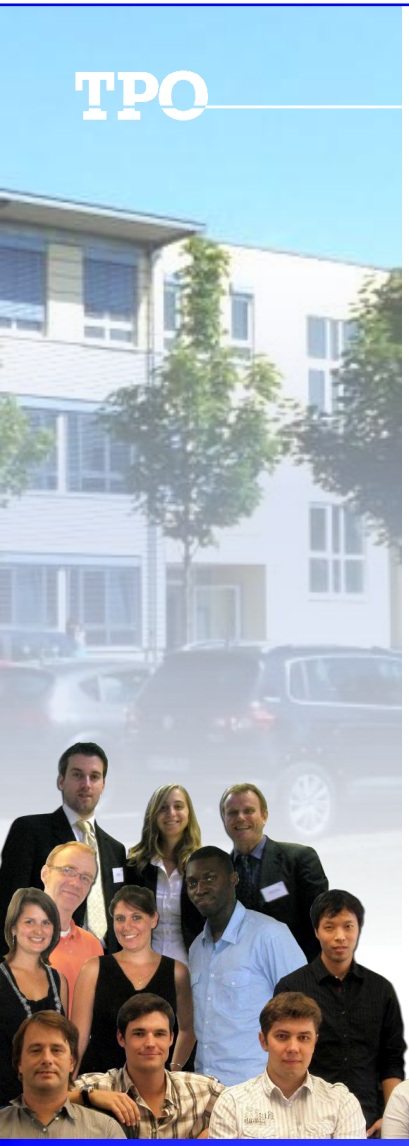
Verifysoft TECHNOLOGY

GRAMMATECH CodeSonar®

Automatisches Aufdecken
von Fehlern und Sicherheitslücken
durch Statische Code-Analyse

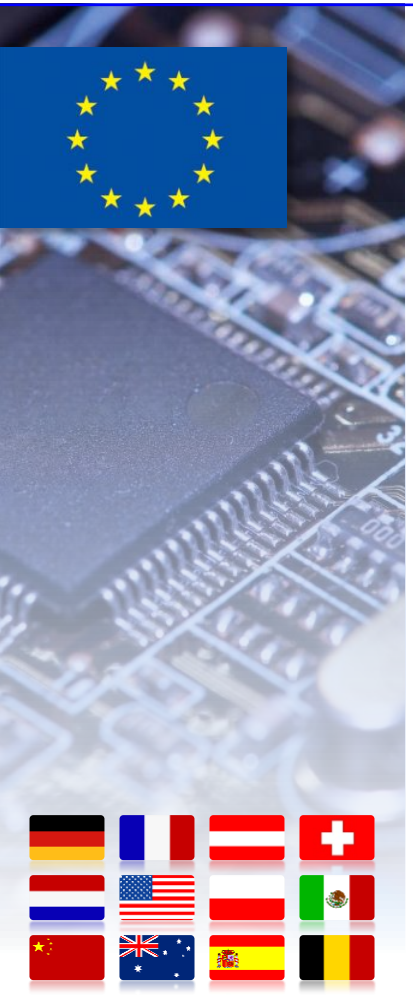


Verifysoft TECHNOLOGY

Technologiepark Offenburg
In der Spöck 10-12
77656 Offenburg, Germany

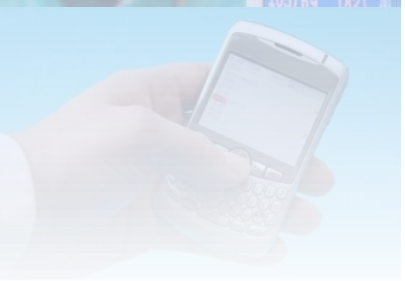
- ✓ Phone: +49 781 127 8118-0 (Deutschland)
- ✓ Phone: +33 3 68 33 58 84 (Frankreich)
- ✓ Fax: +49 781 63 920-29
- ✓ Email: info@verifysoft.com

www.verifysoft.com



Einige Informationen über uns:

- ✓ gegründet 2003
- ✓ Sitz: *TechnologiePark Offenburg* (Deutschland)
- ✓ über 300 Kunden in Europa (Stand: 2013)
- ✓ Distribution, Beratung und Support
  Software-Test- und -Analysetools
- ✓ Distributor für GrammaTech im deutschsprachigen Raum
- ✓ Entwicklung von Software-Testtools
- ✓ Seminare und Trainings



Kunden in folgenden Branchen:

- ✓ Luft- und Raumfahrt
- ✓ Automotive
- ✓ Medizintechnik
- ✓ Automation
- ✓ Transport
- ✓ Energie

… und weiteren sicherheitskritsichen Bereichen mit hohen Qualitätsanforderungen

Testwell	Tampere (Finnland)
CONFORMIQ	Espoo (Finnland) / Saratoga (USA)
GRAMMATECH	Ithaca/New York (USA)
software diagnostics	Potsdam (Deutschland)
Spirula	Paris (Frankreich)
elvior	Tallinn (Estland)
QMETRY	Santa Clara (USA)

Verifysoft-Kunden

Verifysoft TECHNOLOGY

Hauptkompetenzen

Statische Code Analyse von Quellcode und Binarys

Gründer und Mannschaft

- ✓ gegründet 1988
- ✓ Tim Teitelbaum, CEO (ehem. Professor, Cornell University)
- ✓ Tom Reps (Professor, Univ. Wisconsin-Madison)
- ✓ über 50 Mitarbeiter (16 promovierte Experten in der Programmanalyse)

Produkte und Dienstleistungen

- ✓ CodeSonar (statisches Codeanalysetool)
- ✓ GrammaTech-Forschung

Firmensitz, R&D

Ithaca (Bundesstaat New York, USA)

GrammaTech Kunden

Luft- und Raumfahrt

Medizintechnik

Industrial Control / Elektronik

Telecom/Datacom

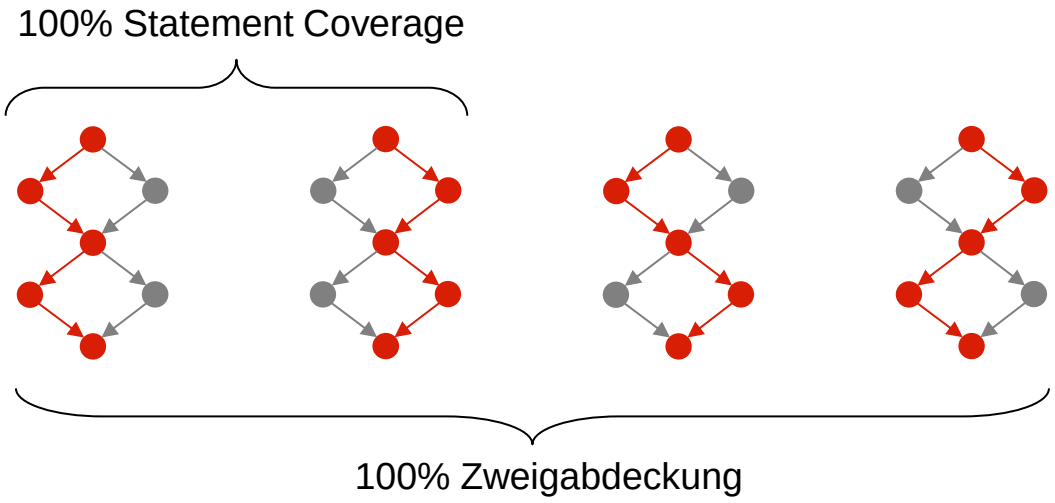
GrammaTech-Forschung Kunden

- Intelligence Advanced Research Projects Activity (IARPA)
- Department of Homeland Security
- Defense Advanced Research Projects Agency (DARPA)
- U.S. Navy: ONR, NAVSEA, SPAWAR, NAVAIR
- National Aeronautics and Space Administration (NASA)
- National Institute of Standards and Technology (NIST)
- Missile Defense Agency (MDA)
- United States Air Force (USAF)
- National Science Foundation (NSF)
- United States Army

