Verifysoft TECHNOLOGY

Softwaretest- & Analyse-Tools für Produktivität & Qualität

- Code Coverage
- Software-Komplexitätsmessung
- Statische Codeanalyse
- Dynamische Codeanalyse
- Safety-critical embedded

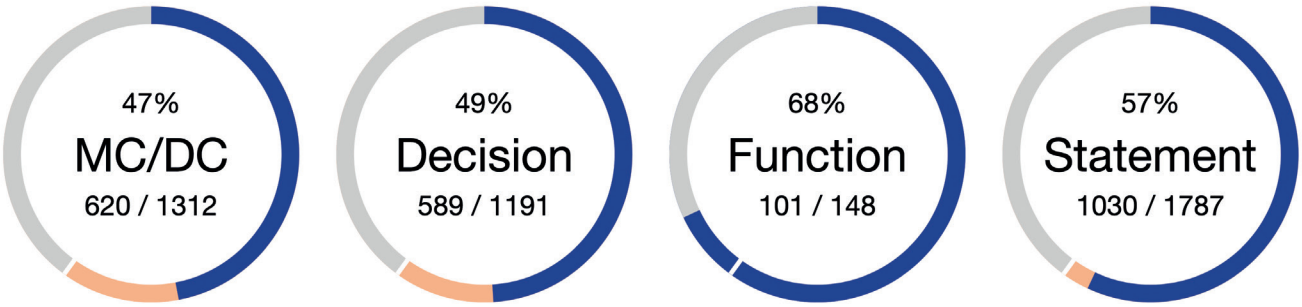
Testwell CTC++ Code Coverage Analyser

Code Coverage für die höchsten Anforderungen der Sicherheitsstandards

Testwell CTC++ analysiert, welche Teile Ihres Quellcodes getestet wurden. Testwell CTC++ unterstützt alle Coverage-Stufen und wird von führenden Unternehmen für sicherheitskritische Projekte eingesetzt.

Coveragemaße

Testwell CTC++ liefert alle Coveragestufen, die von Normen für die sicherkeitskritische Softwareentwicklung gefordert werden: Function Coverage, Statement Coverage, Decision oder synonym Branch Coverage und die Modified Condition/Decision Coverage (MC/DC). Zusätzlich können auch Condition und Multicondition Coverage ermittelt werden.

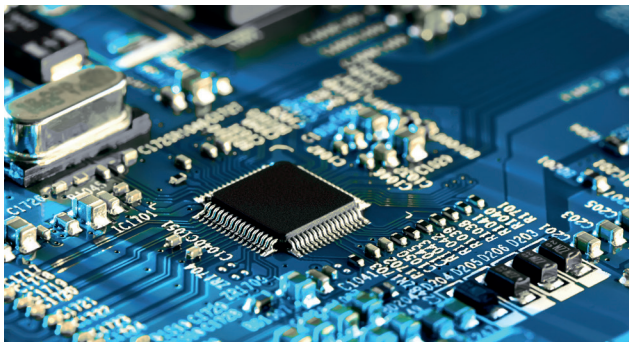


Desktop Applikationen

- ✓ Geringer Einfluss auf den Build Prozess
- ✓ Skalierbar für große Projekte
- ✓ Unabhängig vom Compiler
- ✓ Für Windows, Linux, macOS

Embedded Software

- ✓ Geringe Speicheranforderungen
- ✓ Testen auf jedem Target
- ✓ Mit jedem Cross-Compiler
- ✓ Anpassbarer Runtime-Layer



Einfache Nutzung

- ✓ Generische Build-Integration
- ✓ Sehr schnell in der Ausführung
- ✓ Nahtlose Integration in viele IDEs
- ✓ Modulare Architektur für volle Intergrierbarkeit

Programmiersprachen

- ✓ C, C++
- ✓ Add-On für Java

Arbeitsprinzip

Die Coverage-Messung erfolgt mit Testwell CTC++ in drei unabhängigen Phasen:

Instrumentieren

Während des Kompiliervorgangs instrumentiert Testwell CTC++ automatisch den Quellcode, d. h., er wird mit Messcode angereichert. So entsteht eine instrumentierte Version des Programms oder des Test-Executables – vollautomatisch während des Buildvorgangs oder auf Basis einer einfachen, einmaligen Buildkonfiguration.

Testen

Jede Art von Test kann wie gewohnt ausgeführt werden: Unit- Tests, Integrationstests oder komplette Systemtests. Die Coverage-Messdaten werden in eine Datei geschrieben. Bei Testdurchführung auf einem Target ist dieses Herausschreiben vollständig anpassbar, die Daten können z. B. direkt zum Hostrechner übertragen werden.

