Tool-Qualifizierung für ein Testautomatisierungswerkzeug

Ein Praxisbericht

Kristian Trenkel, iSyst Intelligente Systeme GmbH

Die ISO 26262 enthält Vorgaben für die Qualifizierung von Software-Tools, welche für die Entwicklung und den Test sicherheitskritischer Systeme einzusetzen sind. In der Praxis jedoch gehen die Auslegungen der verschiedenen Vorgaben weit auseinander. Dieser Betrag beschreibt die praktischen Erfahrungen mit der Tool-Qualifizierung des Testautomatisierungswerkzeuges iTestStudio bei einem Tier1. Weiterhin wird ein mögliches Vorgehen zur Tool-Qualifizierung dargestellt. Am Ende bleibt aber die Frage, wie der Inhalt der Norm zu interpretieren ist.

In der Automobilindustrie kommen immer mehr sicherheitskritische Funktionen wie zum Beispiel Notbremsassistent, Spurhalteassistent und autonomes Fahren zum Einsatz im Fahrzeug. Zur Entwicklung und zum Test dieser Systeme sind die Vorgaben der ISO 26262 [1] zu beachten. Dabei stellt diese Norm nicht nur Anforderungen an den Entwicklungsprozess und dessen Tools sondern auch an den Test einschließlich der Testmethoden und Test-Werkzeuge.

Es gibt für die Test-Tools, wie allgemein für Software-Tools, detaillierte Anforderungen, welche im Rahmen einer Tool-Qualifizierung zu bewerten sind.

Tool-Qualifizierung nach ISO 26262

Der Fokus der ISO 26262 in Bezug auf die Tool-Qualifizierung wird in Kapitel 11.1 [2] wie folgt beschreiben:

> *„The first objective of this clause is to provide criteria to determine the required level of confidence in a software tool when applicable.“*

Dazu wird eine Bewertung des Tool nach den Vorgaben der ISO 26262 in Kapitel 11 [2] durchgeführt. Dabei wird einerseits der Tool Impact (TI – TI1 oder TI2) bewertet, welcher das mögliche Einbringen oder Verbergen von Fehlern in sicherheits-kritischen Teilen beschreibt. Anderseits wird das Tool Error Detection (TD – TD1, TD2 oder TD3) untersucht, welches eine Aussage über die Zuverlässigkeit des Tools ausdrückt. Basierend auf dem TI- und dem TD-Wert wird der Tool Confidence Level (TCL – TCL1, TCL2 oder TCL3) bestimmt. Je nach TCL-Bewertung sind zusätzliche Maßnahmen bei der Entwicklung und dem Test des Werkzeugs auszuführen.

In der Praxis ist das vordergründige Ziel, eine Einstufung mit TCL1 zu erreichen, da dann keine weiteren Maßnahmen erforderlich sind. Dabei zeigt die Erfahrung, dass die Einstufung nicht immer objektiv erfolgt. Es steht oft die Aufwands- und damit Kostenminimierung in der Entwicklungsphase im Mittelpunkt. Beispielsweise werden selbst Compiler in der Praxis mit TCL1 eingestuft. Wobei es realistisch nur schwer möglich ist, Fehler, die durch einen Compiler generiert werden, prinzipiell auszuschließen.

Das Tool iTestStudio

Das Testautomatisierungswerkzeuges iTestStudio der iSyst Intelligente Systeme GmbH, welches in diesem Beitrag als Beispiel dienen soll, wird zur Erstellung und automatischen Ausführung von Tests unter Verwendung der Testsysteme, z.B. Hardware in the Loop (HiL)-Testsysteme eingesetzt. Dabei wird auf die verschiedenen Test-Tools, wie zum Beispiel CANape, CANalyzer und ControlDesk, über Automatisierungsschnittstellen (z.B. COM) zugegriffen.

Die Software iTestStudio besteht aus verschiedenen Komponenten, wie sie in Abbildung 1 dargestellt sind.

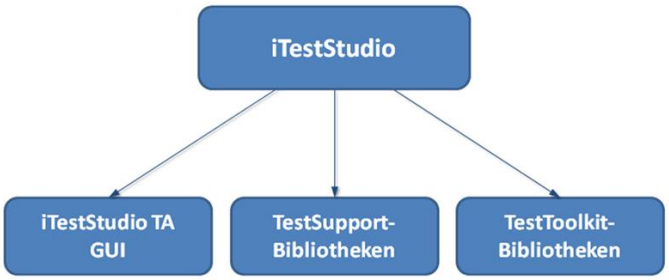

Abbildung 1: Komponentenstruktur des iTestStudio – Übersicht

Das Zusammenspiel der einzelnen Komponenten für die Testausführung ist in Abbildung 2 zu sehen.