GRAMMATECH

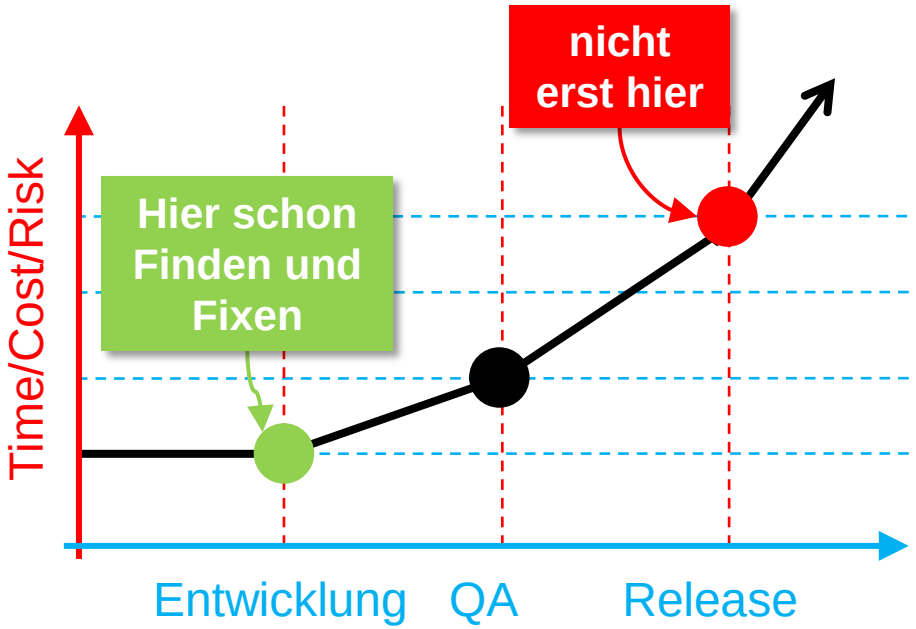
Herkömmliches Testen (dynamische Analyse) ist notwendig, ABER

- ✓ nur so gut wie die Testfälle
- ✓ es ist sehr schwierig alle Pfade durch dynamische Analyse in beschränkter Zeit zu testen



Development-Testing

- ✓ durch den Entwickler schon früh im Entwicklungsprozess
- ✓ je **früher** ein Fehler gefunden wird, desto **kostengünstiger** ist er zu beheben!



Code Sonar® Static Code Analysis

- ✓ Deckt im Vergleich zu anderen statischen Analysetools doppelt so viele **kritische Fehler** auf
- ✓ Verbessert die Sicherheit (Sicherheits-Checker)
- ✓ Nutzt eine fortschrittlichen Analyse-Algorithmus
- ✓ Unterstützt personalisierte Checks
- ✓ Liefert Architekturanalyse

Get Code Sonar® as FREE TRIAL

Betrachtet die Software als Ganzes (übergreifende Analyse)

- ✓ Flow-basierte Analyse von einzelnen Komponenten bzw. des Gesamtprogramms
- ✓ Aufdecken von schwer wiegenden Fehlern und Sicherheitslücken (die zu Abstürzen, Datenverlust und Daten-korruption führen können)

Einfach zu nutzen und erweiterbar

- ✓ arbeitet mit dem bestehenden Build-System / keine Codeänderungen erforderlich
- ✓ Browser-basierte Ausgaben erhöhen die Produktivität
- ✓ weitere Modelle und Checker können hinzugefügt werden

Ergebnis-Datenbank

- ✓ Nutzer kann Warnungen kommentieren
- ✓ Ergebnisse und Kommentare bleiben über verschiedene Builds erhalten

Checker für folgende Defekte

- ✓ Buffer Overrun
- ✓ Null-Pointer Dereference
- ✓ Divide by Zero
- ✓ Uninitialized Variable
- ✓ Data Race
- ✓ Deadlock
- ✓ Free Non-Heap Variable
- ✓ Use After Free
- ✓ Double Free/Close
- ✓ Free Null Pointer
- ✓ Format String Vulnerability
- ✓ Race Conditions
- ✓ Unreachable Code
- ✓ Memory/Resource Leak
- ✓ Return Pointer to Local
- ✓ Mismatched Array New/Delete
- ✓ Race Conditions
- ✓ Unreachable Code
- ✓ Memory/Resource Leak
- ✓ Return Pointer to Local
- ✓ Mismatched Array New/Delete
- ✓ Invalid Parameter
- ✓ Missing Return Statement
- ✓ Dangerous Cast
- ✓ Inconsistent form
- ✓ User-Defined Checks
- ✓ Process Starvation
- ✓ Type Overrun
- ✓ Type Underrun
- ✓ Delete Object Created by malloc