Berichte erzeugen

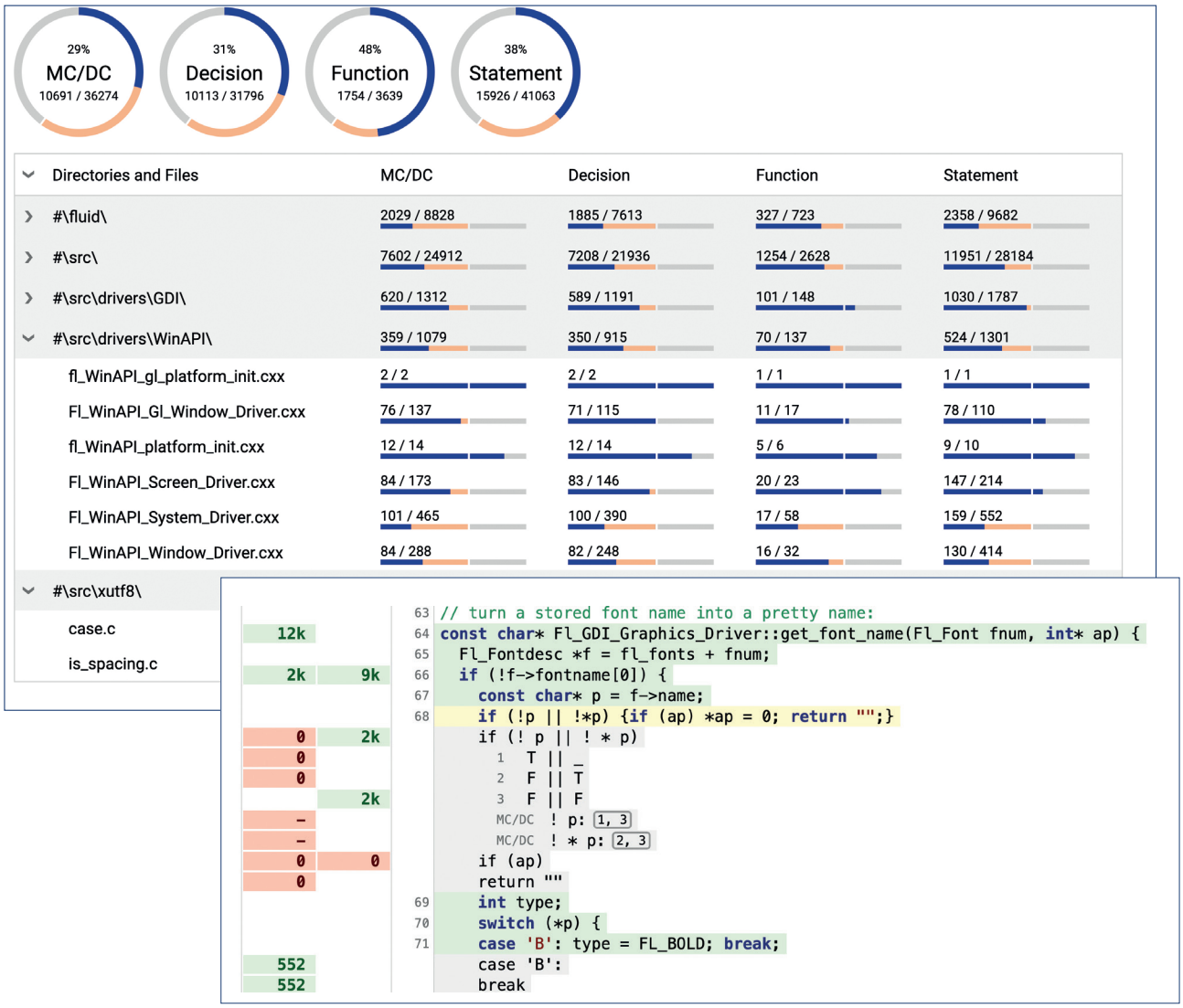
In der dritten Phase erzeugt Testwell CTC++ Coverage-Berichte auf Basis der Rohdaten. Daten von verschiedenen Builds und von verschiedenen Tests können kombiniert werden. Als Outputformat stehen ein strukturierter HTML-Bericht und beliebige textbasierte Austauschformate zur Verfügung.

Funktionale Sicherheit

- ✓ Geeignet für sicherheitskritische Entwicklung gemäß:
  - ✓ ISO 26262
  - ✓ DO 178-C
  - ✓ EN 50716 / EN 50128
  - ✓ IEC 61508
  - ✓ IEC 62304
  - ✓ IEC 60880
  - ✓ ISO 25119 / DIN EN 16590
- ✓ Qualifizierungsunterstützung
- ✓ TÜV-zertifiziert
  - ✓ ISO 26262
  - ✓ IEC 61508
  - ✓ EN 50716
  - ✓ IEC 62304

Coverage-Berichte

Testwell CTC++ bietet einen umfangreichen HTML-Bericht, der an die individuellen Bedürfnisse des Benutzers und an die Art und Größe des Projekts anpassbar ist.



Konfigurierbares Berichtslayout

- ✓ Gewünschte Coverage-Stufen in beliebiger Kombination
- ✓ Wählbare Berichtsebenen mit Drill-Down: Verzeichnisse, Quelltextdateien, Funktionen

Optionale Quellcodeansicht

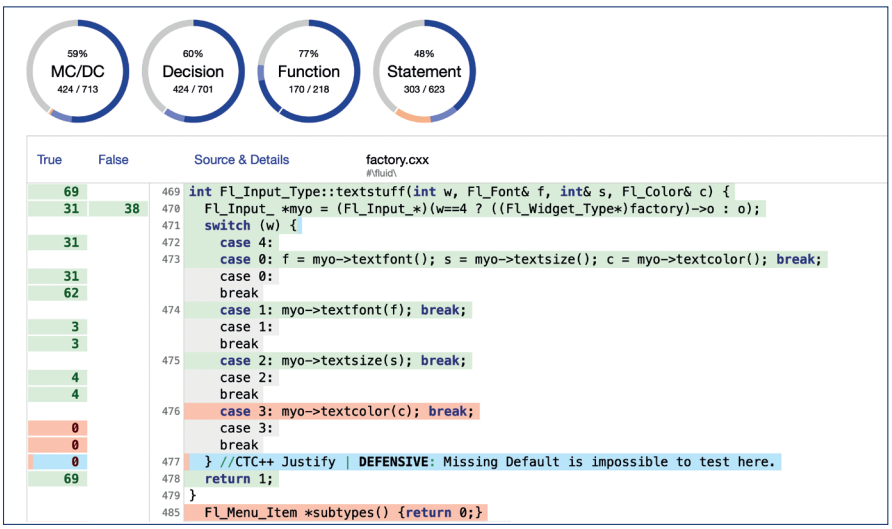
- ✓ Hervorhebung ausgeführter und nichtausgeführter Zeilen
- ✓ Darstellung aller Coverage-Zähler
- ✓ Kompakte Visualisierung der komplexeren Coverage-Maße wie MC/DC
- ✓ Sichtbarkeit fehlender Testfälle

Erklärung fehlender Coverage

Mit Justifications lassen sich Gründe festhalten, weshalb volle Coverage nicht erreicht werden kann.

Testwell CTC++ leitet daraus ab, welche Code-Teile durch eine Rechtfertigung abgedeckt sind.

- ✓ Eigene Kategorisierung von Justifications
- ✓ Erfassung in Kommentaren oder in Begleitdateien
- ✓ Klare Unterscheidung von getestetem und gerechtfertigtem Code
- ✓ Erkennung von Über-Rechtfertigung

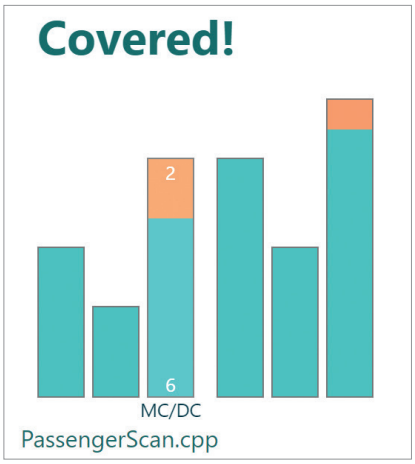


Coverage Daten in jeder Form

Erstellen Sie vorlagenbasiert Berichte in jeder Form. Mit einer einfachen Template-Sprache für den Datenexport unterstützt Testwell CTC++ sowohl strukturierte Berichte wie den HTML-Bericht, als auch den Export einzelner Textdateien.

