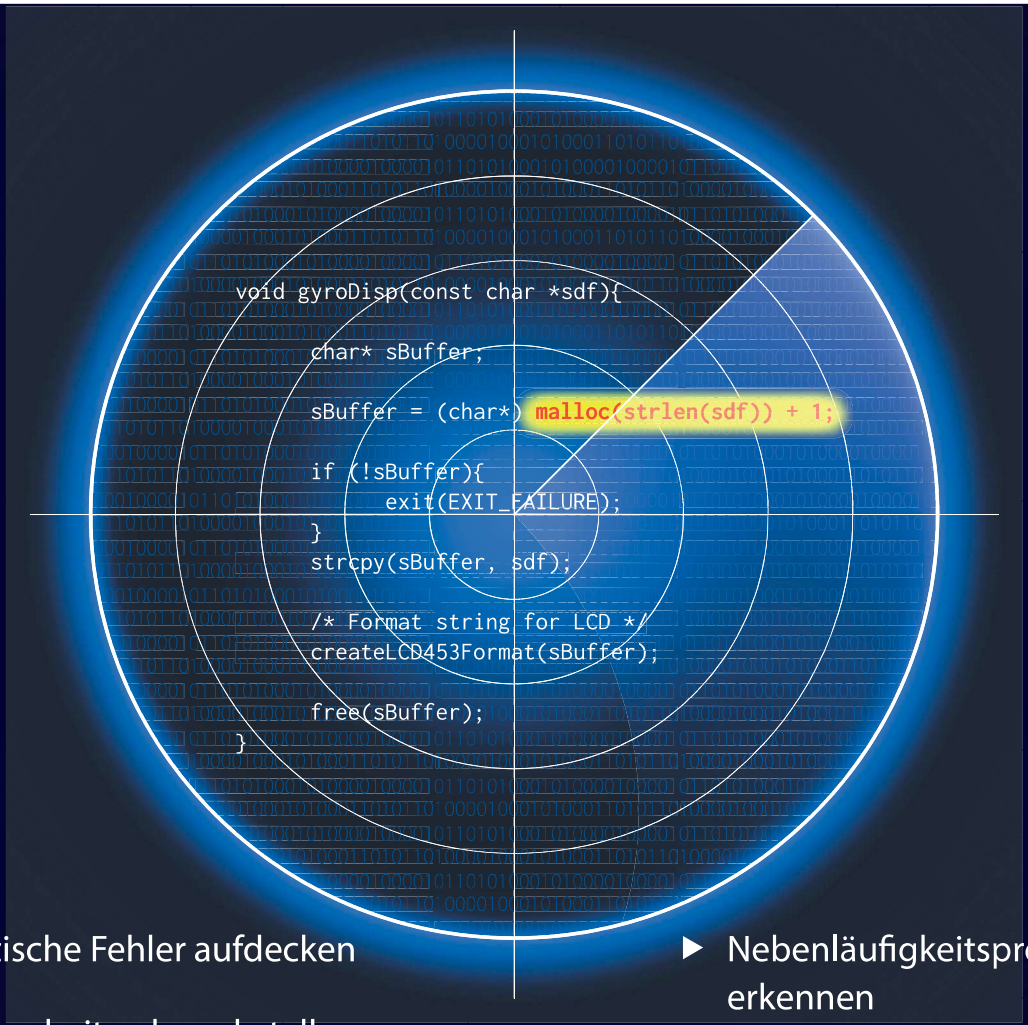
GrammaTech CodeSonar® – Wenn Softwarequalität zum Prinzip erhoben wird

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, hervorgerufen durch z.B. gefährliche Sicherheitslücken, nichtdeterministische Nebenläufigkeitsfehler und Speicherlecks können so minimiert und hohe Wartungskosten, verursacht durch schwer lesbaren Code, 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 zudem 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.

Floating Point Warnungsklassen

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

HINTERGRUNDWISSEN

Unterstützte Programmiersprachen

- C/C++
- C#
- .NET
- Java
- Python

Unterstützte Betriebssysteme

- Windows
- NetBSD
- Linux
- FreeBSD
- Solaris

Unterstützte Compiler

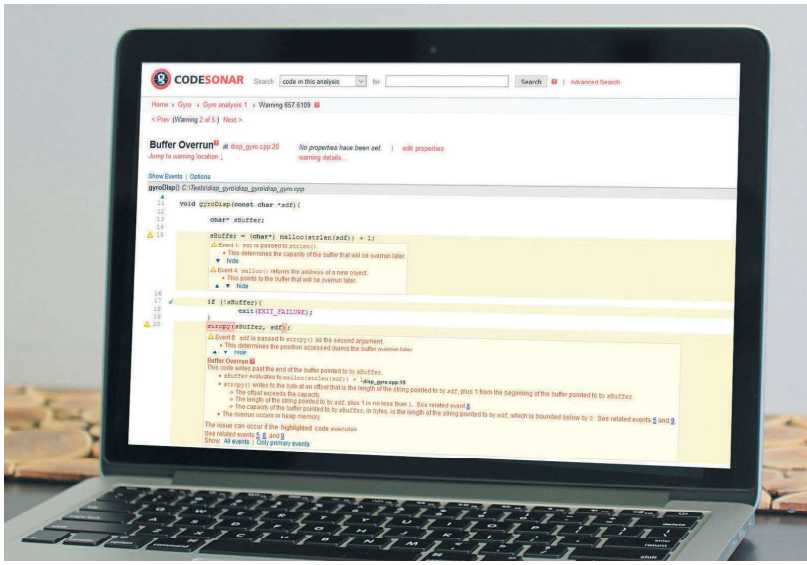
CodeSonar arbeitet mit nahezu allen aktuellen Compilern problemlos zusammen. Zur Verwendung mit werksseitig nicht unterstützten Typen und Derivaten ist eine Konfigurationsanpassung meist schnell und unkompliziert möglich.