weitere Checker

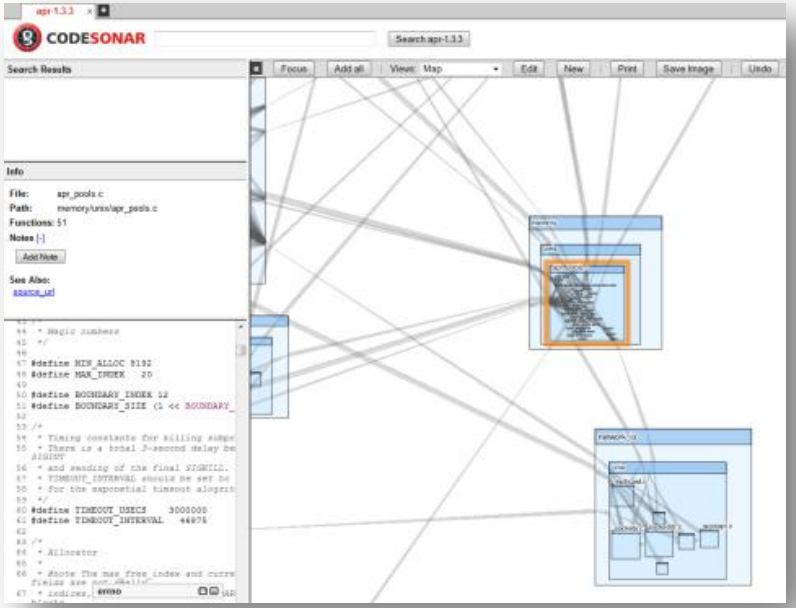
- ✓ Delete Object Created by new[]
- ✓ Free Object Created by new[]
- ✓ Free Object Created by new
- ✓ Redundant Condition
- ✓ Unused Value
- ✓ Useless Assignment
- ✓ Varargs Function Cast
- ✓ Ignored Return Value
- ✓ Null Test After Dereference
- ✓ Double Close
- ✓ TOCTTOU Vulnerability (time-of-check-to-time-of-use)
- ✓ Double Lock
- ✓ Double Unlock
- ✓ Try-lock that will never succeed
- ✓ Misuse of Memory Allocation
- ✓ Misuse of Memory Copying
- ✓ Misuse of Libraries
- ✓ Return Pointer to Free

und viele mehr…

siehe www.verifysoft.com

Architektur-Visualisierung

- ✓ Optimiert die visuelle Inspektion und Analyse
- ✓ Zeigt Abhängigkeiten zwischen den verschiedenen Teilen Ihrer Software
- ✓ analysiert auch große Software > 10 Mio. LOC



Einfache Navigation

- ✓ zoomen, erweiteren etc.
- ✓ auch für große Projekte mit mehreren Millionen Codezeilen

Darstellung der Daten in mehreren Schichten

- ✓ Details werden bei zoomen nach und nach angezeigt
- ✓ Details werden beim herauszoomen abstrakter
- ✓ Eine “Link-Bündlungs“-Option wandelt die “Unordnung” der zahlreichen Verbinungen in eine “High-Level”-Ansicht unter Beibehaltung des Informationsgehalts um.

Visualisierung von ausgewählten Objekten möglich

- ✓ Fokussierung auf beliebige Codeteile möglich
- ✓ unterdrückt Verbindungen, die momentan nicht relevant sind

Metric		Granularities		
		Project	File	Procedure
Basic Counting Metrics				
Blank Lines		x	x	x
Code Lines		x	x	x
Comment Lines		x	x	x
Include file instances		x	,	,
Lines with Code		x	x	x
Lines with Comments		x	x	x
Mixed Lines		x	x	x
Top-level file instances		x	,	,
Total Lines		x	x	x
User-defined functions		x	,	,
Cyclomatic Complexity and Variations				
Cyclomatic Complexity	vG	,	,	x
Essential Complexity	evG	,	,	x
Module Design Complexity	ivG	,	,	x
Integration Complexity	S1	x	,	,
Modified Cyclomatic Complexity	mvG	,	,	x