	A	B	C	D
1		MC/DC		
2	Function	Hits	Total	Ratio
3	hasAdmission	6	8	75%
4	calcPrice	9	10	90%
5	main	6	6	100%
6	TicketApp::showInstruction	3	6	50%
7	TicketApp::switch2BatchMode	2	2	100%
8	TicketApp::ask4Input	3	4	75%
9	TicketApp::reportPrice	3	4	75%

Klassische Austauschformate wie CSV, XML, JSON



Gesamtergebnis als Badge oder auf Dashboards

```
1  # Coverage Report: Coaster as Markdown
2
3  This report was generated at 2024-03-04 09:44:08 using
4      `ctcreport -template example_markdown -o coverage.md`
5
6  ## Coverage in Total
7  - 54% MC/DC (48 / 88)
8  - 85% Statement (60 / 70)
9
10 ## Coverage per Source File
11 ### Directory C:\CoasterCode\
12 #### PassengerScan.cpp:
13 - 75% MC/DC (6 / 8)
14 - 100% Statement (3 / 3)
15
16 #### PriceCalculation.cpp:
17 - 90% MC/DC (9 / 10)
18 - 100% Statement (5 / 5)
```

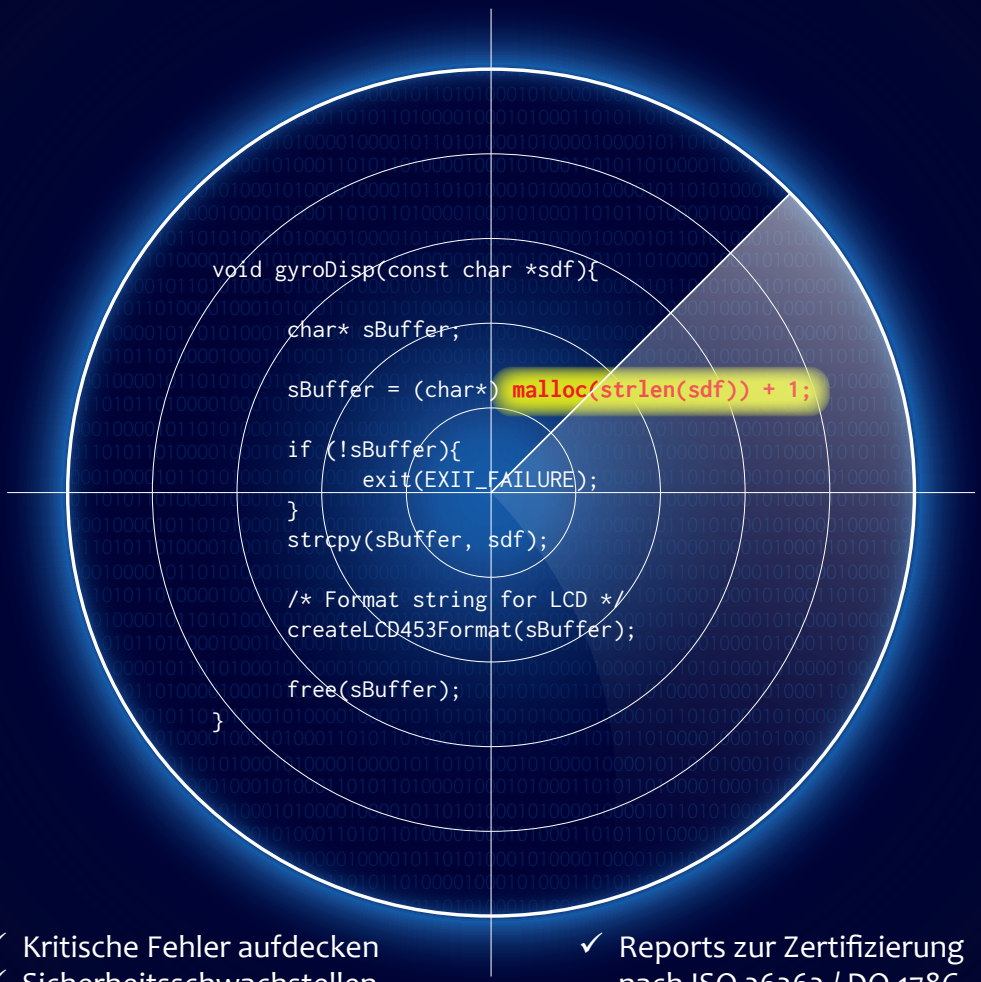
Textberichte, z. B. in Markdown für die einfache Archivierung und Verwaltung in einem Repository

CodeSecure CodeSonar® – Wenn Softwarequalität zum Prinzip erhoben wird

Statische Analyse von Quell- und Binärcode

Da die statische Analyse eine Ausführung der zu analysierenden (Teil-) Applikation nicht erfordert, kann CodeSonar bereits früh im Entwicklungsprozess kritische Softwarefehler aufdecken.

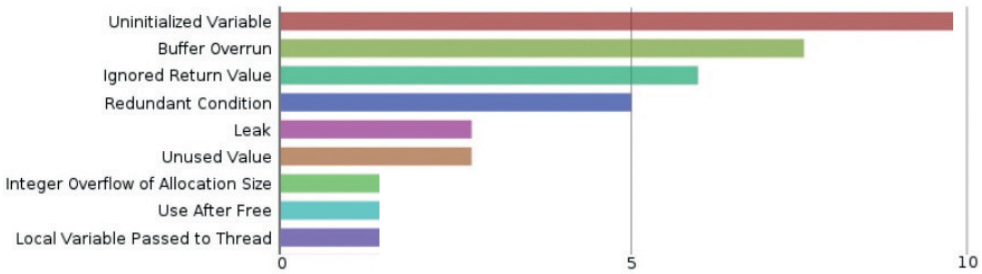
Risiken, die z. B. durch kritische Sicherheitslücken, nichtdeterministische Nebenläufigkeitsfehler oder Speicherlecks entstehen, lassen sich deutlich reduzieren, während gleichzeitig hohe Wartungskosten aufgrund schwer verständlichen Codes vermieden werden.



