

Statische Softwareanalyse und ihre Vorteile für medizinische Software

Software spielt bei modernen medizinischen Geräten eine zunehmend entscheidende Rolle, wie etwa bei EKG-Analyse, Laser-Chirurgie und intravenösen Abgabesystemen.



In moderner Medizintechnik wie Protonentherapiegeräten stecken Millionen von Codezeilen

Angesichts des sicherheitskritischen und teilweise lebenserhaltenden Charakters muss die integrierte Software einwandfrei funktionieren. Nur ein einziger Fehler könnte unter Umständen zu schwerwiegenden Gesundheitsschäden oder sogar zum Tod des Patienten führen. Deshalb liegt doppeltes Augenmerk auf der Sicherheit dieser Software.

Einsatz einer automatisierten statischen Codeanalyse

Die FDA U.S. befürwortet, dass Hersteller von medizinischen Geräten eine detaillierte Überprüfung und Validierung der Software für die Anwendungen ausführen müssen. Bisher beschränkten sich die Methoden für eine solche V&V auf Prüf- und Code-Review. Diese Techniken können zwar eine große Anzahl an Fehlern aufdecken, gewähren aber keine in diesem Segment erforderliche 100prozentige Sicherheit. Wegen der Komplexität der Analysen der eingesetzten Soft-

ware empfiehlt die FDA in ihren Leitlinien jetzt auch den Einsatz einer automatisierten statischen Codeanalyse zur Prüfung aller Programm-Pfade.

Betriebsfertige Lösungen

sind bereits am Markt erhältlich. So ermöglicht Coverity Prevent eine lückenlose Überprüfung der Software in medizinischen Geräten. Es prüft die Pfade und Werte in C, C++, C# und Java zu 100 Prozent und erzielt eine extrem niedrige Rate von False Positive Results. Die höchst skalierbare Technologie kann innerhalb von wenigen Stunden Millionen an Codezeilen analysieren, dabei lässt sie sich problemlos in die bestehende Umgebung integrieren, ohne dass dazu der Code, die Entwicklungsumgebung oder der Entwicklungsprozess unterbrochen werden muss. Die Unternehmen können auswählen, ob sie die Codeanalyse zentral oder auf einzelnen Desktoprechnern ausführen möchten. Mit dem kundenspezifisch anpassbaren Defect Mana-

ger kann das Entwicklungsteam kritische Daten austauschen und Fehler gemeinsam beheben.

Je komplexer Software ist, desto größer wird ihr Potenzial für verborgene Fehler. So enthalten beispielsweise moderne Infusionsgeräte hunderttausend Codezeilen und Protonentherapiegeräte über eine Million Codezeilen. In derartig umfangreichem Code wird der Fehlercheck zur besonderen Herausforderung.

Das Problem könnte sich sogar in die Zukunft verschleppen, denn mit dem „gläsernen Menschen“ auf dem Vormarsch werden bisher eigenständige Geräte nach und nach in Systemen zusammengeschlossen. Dadurch summiert sich der Code auf das Millionenfache, um Operations- und Patientendaten per Mausklick zu übertragen. Ganz offensichtlich erfordert es ein neues Paradigma zur Unterstützung der Sicherheit, um das Auftreten eines Ausfalls oder einer Funktionsstörung in diesen Systemen zu minimieren.

Statische Tools

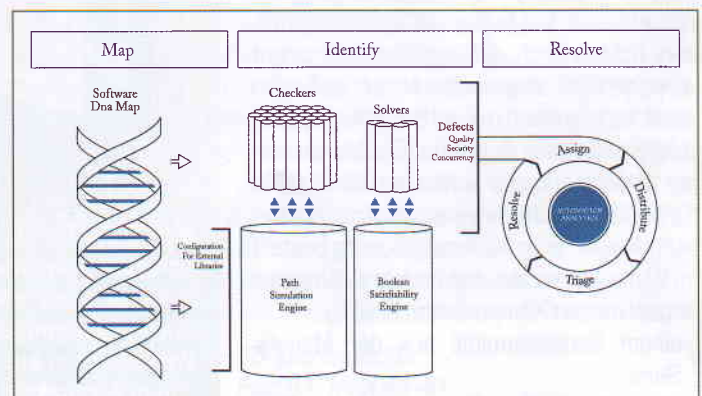
Genau hier bieten statische Tools den Entwicklern von medizinischer Software mehrere Vor-

teile: Sie können schwer auffindbare Fehler bereits früh im Lebenszyklus einer Anwendung aufdecken, wo sie sich noch am einfachsten beseitigen lassen.