Abbildung 2: Komponenten für Testimplementierung und Testausführung

iTestStudio TA

Die Komponente iTestStudio TA (siehe Abbildung 3), stellt die eigentliche Benutzeroberfläche dar. Diese umfasst die Organisation der Testfälle mit der Zusammenstellung von Testserien und Testsequenzen (siehe Abbildung 4), das Ausführungen/Starten der Testskripte und das erstellen der Testreports.

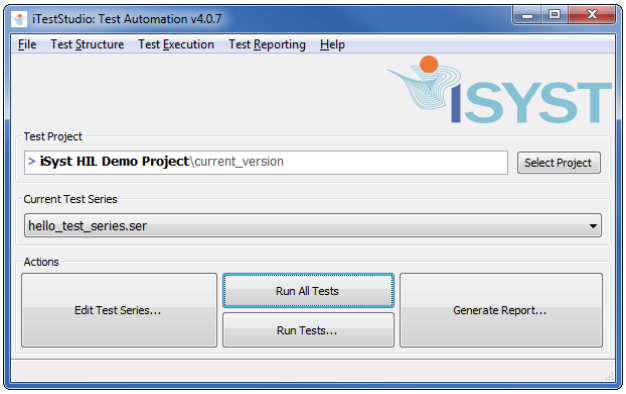

Abbildung 3: iTestStudio TA GUI

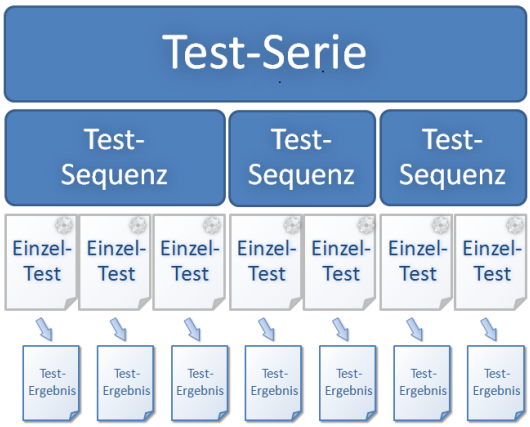

Abbildung 4: Testzusammenstellung und Generierung der Testergebnisse

TestSupport-Bibliotheken

Als zweites sind die TestSupport-Bibliotheken zu betrachten. Diese Bibliotheken kümmern sich um die Ausführung der Tests mit der zugehörigen Python-Version, welche nicht zwingend die Python-Version der iTestStudio TA GUI sein muss. Weiterhin sind Funktionen zum Speichern der Ergebnisse enthalten.

TestToolkit

Als drittes ist das TestToolkit zu untersuchen. Dieses enthält spezielle Testfunktionen, Schnittstellen zu Tools (wie Vector CANape oder dSPACE ControlDesk) sowie Container für Cal- und Hil-Variablen.
Das Zusammenwirken des Testautomatisierungswerkzeuges mit dem Testsystem und verschiedenen Test-Werkzeuge macht die Einschätzung für die Tool-Qualifizierung schnell sehr komplex und dabei schwierig sowie Kosten- und Zeit-intensiv.

Die Qualifizierung beim Tier1

Im Rahmen des Einsatzes des iTestStudio für den Test eines Fahrwerksteuergerätes mit Einstufung in ASIL B und bei einem Tier1 wurde die Qualifizierung nach ISO 26262 notwendig.

Im ersten Schnitt wurden Dokumente zum Einsatz des iTestStudio sowie eine „Selbsteinschätzung“ vom Verantwortlichen für die funktionale Sicherheit des Tier1 angefordert. In Anlehnung an die ISO 26262 ([2] Kapitel 11.4.5 und folgende)

wurde ein entsprechendes Dokument erstellt und übergeben. Dieses Dokument umfasst Angaben zu möglichen Fehlern des Testautomatisierungswerkzeuges, welche zu inkorrekten Bewertungen in den Testberichten führend konnten. Es dokumentiert weiterhin welche Maßnahmen ergriffen wurden bzw. bei dem Einsatz zu ergreifen sind, damit dieses Fehlverhalten nicht unentdeckt bleibt.