- ✓ Kritische Fehler aufdecken
- ✓ Sicherheitsschwachstellen eliminieren
- ✓ Codierrichtlinien überprüfen
- ✓ Nebenläufigkeitsprobleme erkennen
- ✓ Reports zur Zertifizierung nach ISO 26262 / DO-178C generieren

Statische Quellcodeanalyse

Als führendes Werkzeug zur statischen Quellcodeanalyse weist CodeSonar im Vergleich zu vielen anderen statischen Analyse-Tools nicht nur eine bessere Fehlererkennung auf, es zeichnet sich zu dem durch eine vergleichsweise geringe Rate an Fehlwarnungen (False Positives) aus.



Statische Binäranalyse

Vielfach werden in Applikationen von Drittanbietern gelieferte Komponenten (Bibliotheken) eingebunden. Da diese oft nur als Binärdateien vorliegen, lassen sich Zweifel an deren Qualität nur schwer ausräumen und die Stabilität und Sicherheit der Gesamtapplikation steht in Frage. CodeSonar for Binaries geht mit seiner Analyse über den Quellcode hinaus und detektiert kritische Fehler auch in Binärdateien.

Hohe Anzahl von Prüfungen

CodeSonars große Anzahl von Checkern ermöglicht das Auffinden einer Vielzahl von kritischen Fehlern.

Sicherheitsprüfungen

Sicherheit von Applikationen spielt durch zunehmende Vernetzung eine immer wichtigere Rolle. CodeSonar führt umfangreiche Überprüfungen Ihrer Software im Hinblick auf Sicherheitsschwachstellen durch und hilft damit Angriffe abzuwehren.

Nebenläufigkeitsprüfungen

Nebenläufigkeitsprobleme wie Race Conditions und Synchronisationsfehler wie Deadlocks deckt CodeSonar durch Verwendung interner Laufzeitmodelle zuverlässig auf.

HINTERGRUNDWISSEN

Unterstützte Programmiersprachen

- ✓ C/C++
- ✓ Java
- ✓ C#
- ✓ Python
- ✓ Kotlin
- ✓ Go
- ✓ Rust
- ✓ .NET
- ✓ JavaSrcipt
- ✓ TypeScript
- ✓ weitere in Vorbereitung

Unterstützte Betriebssysteme

- ✓ Windows
- ✓ Linux
- ✓ NetBSD
- ✓ FreeBSD

Unterstützte Compiler

CodeSonar arbeitet mit nahezu allen aktuellen Compilern problemlos zusammen. Zur Verwendung mit werksseitig nicht unterstützten Typen und Derivaten ist eine Konfigurationsanpassung meist schnell und unkompliziert möglich.

- ✓ ARM RealView
- ✓ Intel C/C++
- ✓ CodeWarrior
- ✓ MacOS
- ✓ Free BSD
- ✓ Microsoft Visual Studio
- ✓ GNU C/C++
- ✓ Clang C/C++
- ✓ Keil
- ✓ Renesas
- ✓ Green Hills
- ✓ Sun C/C++
- ✓ HI-TECH
- ✓ Texas Intruments CodeComposer
- ✓ IAR
- ✓ Wind River

… und viele mehr

Floating Point Warnungsklassen

CodeSonars Analyse arbeitet auf Basis von Fließkommaarithmetik. Es ist damit Werkzeugen, die innerhalb einer Integer-Domäne arbeiten, in vielen Bereichen überlegen.

Gut dokumentierte Ergebnisse

CodeSonar gibt seine Ergebnisse als Warnungen aus, die durch eine gute Dokumentation leicht verständlich sind. Die Ausgaben können klassifiziert und einzelnen Teammitgliedern zur Korrektur zugewiesen werden.

INTEGRATION

Eclipse Integration

Ein mitgeliefertes Plug-in ermöglicht den Entwicklern sich die Analyseergebnisse direkt in Eclipse anzeigen zu lassen. Änderungen können so leicht an der ausgewiesenen Stelle im Code vorgenommen werden.

Microsoft Visual Studio / Visual Studio Code Integration

Eine Integration in Microsoft Visual Studio / Visual Studio Code erlaubt es CodeSonar aus Visual Studio heraus zu starten und die Analyseergebnisse direkt in Visual Studio auszuwerten.

Continuous Integration

CodeSonar arbeitet problemlos mit Hudson und Jenkins zusammen. Zur komfortablen Anbindung an Jenkins ist zudem ein Plug-in verfügbar.

Anbindung an Bug-Tracking Tools

Mittels sogenannter „Warning Processors“ können diverse Bug-Tracking Tools problemlos angebunden werden. Ein Python Beispiel-Skript zur Bugzilla Integration wird mitgeliefert. Es kann leicht an andere Systeme angepasst werden. Für JIRA ist ein Plug-in erhältlich.

SARIF

SARIF (Static Analysis Results Interchange Format) ist ein neuer offener Standard der OASIS (Organization for the Advancement of Structured Information Standards). CodeSonar kann SARIF-Dateien einlesen und Analyseergebnisse im SARIF-Format exportieren.

GitLab, GitHub und Docker

CodeSonar lässt sich durch GitLab-Pipelines steuern und unterstützt GitHub. Eine Ausführung in Docker-Containern ist möglich.

Performance

CodeSonars Checker sind im Hinblick auf hohe Performance optimiert. CodeSonar skaliert gut auf Multicore- und Mehrprozessormaschinen und erlaubt zudem die Verteilung von Analysen auf mehrere Maschinen. So können schnelle Analyseergebnisse auch großer Projekte von mehreren Millionen Codezeilen ermöglicht werden.

Inkrementelle Analyse

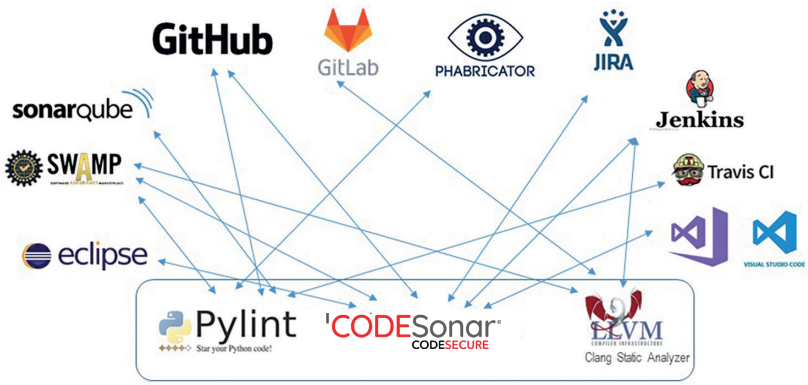
CodeSonars Fähigkeit bei Projektaktualisierungen lediglich die Änderungen unter Berücksichtigung bereits bestehender Analysedaten zu bearbeiten, ermöglicht es die Analysezeiten deutlich zu reduzieren.