Reduziert die manuelle Fehlersuche

Zusätzlich zur deutlich besseren Vorhersagbarkeit, wann Software zur Auslieferung und Inbetriebnahme ausgereift ist, reduziert die statische Analyse die mühsame manuelle Fehlersuche beträchtlich, so dass viel mehr Zeit für das Programmieren und Innovationen übrig bleibt. Damit bieten diese Tools letztendlich wertvolle Unterstützung für Unternehmen, unerwünschte Auswirkungen zu vermeiden, die gerne genau dann auftreten, wenn Probleme erst knapp vor dem Freigabetermin oder, noch schlimmer, nach der Auslieferung behandelt werden. Damit ersparen sie Auswirkungen auf das Geschäft bzw. Unternehmen, auf die Kunden oder die Endanwender der medizinischen Einrichtungen.

► Coverity
www.coverity.com



Coverity Prevent ermöglicht eine lückenlose Überprüfung der Software in medizinischen Geräten.

FDA empfiehlt statische Quellcodeanalyse für höhere Softwarequalität bei medizinischen Geräten



meditronic-journal sprach mit Ben Chelf, Unernehmensgründer und CTO von Coverity Inc., www.coverity.com

meditronic-journal:

Warum ist die statische Codeanalyse für medizinische Software wichtig?

Ben Chelf: Im Laufe der letzten Jahre hat Software verstärkt Einzug in medizinische Geräte gehalten, angefangen bei Herzschrittmachern, über EKGs, Laserchirurgie, Protonentherapiegeräten bis hin zu intravenösen Abgabesystemen, die ihre Dosierung an die Patienten anpassen. Wichtig dabei ist, dass die Software in jedem Fall fehlerfrei läuft – ein einziger Fehler könnte sich gravierend auswirken. Die Herausforderungen moderner Software lauten: Update-Zyklen, mehr Funktionalität und höhere Komplexität. Dazu kommen wachsender Umfang und Wichtigkeit bzw. Bedeutung der Software. Schließlich wächst das Fehlerpotenzial mit der Komplexität der Software.

Die Überprüfung der bisweilen riesigen Codemengen nach Defekten ist eine immense Herausforderung, die mit der Zusammenlegung der Daten in Netzwerken weiter ansteigt. Zur Unterstützung der Sicherheit von Software sind neben den existierenden statischen Softwareanalysetools weiterentwickelte Methoden erforderlich.

meditronic-journal:

Wie antwortet der Markt darauf?

Ben Chelf: Grundsätzlich empfiehlt die U.S. FDA, dass die Hersteller für jede in ihren Geräten eingesetzte Software eine detaillierte Überprüfung und Validierung ausführen. Sie umfassen Testläufe und die manuelle Code-Überprüfung. Allerdings gibt es bei diesen Methoden keine Garantie dafür, dass sie alle Fehler in der Software auffinden. Das erfordert einen neuen Ansatz – nämlich die statische Codeanalyse. In den letzten Jahren hat sie sich deutlich weiter-

entwickelt und enthält nun in ihrer dritten Generation neue ausgeklügelte Technologien wie z.B. Path-Flow-Analyse und Inter-procedural-Analyse. Dadurch erzielt sie eine bisher unerreichte Qualität der Ergebnisse und punktet bei Integration und Arbeitsablauf. Deshalb hat die FDA ihre Empfehlungen um solche statischen Analysetools erweitert.

meditronic-journal:

Womit punktet die dritte Generation der statischen Analyse?

Ben Chelf: Mit Hilfe der statischen Analyse lässt sich eine Fehlerüberprüfung sowohl bei der Programmierung der Softwaremodule als auch bei deren Zusammenführung im zentralen Entwicklungsprozess ausführen. Indem die statische Analyse jeden möglichen Pfad der Software evaluiert, findet sie Fehler schon früh im Entwicklungsprozess – also genau dann, wenn sie am kostengünstigsten zu beheben sind. Je weiter der Entwicklungsvorgang fortschreitet, desto teurer werden Defekte.