In der darauf folgenden Besprechungen wurde das Dokument (Umfang 12 Seiten) als zu umfangreich abgelehnt. Bei der weiteren Diskussion stellte sich heraus, dass der Tier1 ein Formular (Excel-Sheet) für die Erfassung der notwendigen Informationen für die Durchführung der Tool-Qualifizierung verwendet. Dieses Dokument sowie ein Beispiel wurden zur Verfügung gestellt. Bei näherer Betrachtung zeigte sich, dass lediglich eine einzige Zeile innerhalb des Dokumentes für die Eingabe von Daten vorgesehen ist. Dem entsprechend war es nicht möglich, umfangreichere Informationen in diesem Dokument zu generieren. Die sehr oberflächliche Betrachtung wurde aber durch den Tier1 akzeptiert und führte zu einem TCL1 für das Testautomatisierungswerkzeuges iTestStudio.

Zusammenfassend ist festzustellen, dass die Tool-Qualifizierung durch den Tier1 viel oberflächlicher ablief, als es die Vorgaben der ISO 26262 erwarten ließen. Alle Beteiligte, die vorrangig im Test tätig sind, erschien dieses Vorgehen als nicht sinnvoll. Im weiteren Vorgehen sollte geklärt werden, wie umfangreich eine Tool-Qualifizierung sein muss.

Eigenes Vorgehen zur Tool-Qualifizierung

Für den Aufbau von weiterem Know How für die funktionale Sicherheit nach ISO 26262 nahm der Autor an der Schulung und Prüfung zum „Functional Safety Professional“ des TÜV Süd teil. Im Rahmen der Schulung wurde auch das Thema der Tool-Qualifizierung sowie die Sicht des TÜV Süd bezüglich des notwendigen Umfanges besprochen.

Es ergaben sich dabei die folgenden zentralen Ergebnisse:

1) Die Tool-Qualifizierung ist, wie es die ISO 26262 auch gefordert, in der Einsatzumgebung, gegebenenfalls für jedes Projekt neu, durchzuführen.
2) Es sind neben dem Tool selbst, auch die Schnittstellen zu anderen Tools zu betrachten.
3) Alle Einschätzungen und getroffenen Maßnahmen sind schriftlich zu dokumentieren.

Im Ergebnisse daraus wurde für das iTestSudio ein Dokument erstellt, welches folgende Punkte umfasst:

- Aufbau des Tools
- Vorgesehener Einsatz und Einbindung des Tools
- Eingesetzter Entwicklungsprozess
- Durchgeführte Absicherungsmaßnahmen
  (z.B. Reviews, Softwaremodultests, ...)
- Betrachtung der Maßnahme nach ISO 26262 Kapitel 11.4.6 – 11.4.9 [2]

Für die Betrachtung der Maßnahme nach ISO 26262 Kapitel 11.4.6 – 11.4.9 [2] wurde angenommen, dass das iTestStudio nach TCL3 eingestuft wurde. Die notwendigen Maßnahmen sind weiterhin abhängig vom ASIL-Level des zu testenden

Systems. Mit der Annahme des TCL3 konnten nun möglich Maßnahmen aus den Kapitel 11.4.7 – 11.4.9 [2] für alle ASIL-Level betrachtet werden und Zuordnungen zu bereits durchgeführten Aktivitäten aufgezeigt werden. Die genannten Kapitel stellen dabei eine bereite Auswahl an Maßnahmen bereit, mit denen das Vertrauen in das Tool erhöht werden kann, und somit der ein Einsatz für die Entwicklung und den Test eines sicherheitskritischen Systems vertretbar ist.

Beispielhaft sei das Vorgehen am Punkt d) aus Kapitel 11.4.7.2 mit der Angabe „the occurrence of malfunctions and corresponding erroneous outputs of the software tool acquired during previous developments are accumulated in a systematic way.“ beschrieben. Ziel dieses Punktes ist es, das Vertrauen durch Nachverfolgbarkeit von Fehlern und ihrer Behebung zu stärken. Im Fall des iTestStudios kommen die Tools TRAC und Subversionen zum Einsatz. Dabei wird TRAC für die Erfassung aller aufgetretenen Fehler sowie für die Dokumentation der Behebung verwendet. Die behobenen Fehler, Änderungen und Neuerung werden zu jeder Version in Release-Note (changelog.txt) dokumentiert.

