

# DIGITAL ENGINEERING MAGAZIN

LÖSUNGEN FÜR KONSTRUKTEURE, ENTWICKLER UND INGENIEURE + INDUSTRIE 4.0 | INTERNET DER DINGE

TURBO FÜR DIE SOFTWARE-ENTWICKLUNG

## Mit Continuous Integration Engineering-Kosten reduzieren

SIEMENS



› **EXPERTENUMFRAGE**

Status quo in Sachen KI und Industrie 4.0

› **EFFIZIENZGEWINN**

Enge Verzahnung von CPQ und SAP

Mit Special  
**AUTOMATISIERUNG**

# Qualität rauf, KOSTEN RUNTER

Die Verschmelzung von IT und OT ist ein viel zitiertes Ziel der fortschreitenden Digitalisierung. Ein konkretes Anwendungsbeispiel ist Continuous Integration: Dieser Prozess zur Steigerung der Softwarequalität hilft, die Engineering-Kosten in der Automatisierung zu reduzieren. > von Stefan Kläber

Das Engineering zählt im Maschinenbau nicht nur zu den wesentlichen Bestandteilen, sondern auch zu den größten Zeitfressern. Während die Hardware-Kosten eher stagnieren, wachsen die Aufwände – und somit die Kosten – für das Engineering aufgrund immer leistungsfähigerer und komplexerer Maschinenkonzepte. Moderne Engineering-Tools wie das Totally Integrated Automation Portal (TIA Portal) von Siemens unterstützen die Programmierung von speicherprogrammierbaren Steuerungen (SPS) durch zahlreiche Werkzeuge und Komfort-Funktionen. Doch das ist nicht alles. Im IT-Bereich gibt es viele Methoden und Prozesse, um die Software-Entwicklung zu optimieren und zugleich die Qualität der Ergebnisse zu verbessern. Ein solcher Prozess ist Continuous Integration (kontinuierliche Integration), der sich auch auf das Engineering in Automatisierungsprojekten anwenden lässt und zahlreiche Vorteile bietet.

## Turbo für die Software-Entwicklung

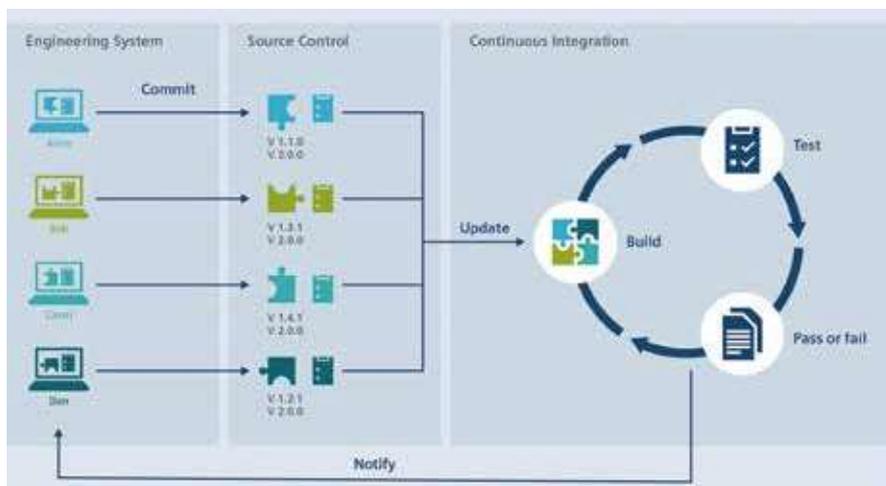
Aber was genau ist Continuous Integration? Dieser Prozess ermöglicht Entwicklern,