Halstead Metrics				
Total Operators	N_1	.	.	x
Total Operands	N_2	.	.	x
Distinct Operators	n_1	.	.	x
Distinct Operands	n_2	.	.	x
Halstead Program Length	N	.	.	x
Halstead Program Volume	V	.	.	x
Halstead Program Level	L	.	.	x
Halstead Program Difficulty	D	.	.	x
Halstead Programming Effort	E	.	.	x
Halstead Programming Time	T	.	.	x
Halstead Intelligent Content	I	.	.	x

MISRA-C 2004 und MISRA-C++ 2008

Built-in-Support für folgende Regeln:

- ✓ 1.2
- ✓ 6.3
- ✓ 8.7 and 8.10
- ✓ 9.3
- ✓ 11.1
- ✓ 13.1
- ✓ 14.4
- ✓ 16.2 and 16.10
- ✓ 17.5
- ✓ 19.1, 19.5, 19.6 and 19.7
- ✓ 20.3 and 20.4
- ✓ 21.1

und andere…

Sicherheitsüberprüfungen

Common Weakness Enumeration (CWE) kompatibel

Untertützung von

- ✓ “Build Security In” (BSI)
- ✓ Power of Ten - Rules
- ✓ JPL Rules (NASA Jet Propulsion Laboratory)

GRAMMATECH

Fallstudie: zurückgerufenes Medizingerät

Software Forensics Lab der FDA

- ✓ 200.000 Codezeilen
- ✓ Nach Problemen beim Einsatz des Medizingerätes hat die FDA eine Untersuchung angeordnet, während der der Hersteller den gesamten Quellcode offenlegen musste
- ✓ Zum Aufdecken der Fehler hat die FDA statische Codeanalyse genutzt

Ergebnisse

- ✓ 127 schwere Probleme wurden aufgedeckt

"Flaws in medical coding can kill", Jonathan D. Rockoff, Baltimore Sun, June 30 2008, http://www.baltimoresun.com/news/health/bal-te.fda30jun30,0,912831.story.
"Using Static Analysis to Evaluate Software in Medical Devices", Raoul Jetley and Paul Anderson, Embedded Systems Design, April 2008, Volume 21, Number 4, pp 40-44, TechInsights, http://www.embedded.com/design/207000574.

GRAMMATECH

Fallstudie: Mars Rover

- ✓ NASA JPL Mars Curiosity Rover
- ✓ 2,7 Meter lang, 900 kg

Elektronisches Modul des Rovers

- ✓ RAD750 CPU, 256 kB of EEPROM, 256 MB of DRAM
- ✓ 2 GB Flash-Memory
- ✓ VxWorks multi-tasking RTOS
- ✓ 2 Millionen Codezeilen

Ergebnis: NASA/JPL beschloss GrammTech CodeSonar® zur Softwareanalyse und Unterstützung des “Power of Ten Coding Standards” einzusetzen

Fallstudie: Satelliten-Bodenstation

- ✓ NASA Tracking and Data Relay Satellite System (TDRSS)
- ✓ 9 Satellites und 3 Bodenstationen
- ✓ Ziel 99.9%-ige Verfügbartkeit bei Unterstützung des Starts und der Kommunikation in der Umlaufbahn

Ergebnis:

CodeSonar wurde zur Vermeidung von Softwarefehleren eingesetzt

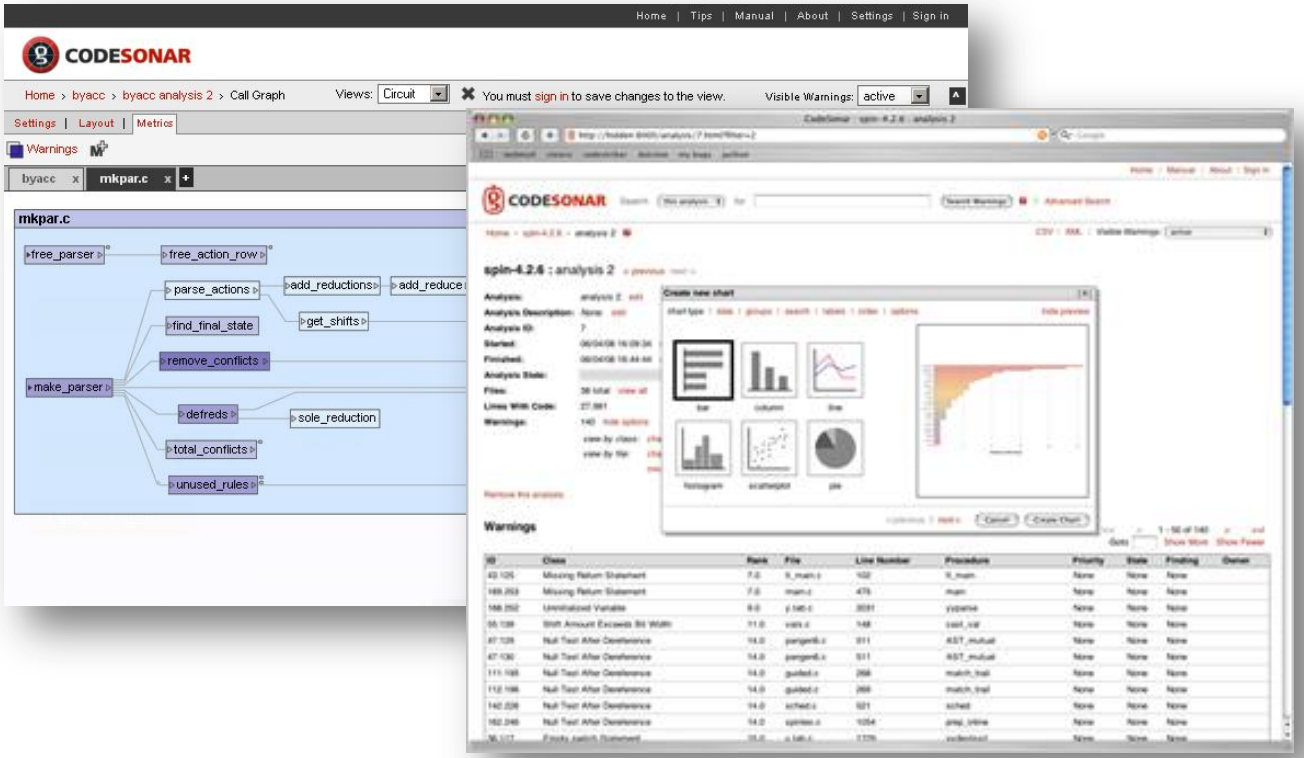
Weitere Fallstudien auf www.verifysoft.com