Im Ergebnis konnten zu den meisten in der ISO 26262 geforderten Maßnahmen auch entsprechende durchgeführte Aktivitäten zugeordnet werden. Einzig der Punkt 11.4.6 1d) „Development in accordance with a safety standard” konnte nicht vollständig dargestellt werden. Wobei die Fußnote “No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.” in der ISO 26262 [2] auf Seite 25 diesen Punkt nicht sehr aussagekräftig macht.

Zusammenfassend ist festzustellen, dass die ISO 26262 eine recht präzise Handlungsanweisung für die Tool-Qualifizierung bietet. Bei einer entsprechend gründlichen Durchführung kann eine nachvollziehbare und belastbare Einstufung eines Tools erstellt werden.

In der Praxis geht die Qualität dieser Betrachtungen aber leider weit auseinander. Hier wäre ein einheitliches Vorgehen wünschenswert, ja oftmals sogar notwendig, um kleineren Zulieferern den Aufwand für die Erstellung neuer Dokumente für jeden Kunden zu ersparen.

Zukünftige Planung

Für eine Vereinfachung der Tool-Qualifikation beim Kunden ist es geplant, ein Tool Qualifikation Kit für das iTestStudio zu erstellen. Dies Kit wird besten aus:

- Dokumentation des Tools
- Dokumentation der Einsatzumgebung
- Einschätzung zur Safety nach ISO 26262 (wie in Kapitel *Eigenes Vorgehen nur Tool-Qualifizierung* beschreiben)
- Liste kompatibler Test-Tools und Testsysteme
- Ausführbares Testprojekt zur Prüfung des korrekten Installation

Bei der Zusammenstellung der Umfänge des Tool Qualifikation Kits hat sich gezeigt, dass die meisten Teile bereits vorhanden waren. Es lassen sich interne Dokumente sowie die Beschreibung der Umgebung für die internen Tests des Tools mit kleinen Anpassungen einsetzen.

Zusammenfassung

Die Erfahrung zeigt, dass das Thema Tool-Qualifizierung nach ISO 26262 in verschiedenen Unternehmen sehr unterschiedlich betrachten wird. Nur in seltenen Fällen werden sehr tief gehende Betrachtungen durchgeführt. Meist geht es aber nur darum einen TCL1 mit möglichst minimalem Aufwand zu bewerten, wie dies in diesem Beitrag beschreiben ist.

Aus Sicht eines Testers ist dieses Vorgehen kurzsichtig bzw. sogar fahrlässig. Die Vorgaben der ISO 26262 sind sinnvoll und sollten berücksichtigt werden. Einerseits hat sich bei der eigenen Bewertung des Tools iTestStudio gezeigt, dass eine unvoreingenommene Betrachtung dazu führt, eigene Schwächen in der Entwicklung zu erkennen und zu einer Verbesserung der Qualität des eigenen Tools führt. Andererseits deckt ein typischer Entwicklungsprozess (z.B. nach dem V-Modell) bereits einen Großteil der Anforderungen der ISO 26262 ab.

Wenn von Beginn der Entwicklung an die Prozessvorgaben eingehalten werden, dann ist für die Tool-Qualifizierung nach ISO 26262 kein signifikanter Mehraufwand notwendig.
Insgesamt wäre ein einheitliches Vorgehen der verschiedenen Firmen (vor allem des OEMs und Tier1s) im Bereich der Tool-Qualifizierung wünschenswert.

Quellenverzeichnis

[1] ISO, ISO 26262:2011, Genf: ISO, Road vehicles -- Functional safety.

[2] ISO, ISO 26262-8:2011, Genf: ISO, Road vehicles -- Functional safety -- Part 8: Supporting processes.

Autor

Kristian Trenkel studierte ab 2001 an der FH Jena Elektrotechnik / Technische Informatik. Danach folgte eine Tätigkeit als Entwicklungsingenieur. Im Rahmen der Zusammenarbeit der TU Chemnitz mit der iSyst GmbH konnte seine kooperative Promotion durchgeführt werden. Seit 2013 betreut Herr Trenkel die Forschungs- und Förderprojekte sowie die studentischen Arbeiten bei der iSyst GmbH und als „Functional Safety Professional“ Problemstellung der funktionalen Sicherheit im Bereich der Testsysteme und Testautomatisierungs-Software.

Kontakt

Internet: www.isyst.de
Email: Kristian.Trenkel@isyst.de
Telefon: +49(0)911/37665–114