Safe Continuous Integration

Continuous Integration in sicherheitsgerichteter Entwicklung

Gudrun Neumann, SGS-TÜV Saar GmbH

Ein manueller Software Integration-Prozess kann nicht immer mit der Geschwindigkeit der Änderungen in der Softwareentwicklung mithalten. Deshalb gehen mehr und mehr Hersteller dazu über einen Continuous Integration (CI) Prozess durchzuführen. D.h. Software Build Zyklen zur Software-Integration werden zusammen mit automatisierten Tests früh und häufig durchgeführt.
In diesem Beitrag wird anhand eines beispielhaften CI Prozesses dargestellt, welche Anforderungen der Standards zur Funktionalen Sicherheit beachtet werden müssen.
Dabei wird ein besonderes Augenmerk auf Versionsmanagement, Traceability und Automatisierung von Tests liegen.
Auch bei dieser Vorgehensweise müssen die Nachweise für die sicherheits-relevanten Tätigkeiten erbracht werden, wie z.B. Auswahl der Testmethoden und Testspezifikationen.
Eine Zusammenfassung der wesentlichen Aspekte erfolgt am Ende des Beitrags.

Einleitung

In der Industrie werden ganz verschiedene Anforderungen an einen Software Integrations-Prozess gestellt. Dazu gehören, besonders bei sicherheitsgerichteter Entwicklung, die Wiederholbarkeit der Verifikationsschritte und ein frühes sichtbares Ergebnis, d.h. ein zumindest in Teilen funktionierendes Software System. Dies kann durch die im Folgenden beschriebenen Continuous Integration der Software erreicht werden.

Definitionen

Softwareintegrationstests dienen

- zur Verifizierung, dass die Anforderungen an die sicherheitsbezogene Software erreicht wurden und
- zum Nachweis, dass alle Softwaremodule, -elemente und Teilsysteme ordnungsgemäß zusammenarbeiten und ihre bestimmungsgemäßen Funktionen und keine anderen ausführen

(siehe auch IEC 61508, Teil 3).
Man unterscheidet zwischen reiner Softwareintegration und der Integration von Software auf der Zielhardware.
Im Folgenden wird nur die reine Softwareintegration betrachtet.

Es gibt verschiedene Strategien zur Integration von Software Komponenten zu einem Softwaresystem:

- „Big Bang“, d.h. alle Software-Komponenten werden in einem Schritt integriert.
- „Schrittweise Integration“, d.h. sinnvoll zusammengehörende Software Komponenten werden integriert und in einem weiteren Schritt die so entstandenen Software-Komponenten.

- „Continuous Integration (CI)“, d.h. jede durch den Entwickler freigegebene Software Änderung wird möglichst sofort mit schon bestehendem Code integriert und getestet.

Voraussetzungen für eine CI sind:

- Versions- bzw. Konfigurationsmanagement
- Automatisiertes Software Build
- Automatisiertes Testen

Beispiel CI Prozess

Im Folgenden wird ein Beispiel CI Prozess vorgestellt:

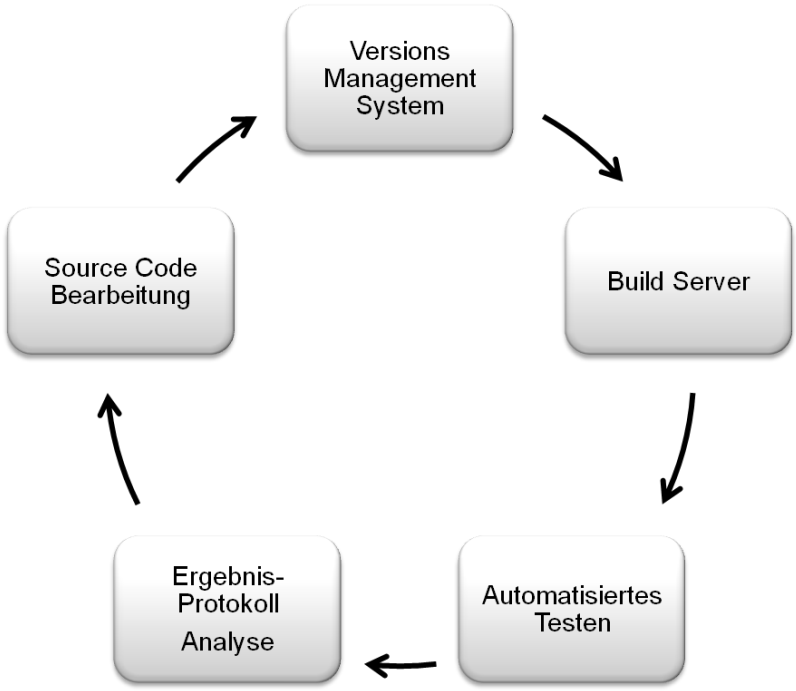

Abbildung 1: Beispiel CI-Prozess

Folgende Schritte werden betrachtet:

1. Der Entwickler gibt seinen erzeugten und lokal getesteten Source Code zum Test frei, d.h. er speichert seine Arbeitsversion als gültige Version im Versionsmanagementsystem (z.B. SVN Tool).
2. Zu bestimmten Zeitpunkten, z.B. „über Nacht“, wird automatisch aus allen gültigen Versionen auf dem Build Server eine neue Software System Version erzeugt.
3. Die neue Software System Version wird automatisierten Tests unterzogen. Dies können sehr viele (>10.000) Testfälle sein. Die Ergebnisse der Tests werden mit den vorher spezifizierten erwarteten Testergebnissen verglichen. Die Ergebnisse dieses Vergleichs werden zur Verfügung gestellt.
4. Diese Testergebnisse werden am nächsten Morgen analysiert und im Rahmen eines vorgegebenen Prozesses gegebenenfalls Korrekturen durchgeführt.

Schritt 2. Und 3. gibt es auch in einer verkürzten Variante, dabei wird der Build Prozess und ein eingeschränkter Test durchgeführt, um zu prüfen ob der „neue“ Source Code übersetzbar und prinzipiell kompatibel mit der schon existierenden Software ist. Dieses verkürzte Verfahren findet in Kombination mit einem vollen Testlauf Verwendung.

Anforderungen und deren Umsetzung im Beispiel CI Prozess

Eine entscheidende Bedeutung in einem solchen Prozess haben zuverlässige Software-Tools. Alle Standards zur Funktionalen Sicherheit fordern die Dokumentation, Klassifikation und, wenn notwendig, Qualifikation der verwendeten Tools. Im Falle eines CI Prozesses sind die Tools für das Versionsmanagement, z.B. SVN, die Build-Skripte, z.B. Perl-Skripte, die automatisierte Testumgebung und die Entwicklungsumgebungen zu betrachten. Besonderes Augenmerk liegt auf der reibungslosen Zusammenarbeit der verschiedenen Tools als Toolchain für den CI Prozess.

Beim Erzeugen der Testfälle für den automatisierten Test ist zu beachten, dass nachvollziehbar ist welche Anforderungen durch diesen Testfall verifiziert werden. Dies sollte auch im Ergebnisprotokoll zu erkennen sein, um die Nachvollziehbarkeit zu erleichtern. Bei Problemen, d.h. Testergebnis „Fail“, kann so auf einfache Weise der Bezug zu den relevanten Anforderungen hergestellt werden.
Minimal sollten die Testprotokolle besonderer Testläufe, wie z.B. letzter Test vor der Freigabe der Software, archiviert werden, auch wenn einzelne Testläufe im CI Prozess jederzeit nachvollziehbar sind. Diese Archivierung wird von einigen Standards zur Funktionalen Sicherheit gefordert, z.B. IEC 60880. Bei CI werden üblicherweise im Laufe der Verifikationsphasen weitere Testfälle ergänzt, was zu einer besseren Testabdeckung führen kann.

Die meisten Standards zur Funktionalen Sicherheit fordern zusätzlich die Dokumentation der verwendeten Testmethodik, wie z.B. anforderungsbasiertes Testen. Dies kann in der Dokumentation des Erstellungsprozesses der Testfälle berücksichtigt werden, die ein Teil der Dokumentation des CI Prozesses bildet.

Zusammenfassung

Die verwendeten Software-Tools müssen entsprechend den Anforderungen des anzuwendenden Standards zur Funktionalen Sicherheit dokumentiert, klassifiziert und gegebenenfalls qualifiziert werden. Dabei sind die Schnittstellen zwischen den einzelnen Software-Tools besonders aufmerksam zu betrachten.
Die Nachvollziehbarkeit der Implementierung der Anforderungen im Source Code zu Testfall und Testergebnis, sollte im CI Prozess fest verankert werden, um die Erfüllung dieser Anforderung für die Entwickler zu erleichtern. In der Dokumentation des CI Prozesses sollten auch weitere Anforderungen der sicherheitsgerichteten Softwareentwicklung berücksichtigt werden.
Eine Continuous Integration von Software kann sich so gut in eine sicherheitsgerichtete Software-Entwicklung einfügen.

Abkürzungsverzeichnis

CI	Continuous Integration

Literatur

IEC 61508:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems
DIN EN 61508:2011: Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme
IEC 60880:2006: Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions

Autorin

Gudrun Neumann studierte Informatik an der Technischen Universität München und schloss das Studium als Diplom-Informatikerin Univ. (Dipl.-Inform. Univ.) im Jahr 1990 ab.
Im Anschluss war sie bei der Siemens AG in verschiedenen Bereichen tätig.
Seit 2010 ist sie als Product Manager Functional Safety Software bei der SGS-TÜV Saar tätig und ist dort verantwortlich für die Durchführung von Analysen und Beurteilungen von komplexen Systemen, z.B. Qualifizierung von Software Tools. 2012 wurde sie Team Leiterin des Software Teams. Neben diesen Tätigkeiten führt Sie auch Beratungen und Schulungen zur Funktionalen Sicherheit von Software durch.

Kontakt

Internet: www.sgs-tuev-saar.com
Email: gudrun.neumann@sgs.com