software quality lab
improve your quality

Quality Knowledge Letter

www.software-quality-lab.at

Ausgabe 2009 / 3

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

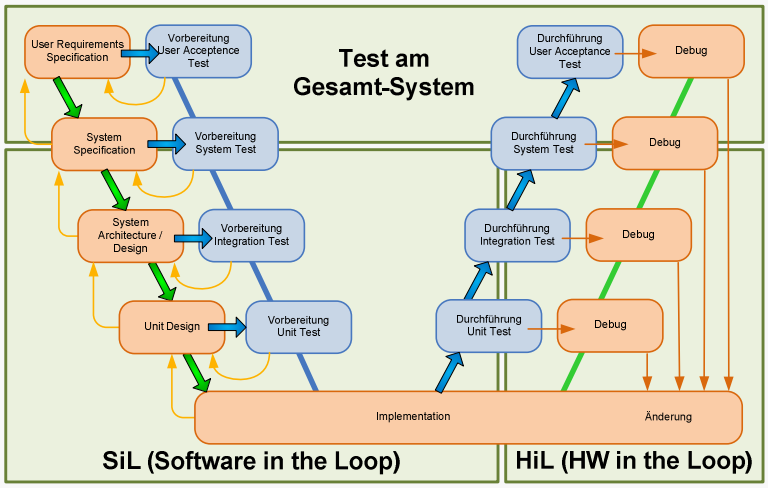
Black-Box-Testautomatisierung für Embedded-Systems

In dieser Ausgabe:

SiL (Software in the Loop) und HiL (Hardware in the Loop)

➲ SiL (Software in the Loop)

Bei der Methode Software in the Loop (SiL) wird ein erstelltes Modell der Software in einen für die zu testende Zielhardware verständlichen Code umgewandelt. Der Code wird auf dem Entwicklungsrechner zusammen mit dem simulierten Modell ausgeführt, anstatt wie bei HiL auf der Zielhardware zu laufen. Es handelt sich dabei also um eine Methode, die *vor* dem HiL anzuwenden ist. Vorteile von SiL sind unter anderem, dass die Zielhardware noch nicht feststehen muss, und dass die Kosten aufgrund der fehlenden Simulationsumgebung weitaus geringer ausfallen. Das Modell kann ev. auch beim HiL weiter verwendet werden, und somit sind die einzelnen Testläufe miteinander vergleichbar.



Schematische Anwendungsbereiche von HiL und SiL im W-Modell

➲ HiL (Hardware in the Loop)

Dabei wird das zu steuernde Gesamtsystem (z. B. Auto) über Modelle simuliert, um die korrekte Funktion des zu entwickelnden Teilsystems (z. B. Motorsteuergerät) zu testen. Die Eingänge des Steuergeräts werden mit Daten aus dem Modell stimuliert. Um die Test-/Reglerschleife (Loop) zu schließen, wird die Reaktion der Ausgänge des Steuergeräts, z. B. das Ansteuern eines Elektromotors, in das Modell zurückgeführt.

Die HIL-Simulation muss meist in Echtzeit ablaufen und wird in der Entwicklung und Test benutzt, um Entwicklungszeiten zu verkürzen und Kosten zu sparen. Insbesondere lassen sich wiederkehrende Abläufe simulieren und automatisieren (für Regressionstests).

Die Tests an realen (oft recht teuren) Systemen lassen sich dadurch stark verringern und zusätzlich lassen sich Systemgrenzen testen, ohne das Zielsystem (z. B. Auto und Fahrer) zu gefährden.

Embedded-System-Tests sind anders?

Tests im Rahmen von Embedded-Systemen sind in der Regel komplexer in der Durchführung wie Tests von reinen Software-Systemen.

Da die Test- und auch die Ziel-Systeme sehr oft teurer sind als bei reinen Software-Lösungen, muss dies bei der Architektur der Testautomatisierung auch entsprechend berücksichtigt werden (hier sind auch Themen wie HiL und SiL relevant).

Außerdem sind die Produktlebenszyklen von Embedded-Systeme meist auch langlebiger und in der Entwicklung stabiler als z.B. Web-SW-Lösungen. Damit rechnen sich in diesem Bereich auch aufwändige Testautomatisierungslösungen deutlich schneller.

Im Bereich der Testspezifikation und des Testprozesses sind die anzuwendenden Methoden und Vorgehensweisen in den meisten Fällen jedoch sehr ähnlich, wobei auf diese Themen im Rahmen des Newsletters nicht eingegangen wird.

Dieser Newsletter gibt einen grundlegenden Einblick in das Testen von Embedded-Systemen und stellt konkret anhand eines Projekt-Beispiels für Embedded-Systeme sowie eines kurzen Exkurses über NUnit dar, wie ein Testframework umgesetzt werden kann.

Dipl.-Ing. Johannes Bergsmann
Staatl. befugter und beeideter Ingenieurkonsulent für Informatik

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.at .