Der Aufwand zur Fehlerbehebung nach der Inbetriebnahme der Software kann bis zu 100mal höher sein, als hätte man den Fehler im Entwicklungsprozess gefunden. Nach der Inbetriebnahme eines Geräts im Markt kann er sogar bis ins 1000fache steigen, abgesehen von Image- und Personenschaden. Mit den Tools der dritten Generation der statischen Analyse lassen sich diese Kosten vermeiden, da sie noch präziser als die Vorgänger laufen. Fehler wie z.B. Pufferüberläufe, Uninitialized Variable, Speicherlecks, oder Null-Pointer Reference werden schon früh im Entwicklungszyklus aufgedeckt.

meditronic-journal:

Worauf sollte laut FDA bei der Auswahl eines statischen Codeanalysetools geachtet werden?

Ben Chelf: Speziell für den Einsatz von Software in medizinischen Geräten unterteilt die FDA in ihren Empfehlungen in ‚quantitative‘ und ‚qualitative‘ Kriterien. Die quantitativen sind messbar für vorgegebene Analysen, dazu zählen Genauigkeit (Anzahl der False-Positive-Ergebnisse), Vollständigkeit (Anzahl der tatsächlichen Fehler) und Durchsatz (wie schnell läuft die Analyse). Zusätzlich zu diesen harten Faktoren beinhalten die ‚qualitativen‘ Kriterien Konfigurierbarkeit, Integration in den bestehenden Entwicklungsprozess, durchgehende Überprüfung nach Fehlern, Priorisierung der Fehler durch Entwickler, kundenspezifische Fehlerdefinition sowie multi-threaded Checking. Sie lassen sich nicht so einfach erfassen, sind aber für die Bewertung eines statischen Analysetools ebenso wichtig.

Grundsätzlich hängt eine erfolgreiche Implementierung eines statischen Analysetools in den Entwicklungsprozess von der An- bzw. Übernahme seitens der Entwickler und der problemlosen Integration in den bestehenden Entwicklungsablauf ab. Vertrauen die Entwickler dem Analysetool nicht (verweist nicht auf richtige Fehler oder zeigt zu viele „False Positive“- oder „False Negative“-Ergebnisse), kann dies das sofortige Absetzen des Tools bewirken. Die Kombination aus Präzision und Relevanz ist Voraussetzung.

meditronic-journal:

Was zeichnet den Lösungsansatz von Prevent aus?

Ben Chelf: Mit Prevent läutet Coverity die 3. Generation der statischen Codeanalyse ein. Es ist eines der am Markt führenden Tools zur Auffindung und Behebung von Sicherheitslecks und Qualitätsfehler in Software bereits im frühesten möglichen Entwicklungszyklus. Prevent überprüft Codebasen (C/C++/C# und Java) automatisch. Dabei lässt sich das inhaltsreiche Tool problemlos in die bestehende Umgebung integrieren, ohne dass dazu der Code, die Entwicklungsumgebung oder der Entwicklungsprozess unterbrochen werden muss. Es findet gefährliche Defekte bei Sicherheit und Qualität sowie Gleichzeitigkeitsfehler mit einer sehr niedrigen Rate an „Falsch-Positiven“ Ergebnissen.

Die Arbeitsweise von Prevent lässt sich in die drei Phasen MAP – IDENTIFY und RESOLVE einteilen. Die Schlüsselkomponente von Prevent ist seine so genannte, patentierte „Software DNA Map“ - eine äußerst präzise Darstellung von Quellcode, Entwicklungsumgebung bis hin zu den Links. Im zweiten Schritt (IDENTIFY) folgt die detaillierte Analyse und Identifizierung von Fehlern – hier kommen Path Simulation Engine und die Boolean Satisfiability Engine zum Einsatz, deren Solver und unabhängige Checker individuell eingerichtet werden können. Die damit gefundenen Fehler werden im letzten Step (RESOLVE) automatisch ausgewählten Entwicklern zugeordnet, die für deren Behebung verantwortlich zeichnen. Dazu erfolgt eine genaue Darstellung der Codebasis mit Pfaden und Ereignissen. Dank seiner Präzision und Innovation setzen weltweit bereits über 500 Unternehmen auf Prevent, um Codesicherheit und -qualität zu gewährleisten. Zu den Anwendern zählen die Entwickler selbst, aber auch Software-Architekten bis hin zu Managern und Administratoren.

meditronic-journal:

Wir danken Ihnen für das Gespräch.