.



Als **Experten für Softwarequalität** sind wir die richtigen Ansprechpartner, wenn ein Softwareprodukt oder ein Prozess allen qualitativen und quantitativen Anforderungen entsprechen muss. Wir validieren Softwareprodukte, Produktions- und Entwicklungsprozesse sowie unterstützende Werkzeuge. Dabei bewährt sich unser **professionelles Testmanagement**, das alle Projektschritte von der Testspezifikation über die Testdurchführung bis hin zur Abnahme durch den Kunden umfasst.

Wir bieten umfassende Prozessberatung und unterstützen unsere Kunden bei der Vorbereitung von IT-Prozess-Assessments wie **SPICE** oder **ITIL**.

Im Umfeld der medizinischen Informationstechnologie nehmen wir die **Integration von RIS, PACS und KIS** in bestehende Umgebungen vor. Unsere Kompetenz reicht dabei von der Zieldefinition bis hin zur Gestaltung, Lenkung und Entwicklung des Integrationsprozesses. Darüber hinaus wickeln wir alle Integrationsaufgaben ab.

Unser umfassendes Wissen über neue und komplexe Architekturen konnten wir bereits in einer Vielzahl unterschiedlicher **Softwareentwicklungsprojekte** erfolgreich unter Beweis stellen. Mit unserer Erfahrung erkennen wir in Kundenprojekten exakt die fachlichen und technischen Anforderungen. Je nach Bedarf agieren wir bei unseren Kunden vor Ort oder als Outsourcing-Partner.

Das Know-how aus der engen **Kooperation mit Hochschulen** und der **regelmäßigen Fort- und Weiterbildung** unserer Mitarbeiter fließt unmittelbar in unsere innovativen Lösungen für industrielle Projekte mit ein.

Als **akkreditierter Schulungsanbieter** für **Certified Tester Foundation Level** vermitteln wir die grundlegenden Aufgaben, Methoden und Techniken des Softwaretests. Zudem bieten wir individuelle **Schulungen im Umgang mit den verschiedensten Werkzeugen und Produkten** an.

Zusatzinformationen über CETES erhalten Sie unter [www.cetes.eu](http://www.cetes.eu).

Die CETES-Partner erreichen Sie wie folgt:

**Kölsch & Altmann  
Software & Management  
Consulting GmbH**  
Perlacher Straße 21  
81539 München  
[www.koelsch-altmann.de](http://www.koelsch-altmann.de)  
[cetes@koelsch-altmann.de](mailto:cetes@koelsch-altmann.de)

**RST Industrie Automation GmbH**  
Rosenheimer Landstraße 145  
85512 Ottobrunn  
[www.rst-automation.de](http://www.rst-automation.de)  
[cetes@rst-automation.de](mailto:cetes@rst-automation.de)

**sepp.med gmbh**  
Gewerbering 9  
91341 Röttenbach  
[www.seppmed.de](http://www.seppmed.de)  
[cetes@seppmed.de](mailto:cetes@seppmed.de)