Überprüfung auf Einhaltung von Coding Standards

CodeSonar überprüft Applikationen auf die Einhaltung von Coding Standards. Folgende Regelwerke werden unterstützt:

- ✓ MISRA C
- ✓ MISRA C++
- ✓ AUTOSAR C++ Coding Guidelines
- ✓ Power Of Ten
- ✓ JPL
- ✓ SEI CERT
- ✓ DISA STIG
- ✓ OWASP
- ✓ CWE Top 25
- ✓ ISO/IEC TS 17961
- ✓ BARR Naming Conventions

Die Implementierung eigener Regeln ist möglich.



Taint Data Tracking

CodeSonars Taint Data Tracking Analysefunktion identifiziert Sicherheitsschwachstellen, die eine eventuelle Einspeisung von Schadcode in die Applikation ermöglichen und weist die davon direkt oder indirekt betroffenen Abschnitte farblich markiert im Quellcode aus. Präventivmaßnahmen sind dadurch einfach zu treffen.

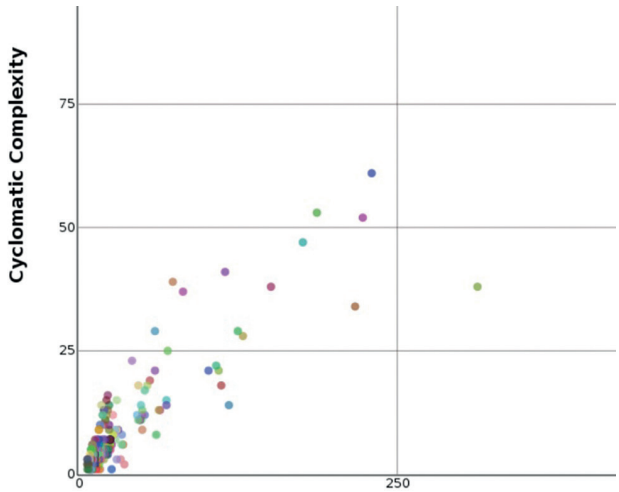
Erhebung von Metriken

Zur Beurteilung der Wartungsfreundlichkeit von Quellcode berechnet CodeSonar eine Vielzahl verschiedener Metriken wie z. B.

- ✓ Zyklomatische Komplexität
- ✓ Halstead Metriken
- ✓ Watson und McCabe
- ✓ HIS

... und viele mehr.

Auch die Erhebung eigener, zusätzlicher Metriken lässt sich implementieren bzw. durch Aggregation bestehender Metriken mittels Konfiguration realisieren.



Nebenläufigkeitsanalyse

CodeSonar deckt zuverlässig Nebenläufigkeitsprobleme auf, wie z. B. Data Races, Dead Locks und blockierende Funktionen in Critical Sections. Folgende Threading-Modelle werden durch CodeSonar unterstützt: Apache Portable Runtime (APR), ARINC 653, CMX-RTX, FreeRTOS, libc, Linux Kernel, Mac OS X, Win32/MFC, Netscape Portable Runtime (NSPR), Qt, ThreadX, uC/OS-III, VxWorks, Win32, wxWidgets. Zudem ist es möglich eigene Threading-Modelle zu erstellen.

Plug-in API

Zur Implementierung eigener Checker stehen gut dokumentierte APIs der Sprachen C, C++, Python, C# und Java zur Verfügung

HINTERGRUNDWISSEN

Zertifizierung

CodeSonar wurde von der **exida** zertifiziert als ein geeignetes Werkzeug zur Erlangung von Zertifizierungen nach:

- ✓ ISO 26262 bis ASIL D, TCL3
- ✓ IEC 61508 bis SIL4
- ✓ EN 50128 bis SW-SIL 4

Präqualifizierungsdokumente helfen Ihnen Zeit und Kosten zu minimieren.

Qualifizierung

Im Hinblick auf eine Zertifizierung Ihrer Applikationen ist, abhängig vom Ergebnis der Klassifizierung, in einigen Fällen (z.B. nach DO 178-B/C) eine Qualifizierung von CodeSonar als Teil der eingesetzten Toolchain nötig. Die dafür erforderlichen Testfälle können zur Verfügung gestellt werden.

Imagix 4D

Das Werkzeug für alle, die Third Party Source Code und Legacy Source Code verstehen müssen

Werkzeug zur Visualisierung und Überprüfung von C, C++ und Java Projekten.
Imagix 4D ist ein Werkzeug, um komplexen, in C, C++ und Java geschriebenen Third Party Source Code und Legacy Source Code zu verstehen, zu dokumentieren und zu verbessern. Imagix 4D automatisiert die Analyse des Kontrollflusses und der Abhängigkeiten. Das Werkzeug deckt Probleme in der Datennutzung und bei Task-Interaktionen auf. Mit Imagix 4D steigern Sie Ihre Produktivität und Qualität und reduzieren Risiken.

Finden Sie Brennpunkte und verbessern Sie die Qualität

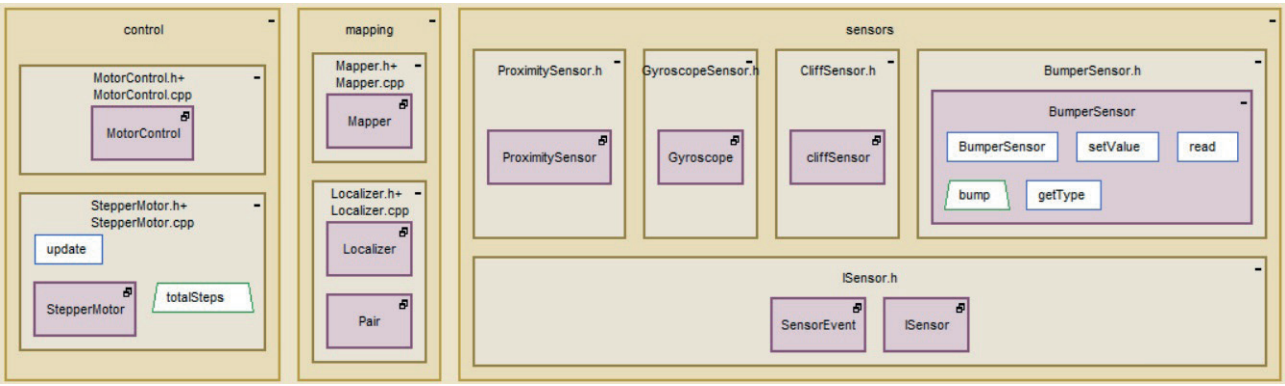
- ✓ Automatische Checker finden Anomalien im Quellcode
- ✓ Die Erhebung diverser Metriken (z. B. zyklomatische Komplexität, Decision Density u. v. m.) lässt Sie kritische Komponenten schnell identifizieren
- ✓ Teilautomatisierte Reviews ermöglichen eine effiziente Durchführung qualitativer Analysen nach CWE oder nach eigenen Vorgaben