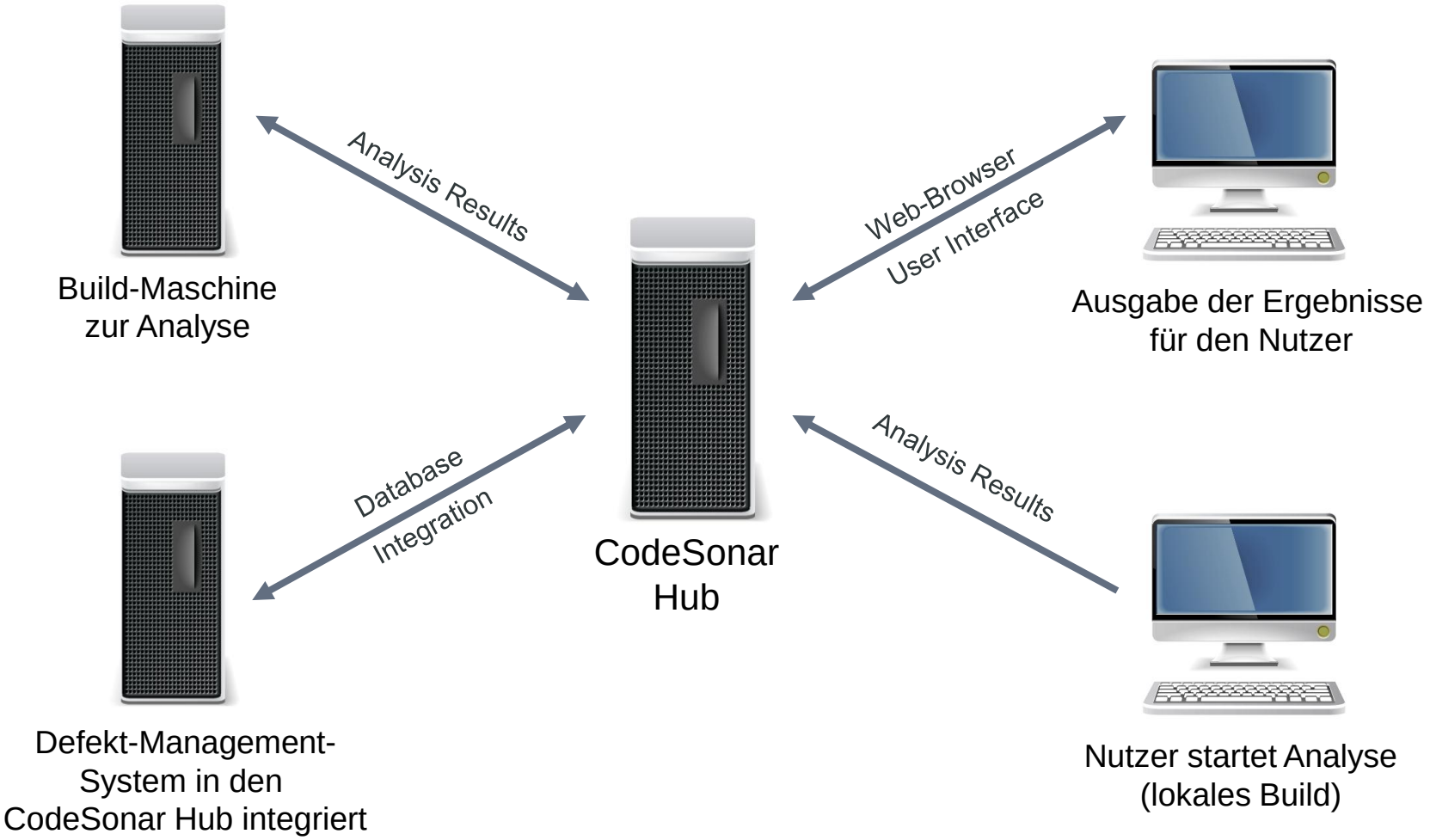
GrammaTech CodeSonar®: Fallstudien

- NASA: Mars Rover Searches for Signs of Life with the Help of CodeSonar
- GrammaTech Helps BCA Ensure Reliability of Life-Saving Mobile App
- FDA Uses GrammaTech to Analyze Recalled Medical Devices
- NASA Uses GrammaTech to Increase Satellite Uptime
- Boston Scientific Streamlines Analysis of Medical Device Software
- CodeSonar Helps Vivante Deliver Reliable GPU Cores On Time
- Critical Link: GrammaTech Ensures High Reliability of DSP Software
- CodeSonar Helps Harvard Apparatus Tackle the Medical Device Market
- CodeSonar Streamlines Certification of High-Security Devices

CodeSonar® Demo

Evaluation verfügbar – überzeugen Sie sich von der Leistungsfähigkeit des Tools an Ihrer eigenen Software




Analysis Results
Web-Browser
User Interface
Database
Integration
Analysis Results
Build-Maschine
zur Analyse
CodeSonar
Hub
Ausgabe der Ergebnisse
für den Nutzer
Defekt-Management-
System in den
CodeSonar Hub integriert
Nutzer startet Analyse
(lokales Build)

Verifysoft TECHNOLOGY

Technologiepark Offenburg
In der Spöck 10-12
77656 Offenburg, Germany

- ✓ Phone: +49 781 127 8118-0 (Deutschland)
- ✓ Phone: +33 3 68 33 58 84 (Frankreich)
- ✓ Fax: +49 781 63 920-29
- ✓ Email: info@verifysoft.com

www.verifysoft.com