Weitere Infos zu diesem und anderen Themen finden Sie auf http://www.software-quality-lab.at .

software quality lab
improve your quality

Quality Knowledge Letter

www.software-quality-lab.at

Black-Box-Testautomatisierung für eingebettete Systeme

von Bernhard Groiß, MSc, Software Test Consultant bei Software Quality Lab

Eingebettete Systeme wie Mobiltelefone, Geräte der Medizintechnik oder Unterhaltungselektronik spielen eine entscheidende Rollen in unserem Leben und prägen den Alltag. Es handelt sich hierbei meist um sehr spezifische Produkte, die sowohl von der Hardwarearchitektur als auch vom Softwaredesign speziell an ihre Aufgaben angepasst werden.

Aufgrund finanzieller Vorgaben werden oft kostengünstige Hardware-Software Implementierungen gewählt, welche die Leistungsfähigkeit von Hardware mit der großen Flexibilität aber auch Fehleranfälligkeit von Software vereinen.

Dieser Newsletter gibt einen generellen Einblick in das Testen von eingebetteter Software und zeigt anhand eines Beispiels, wie Tests für solche Systeme automatisiert werden können.

Embedded Systems Testing — Eine Einführung

Beim Testen von Software für eingebettete Systeme müssen im Vorfeld unterschiedliche Aspekte betrachtet werden.

Eine Schwierigkeit liegt darin, dass sich schon bei der Softwareentwicklung für diese Systeme gravierende Unterschiede zur klassischen PC-Softwareentwicklung (als synonym für die nicht embedded SW-Entwicklung) zeigen.

Bei der Applikationssoftwareentwicklung am PC wird meist auf Standardhardware mit standardisierten Schnittstellen und gängigen Betriebssystemen entwickelt.

Bei der Entwicklung von Software für eingebettete Systeme kommen oft spezielle Hardwarearchitekturen zum Einsatz, die für konkrete Aufgaben (z.B. Fahrzeugindustrie, Automatisierungstechnik, Funkübertragung) konzipiert wurden. Die Entwicklung der Software passiert hierbei oft auf der sogenannten Hostplattform, da auf dem Zielsystem (Target) meist nicht direkt entwickelt werden kann.

Nach Fertigstellung der Software kann diese dann auf die entsprechende Zielplattform portiert werden. Zusätzlich erfolgt die Entwicklung meist in Assembler, C oder anderen von der jeweiligen Hardware unterstützen Programmiersprachen und beinhaltet meist viele hardwarenahen Programmcode.

Weiters übernehmen eingebettete Systeme oft hochkritische Aufgaben, wie zum Beispiel in der Aeronautik oder der Medizintechnik, wodurch sich einer immer größer werdenden Anzahl an Standards entsprechen müssen.

Diese Eigenschaften und Anforderungen ändern und erweitern daher die Voraussetzungen an die immer mehr an Bedeutung gewinnende Qualitätssicherung solcher Systeme.

Je komplexer die Aufgaben der eingebetteten Software sind, desto schwieriger und zeitaufwendiger wird die Implementierung des Codes. Aufgrund der Komplexität und steigenden Anzahl der *Lines of Code* steigt auch die Fehleranfälligkeit der Software, was für sicherheitskritische Systeme oft auch unmittelbare Auswirkungen auf den Menschen und dessen Gesundheit haben kann.

Aus den genannten Gründen erhöht sich für solche Systeme demnach auch der Testaufwand.

Abbildung 1. zeigt schematisch die Kosten, um einen Fehler in der Software über die Projektlaufzeitzu beheben.

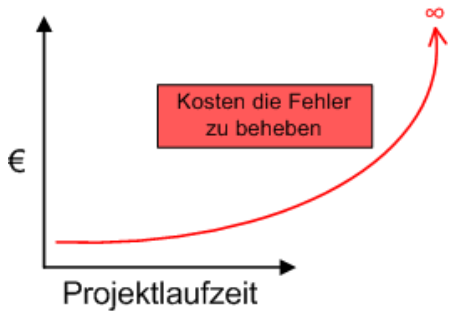


Abb. 1 — Kosten einen Fehler zu beheben als Funktion der Zeit im Produktlebenszyklus , Quelle: Ganssle

Um einen hohen Testabdeckungsgrad über die einzelnen Softwarekomponenten zu bekommen, ist eine systematische Qualitätssicherung während des gesamten Entwicklungsprozesses wichtig.

Die beginnt mit Unit Tests, die meist durch die Entwickler selbst erstellt und durchgeführt werden, über Systemtests bis hin zu Abnahmetests mit den Kunden. Gerade im embedded Systems-Umfeld ist oft die Qualität eines Produkts der ausschlaggebende Parameter für den Kauf, da Fehler in den embedded-Systemen oft auch mit teuren Rückholaktionen verbunden sein können.

Wie auch beim Test von PC-basierter Applikationssoftware wird im Bereich Embedded Systems Testing zwischen den folgenden Verfahren unterschieden:

(Fortsetzung auf Seite 3)

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.

Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter
http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Lettern von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.