Behalten Sie die Kontrolle auch bei umfangreichen Projekten

- ✓ Aussagekräftige Diagramme ermöglichen Einsichten von einer globalen Perspektive bis zu den granularen Eigenschaften einzelner Datentypen
- ✓ Δ-Analysen erlauben eine detaillierte Nachverfolgung von Änderungen zwischen Revisionen
- ✓ Anhand von Architekturdiagrammen ist eine effiziente Prüfung der bestehenden Architektur auf strukturelle Anforderungen problemlos möglich

KI Analyse

Die KI-Analyse in Imagix4D kann in allen Anzeigen und Berichten aufgerufen werden. Die Anfragen können an ChatGPT oder Google Gemini gesendet werden. Imagix stellt im Rahmen seiner Lizenz eine begrenzte Anzahl von Anfragen zur Verfügung.



Beurteilung von Quellcode

Imagix 4D beinhaltet eine Vielzahl von nützlichen Werkzeugen zur Beurteilung Ihrer Software:

Quellcodeanalyse

- ✓ Subsystemarchitekturdiagramme
- ✓ Designstrukturmatrizen (DSM)
- ✓ UML-Klassendiagramme
- ✓ Klassenvererbungsdiagramme
- ✓ Datei-Include-Hierarchieansichten
- ✓ UML-Task-Kollaborationsdiagramme

Delta-Analyse

- ✓ Aufzeigen struktureller Unterschiede zwischen alternativen Versionen Ihres Codes
- ✓ Ausweisen von Kontrollflussdifferenzen in Architekturdiagrammen
- ✓ Markieren logischer Differenzen im Quellcode

Statische Analyse

Aufdeckung funktionaler Probleme wie:

- ✓ Nebenläufigkeitsfehler
- ✓ nicht initialisierte Variablen
- ✓ nicht übereinstimmende Zeiger-Casts
- ✓ verdächtige For-Schleifen
- ✓ Portabilitätsprobleme
- ✓ und vieler weiterer kritischer Fehler

Code Review-Funktionalität

Ergebnisbewertung nach:

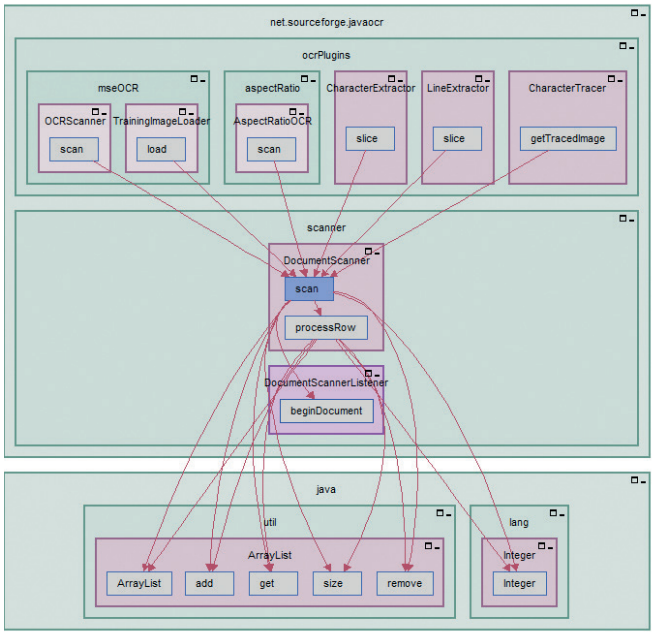
- ✓ Gesamtzahl der Untersuchungen
- ✓ bewertete Untersuchungen
- ✓ als bedenklich eingestufte Untersuchungen
- ✓ als Verstoß eingestufte Untersuchungen

Erhebung von Metriken:

- ✓ HIS
- ✓ McCabe-Zyklomatische Komplexität
- ✓ Wartbarkeitsindex (Welker)
- ✓ Chidamber- und Kemerer-Metriken objektorientiert (6)
- ✓ Klassenkohäsion (Hitz/Mont.)
- ✓ Klassenkopplung
- ✓ Stabilität des Subsystems
- ✓ Kommentarverhältnis
- ✓ Entscheidungstiefe
- ✓ Halstead-Komplexität
- ✓ Knoten (Woodward usw.)
- ✓ Anweisungen, Zeilen usw.

Überprüfung auf MISRA-Konformität

- ✓ MISRA C 2012
- ✓ MISRA C++ 2008
- ✓ AUTOSAR C++14
- ✓ Überprüfung auf Einhaltung des CWE-Standards 2.8 - 3.3



Automatische Dokumentationsgenerierung

- ✓ Beschreibungen und Analysen auf Architektur-, Datei-, Paket-, Namespace- und Klassenebene
- ✓ Grafische Darstellungen von Schnittstellen, Verwendung und externen Abhängigkeiten
- ✓ Über 80 Metriken, darunter objektorientierte und zyklomatische Komplexität

Bis hin zu Implementierungsdetails:

- ✓ Funktionsflussdiagramme
- ✓ Variablenzuweisungen und -zugriffe
- ✓ Querverweise zur Typverwendung

