SOLVIMUS
METERING SOLUTIONS

ANALYSIEREN STATT DEBUGGEN

13.09.2018

Verifikation der Software durch statische Codeanalyse

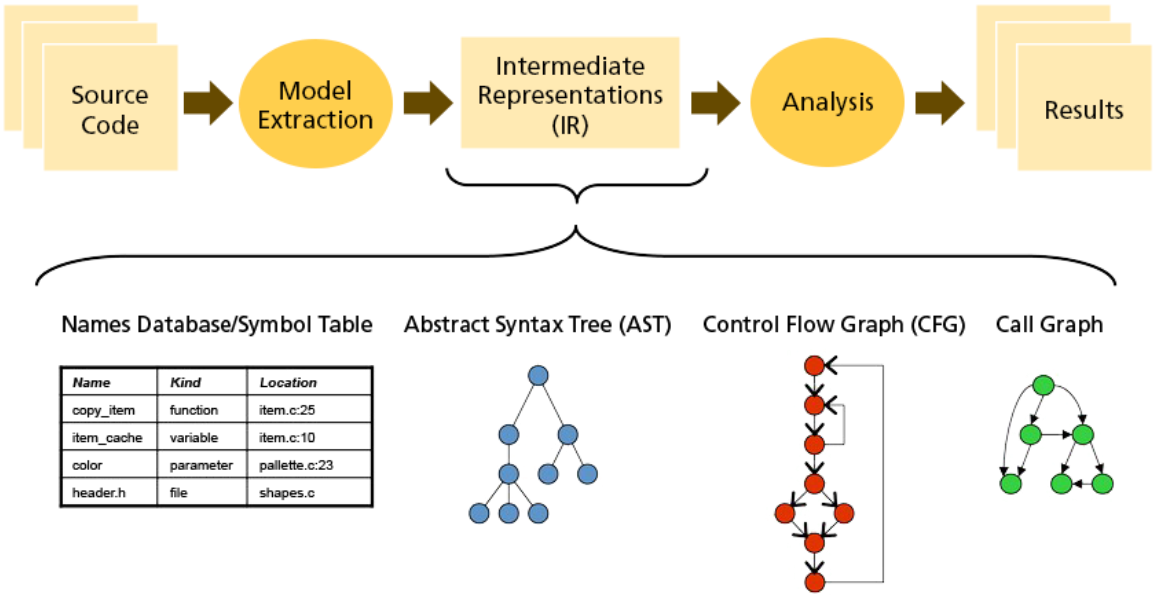
Jeder der schon einmal im Bereich der Softwareentwicklung tätig war, weiß, dass das Auffinden von Fehlern oft mehr Zeit beansprucht als das Programmieren des eigentlichen Quellcodes.

Einige Fehler im Code machen sich erst im späteren Projektverlauf bemerkbar. Meistens müssen diese Fehler dann durch zeitraubendes Debugging gefunden werden. Besonders bei sporadisch auftretenden Fehlern ist dies teilweise nur schwer möglich.

Durch die statische Codeanalyse können Fehler zu einem sehr frühen Zeitpunkt automatisiert gefunden werden, bevor diese dann zur Laufzeit auftreten.

Unsere Metering Produkte werden im Feld über längere Zeiträume betrieben. Gerade hier ist es wichtig, dass die Software robust und hinsichtlich Fehler geprüft ist, die erst mit großer zeitlicher Verzögerung auftreten. Wir setzen hierbei unter anderem auf das **statische Codeanalysetool CodeSonar.**

Hier kurz der Workflow skizziert:



Der zu analysierende Quellcode wird hierbei nicht auf dem Zielsystem ausgeführt, sondern auf dem Entwicklungs- oder Testsystem durch ein Analysetool anhand generischer Regeln geprüft. Anders als bei einem Black-Box-Test (wie beispielsweise ein System- oder Unittest) handelt es sich hierbei um einen White-Box-Test, da der Quellcode in offener Form geprüft wird.

Die statische Codeanalyse prüft beispielsweise, dass alle verwendeten Speicherbereiche gültig sind und die Software diese korrekt nach der Verwendung wieder freigibt. Ebenso wird der parallele Zugriff durch nebenläufige Prozesse und die dazugehörigen Zugriffsmechanismen (Mutexes) geprüft. Es zeigt innerhalb der Software verwendete ungünstige Implementierungen an, wie z. B. zu komplexe Funktionen oder sich selbst aufrufende Funktionen (Rekursionen), die eine nachträgliche Verifikation und damit die Wartung und Pflege erschweren.

Wir bewerten jede Warnmeldung, die durch die statische Codeanalyse generiert wird, prüfen die angezeigten Implementierungen genauer und setzen ggf. erforderliche Korrekturen innerhalb der Software um.

Die Prüfung umfasst alle unsere Metering Produkte einschließlich kundenspezifisch angepasster Softwareversionen. Hilfreich hierbei ist, dass gleiche Warnmeldungen aus mehreren Softwarevarianten automatisch durch das statische Codeanalysetool zusammengefasst werden.

Eine **andere Prüfungsvariante** hierzu bietet auch der **von uns verwendete Compiler (GCC)** selbst, welcher bei der Umsetzung der linuxbasierten Produkte zum Einsatz kommt. Durch die Aktivierung aller Warnmeldungen („all“ und „extra“) erfolgt u. a. eine Prüfung, ob der Vergleich vorzeichenbehafteter und vorzeichenloser Variablen korrekt erfolgt oder ob bei der Verarbeitung der Zustandsvariablen (Enumerationen) alle Zustände abgebildet sind. Zusätzlich prüft der Compiler die Software auch genauer hinsichtlich Stacktiefe oder Speicheranordung (Alignment), welche bei Linux/ARM-Prozessoren zu berücksichtigen ist.

Diese Analysen ermöglichen eine frühzeitige Erkennung und Behebung von Softwarefehlern, so dass diese nicht erst innerhalb von nachgelagerten Tests bzw. im Feld mit größeren Zeit- und Kostenaufwand behoben werden müssen.