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

… und viele mehr

Gut dokumentierte Ergebnisse

CodeSonar gibt seine Ergebnisse als Warnungen aus, die durch eine gute Dokumentation leicht verständlich sind.



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 Integration

Eine Integration in Microsoft Visual Studio erlaubt CodeSonar aus dem Visual Studio heraus zu starten und die Analyseergebnisse direkt im Visual Studio auszuwerten.

Continuous Integration

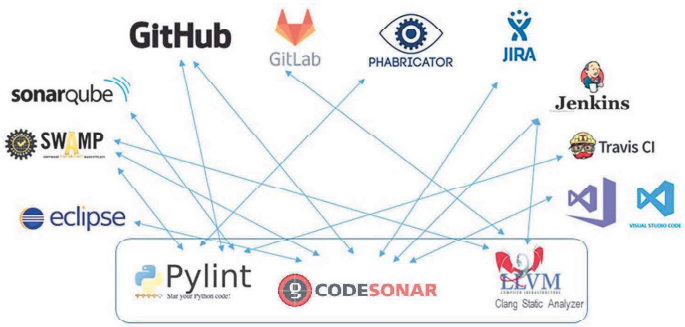
CodeSonar arbeitet problemlos mit Hudson und Jenkins zusammen. Zur komfortablen Anbindung an Jenkins ist zudem ein Plug-in verfügbar

Anbindung an Bug-Tracking Tools

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

SARIF Importer Plug-in

SARIF (Static Analysis Results Interchange Format) ist ein neuer offener Standard der OASIS (Organization for the Advancement of Structured Information Standards). CodeSonars SARIF Importer Plug-in ermöglicht den Import von SARIF-Dateien.



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 bestehenden Analysedaten zu bearbeiten, ermöglicht 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 2012
- MISRA C++ 2008
- AUTOSAR C++ Coding Guidelines
- Power Of Ten
- JPL
- SEI CERT
- DISA STIG
- OWASP

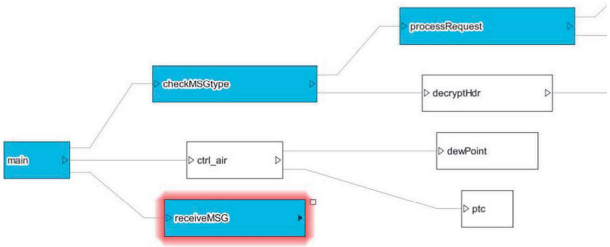
Die Implementierung eigener Regeln ist möglich.

GitLab, GitHub und Docker

CodeSonar lässt sich durch GitLab-Pipelines steuern und unterstützt GitHub.
Eine Ausführung in Docker-Containern 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 sowohl farblich markiert im Quellcode als auch in einer graphischen Repräsentation im Diagramm aus. Präventivmaßnahmen sind so einfach zu treffen.



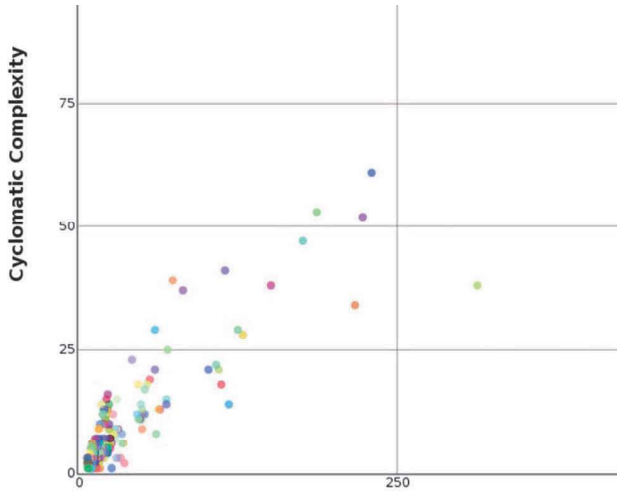
Erhebung von Metriken

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

- Modifizierte zyklomatische Komplexität
- Zyklomatische Komplexität
- Halstead Metriken
- Watson und McCabe

...und viele mehr.

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



Plug-in API

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

HINTERGRUNDWISSEN

Zertifizierung

CodeSonar wurde von der exida zertifiziert als ein geeignetes Werkzeug zur Erlagung 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.

Für sicherheitskritische Anwendungen ist eine tiefgehende statische Analyse unverzichtbar.