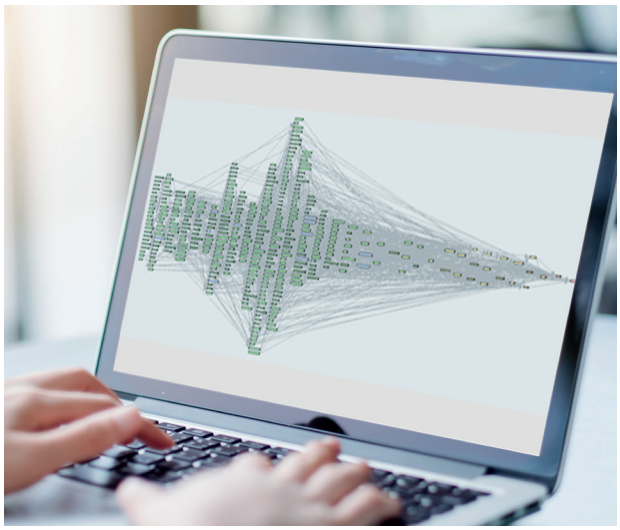
Verbessern Sie den Entwicklungsprozess

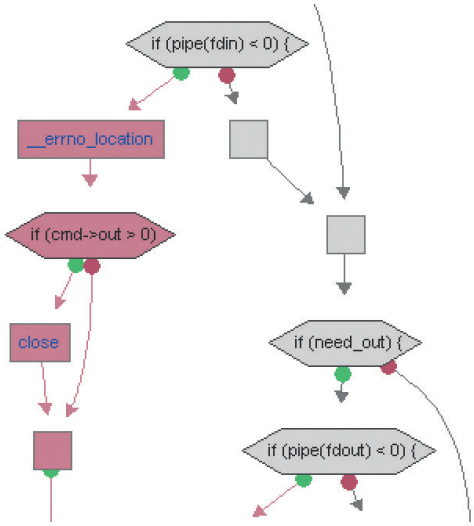
Datenbankabfragen beschleunigen das Auffinden von Informationen zu spezifischen Symbolen

- ✓ Unbekannter Quellcode kann mit Imagix 4D leicht verstanden und bewertet werden
- ✓ Die automatisch generierten Dokumente auf Basis des vorliegenden Quellcodes geben stets den tatsächlichen Ist-Stand des Projektes wieder.

Profitieren Sie von der Testwell CTC++ Integration

- ✓ Visualisierung der Testabdeckung im Kontrollflussdiagramm
- ✓ Zusammenhänge zwischen Tests und Test-abdeckung werden verständlicher, was die Entwicklung passender Testfälle beschleunigt
- ✓ Function- und Call-Coverage-Berichte ergänzen das Portfolio von Testwell CTC++





Testwell CMT++ / Testwell CMTJava

Testwell CMT++
Testwell CMTJava

Softwarekomplexitätsanalyse für C, C++, C# und Java

Testwell CMT++ und Testwell CMTJava sind Tools zur Softwarekomplexitätsanalyse von C, C++, C# und Java Quellcode. Beide Tools analysieren Ihren Quellcode und geben Ihnen sofortige Rückmeldung über Ihre innere Softwarequalität, auch bei größeren Softwareprojekten. Durch eine gute Struktur der Software wird Software-Erosion vermieden. Die Code-Qualität, Wartbarkeit und Testbarkeit wird deutlich verbessert.

Codee

Statische Performance-Prüfung mit Codee für C/C++/Fortran

Codee ist ein statisches Analysewerkzeug, das sehr früh im Entwicklungsprozess sowie in der Testphase eingesetzt werden kann. Das Tool deckt Performanceprobleme im Quellcode auf und kann sie oft automatisch beheben. Schwachstellen können so frühzeitig und kostengünstig beseitigt werden. Als automatische Quellcode-Inspektionsplattform analysiert Codee zeitkritische C/C++- und Fortran-Anwendungen.

Hauptmerkmale

- ✓ Automatisches Optimieren des Quellcodes im Auto-Modus
- ✓ Halbautomatische Optimierung des Quellcodes mit dem geführten Modus
- ✓ Integration mit Compiler-Vektorisierungsberichten
- ✓ Interoperabilität mit CI/CD-Frameworks

Vorteile

- ✓ Schnelle Prüfung
- ✓ Automatisierte Code-Optimierung
- ✓ Schnellere Applikationen
- ✓ Einsparung von Entwicklungskosten
- ✓ Entwicklung von Applikationen, die weniger Energie benötigen

Codee scannt den Quellcode, ohne ihn auszuführen und erstellt einen Bericht für den Entwickler, um den Code schneller ausführen zu können:

- ✓ Das Problem
- ✓ Seinen Ort
- ✓ Warum es die Leistung beeinträchtigt
- ✓ Wie man es beheben kann

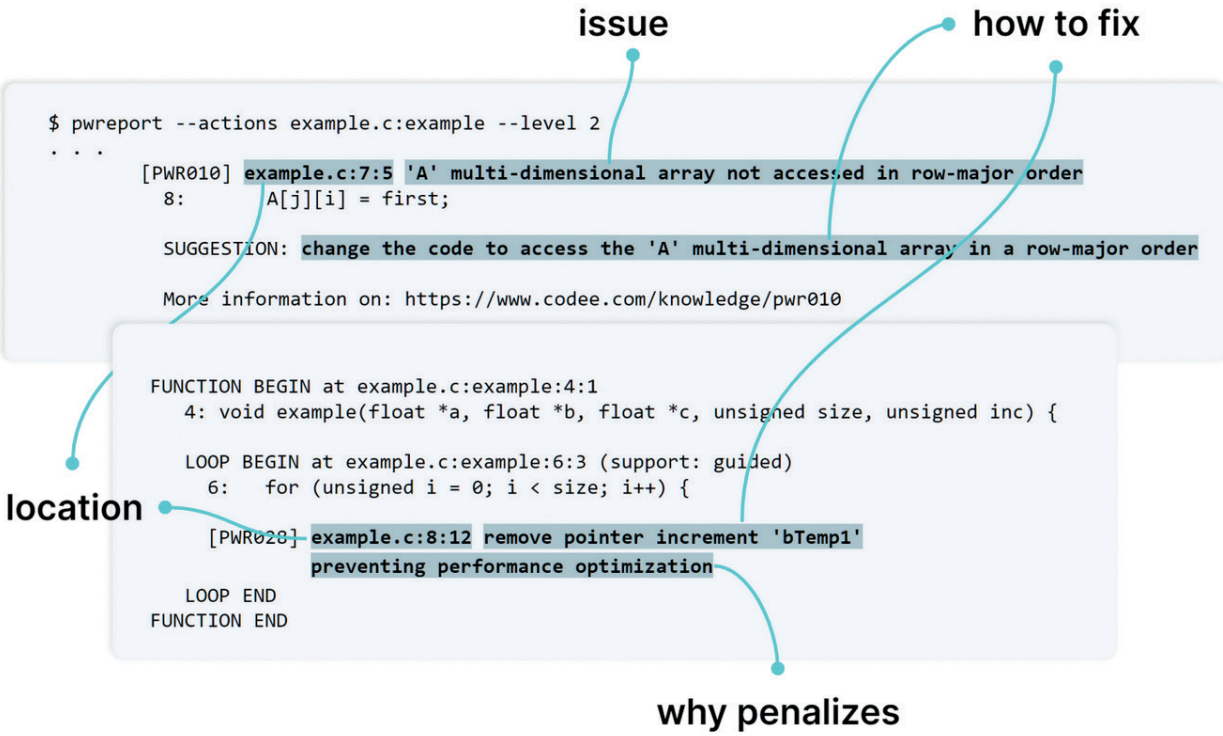
HINTERGRUNDWISSEN

Unterstützte Programmiersprachen

✓ C/C++ ✓ Fortran

Unterstützte Betriebssysteme

✓ Windows ✓ Linux ✓ MacOS



Seminare

Auch in Zukunft entscheiden Software und deren Qualität über den wirtschaftlichen Erfolg eines Unternehmens. Firmen mit guten Prozessen und Strukturen zur Sicherung der Softwarequalität werden ihre Marktposition behaupten und erfolgreich ausbauen können.

Sie wollen sich auf den neuesten Stand bzgl. Qualitätsmaßnahmen und Softwaretestaktivitäten bringen? Ihre Firma plant Aktivitäten im Bereich Softwarequalität aufzusetzen oder auszuweiten? Ihre Mitarbeiter sollen geschult werden, um unsere Softwaretest- und Analyse-Tools in kürzester Zeit produktiv einsetzen zu können? Wir unterstützen Sie bei der Lösung dieser Ziele.

Profitieren Sie von unseren Seminaren und erwerben Sie die notwendigen Kenntnisse um Software effizient und sicher entwickeln und testen zu können.

Weitere Informationen unter
www.verifysoft.com/de_events

Verifysoft

Seit 2003 erstklassige Tools weltweit!

Als eigenständiges und solide wirtschaftendes Unternehmen unterstützt Verifysoft Kunden weltweit mit hochspezialisierter Software bei der Qualitätssicherung ihrer Softwareprodukte. Wir entwickeln unsere Kernprodukte der Marke TESTWELL mit einem handverlesenen, schlagkräftigen Team und bieten zudem erstklassige komplementäre Tools, Seminare und Dienstleistungen an.

Wir verstehen unsere Kunden und sind ihnen – auch langfristig – ein verlässlicher Partner. Bei Verifysoft steht der Mensch im Mittelpunkt. Mit Freude arbeiten wir in einer guten und fairen Arbeitsatmosphäre zum Wohle unserer Kunden, Kollegen und unserem sozialen Umfeld.

Unsere Strategie ist langfristig angelegt. Kundenzufriedenheit ist uns wichtiger als der „schnelle Euro“. Erfolg ist für uns, wenn Kunden und Mitarbeiter zufrieden sind.

Wir freuen uns auf die Zusammenarbeit mit Ihnen.

Über 800 Kunden weltweit

ABB AIRBUS AISIN ALSTOM AMG arbe ARGUS arianegroup arm ASMPT Audi
BAUMULLER Baxter BAIC Group B&R bertrandt BMW GROUP BOMBARDIER BORGWARNER BOSCH
BROADCOM BT Bundesamt für Sicherheit in der Informationstechnik ca technologies Canon CARIAD CATL CISCO COMARCH Continental
CURTIS DAIMLER TRUCK DAIWHA DGA DENSO Digiteq Automotive DLR DONGFENG dSPACE ETAS
EATON Elektrobit EMBRAER Endress+Hauser esa ESO Exonetik framatome GETINGE
GE Giesecke+Devrient Creating Confidence 长城汽车 Great Wall Motors Hama HARMAN HELLA HIMA HITACHI HONDA The Power of Dreams Honeywell
HYDAC HYUNDAI HYUNDAI TRANSYS IAV Infineon intel jenaoptronik kappa KNORR-BREMSE KONGSBERG
KPIT LL Mentor A Siemens Business LIEBHERR McLaren medela Miltenyi Biotec Micron
ModuTram MONTERIS MEDICAL NASA Itelma Electronics NDT Global NISSAN MOTOR CORPORATION NOKIA
NVIDIA NXP OHB Opsys Technologies PHILIPS PHOENIX CONTACT PINTSCH Safety for Rail PORSCHE Raytheon
RENESAS Qualcomm Roche SAFRAN SAMSUNG SCANIA SCHAEFFLER Schindler Schneider Electric
secunet seeingmachines SICK Sensor Intelligence. SIEMENS SKF socionext SONY SPIE
SRMTech ideas@work ST life.augmented ST Engineering TATA TELESPAZIO a LEONARDO and THALES company TenaFe THALES thyssenkrupp
tomtom TTTECH UAES 联合汽车电子有限公司 Vibracoustic VIESSMANN vitesco TECHNOLOGIES VW VOLVO VORWERK
WATLOW Powered by Possibility WEICHAI West wipro YASKAWA ZEISS ZF

Verifysoft TECHNOLOGY

Die Verifysoft Technology GmbH ist auf Entwicklung, Vertrieb und Support von Softwaretest- und Analyse-Tools spezialisiert. Neben den eigenen Testwell-Tools vertreiben wir auch komplementäre Werkzeuge unserer Partner.

Verifysoft wurde 2003 im Technologiepark Offenburg gegründet und ist seither erfolgreich als eigenständiges und von Investoren unabhängiges Unternehmen im Bereich der Softwarequalität tätig.

Mit einem internationalen Team betreuen wir mehr als 800 Kunden in über 45 Ländern. Unsere Entwicklungs- und Supportmitarbeiter haben langjährige Erfahrung im Testtool-Bereich.

Finden Sie Softwaredefekte und -probleme vor dem Release und garantieren Sie höchste Softwarequalität mit Tools von Verifysoft Technology.

Alle hier vorgestellten Tools können kostenfrei evaluiert und jederzeit unverbindlich angefordert werden. Haben Sie Interesse an unseren Tools oder Seminaren?

Nehmen Sie jetzt Kontakt mit uns auf!

© 2026 Verifysoft Technology GmbH
Testwell CTC++, Testwell CMT++, and Testwell CMTJava
are products and trademarks of Verifysoft Technology GmbH, Offenburg (Germany).

CodeSonar is product and trademark of CodeSecure (USA)
Imagix 4D is a product and a trademark of Imagix Corp., San Luis Obispo CA (USA)
Codee is a product and a trademark of Appentra Solutions S.L., A Coruña (Spain)

Verifysoft Technology GmbH, Am Alten Schlachthof 14, 77652 Offenburg (Germany)
Phone: +49 781 127 8118 - 0, info@verifysoft.com

Folgen Sie uns auf

Ausgabe: Februar 2026; Fotos: www.fotolia.de, www.aeroweb-fr.net, www.heinzalbers.org, www.mev.de, NASA, Verifysoft Technology GmbH