ihre Änderungen so oft wie notwendig in die gemeinsame Quellcode-Basis zu integrieren sowie diese automatisiert zu generieren und zu testen. Dabei wird das Ziel verfolgt, Integrationsprobleme frühzeitig zu erkennen. In der Praxis bedeutet dies, dass traditionelle Vorgehensweisen bei der Software-Entwicklung durch agiles interdisziplinäres Arbeiten im Team abgelöst werden. Mehrere Programmierer arbeiten zusammen als Team an einem TIA-Portal-Projekt. Jeder Entwickler bearbeitet die ihm zugeordneten Software-Module anhand eines standardisierten Programmierleitfadens und checkt die Ergebnisse in einer Versionsverwaltung ein. Ein Continuous-Integration-Server kann automatisiert aus den einzelnen Software-Modulen einen Build erstellen und die Funktionen der Software automatisiert testen. Das Ergebnis der Tests wird als Bericht an die Programmierer zurückgespielt, damit diese – wenn nötig – Korrekturen vornehmen können. Der kontinuierliche Kreislauf aus Build, Test und Feedback verbessert die Software-Qualität und spart Entwicklungszeit. Gleichzeitig wird die Fehlersuche und -behebung anhand über-

schaubarer, logisch-geschlossener Programmsegmente vereinfacht und (teil-)automatisiert.

## Standardisierung beseitigt Insellösungen

Damit sich Continuous Integration in der Automatisierung einsetzen lässt, sind bestimmte Voraussetzungen notwendig. Stichwort Standardisierung: Im Maschinenbau ist es heute noch häufig gang und gäbe, dass die Programmierung der Steuerungen unabhängig von anderen Arbeitsschritten erfolgt und die Entwickler am Ende des Entwicklungsprozesses von Maschinen stehen. Dies hat zur Folge, dass der SPS-Code oft erst spät in der Entwicklungsphase einer Maschine geprüft wird beziehungsweise geprüft werden kann. Dabei gilt: Je später Fehler entdeckt werden, umso aufwändiger ist deren Behebung. Wenn mehrere Entwickler an einem Projekt arbeiten, fehlt oft die Transparenz bei Änderungen, da es an den nötigen Werkzeugen mangelt. Und jeder Entwickler pflegt gerne seinen eigenen Stil, was Namens-



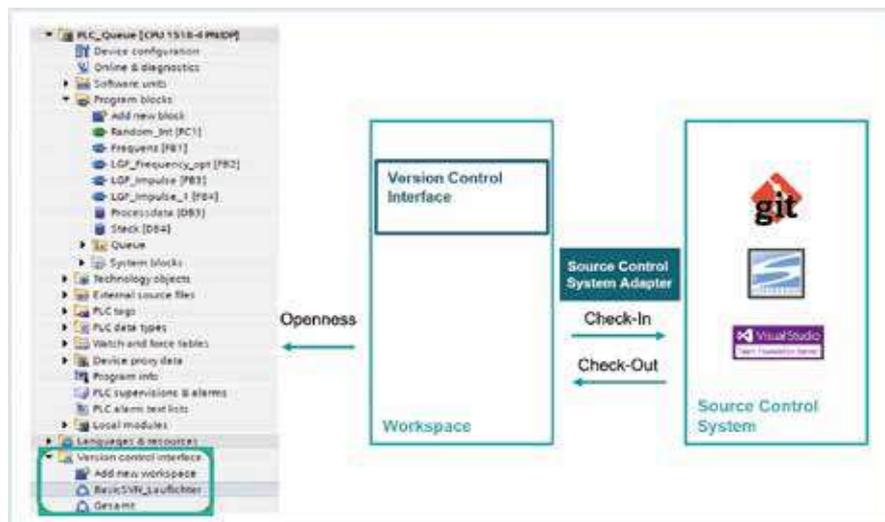
Grundprinzip der Software-Entwicklung mit Continuous Integration.



gebung, Struktur und Ähnliches betrifft. Im ersten Schritt muss daher die Entwicklung standardisiert werden, damit man von einer einheitlichen Datenbasis, nahtlos miteinander verbundenen Workflows und durchgängigen Schnittstellen profitiert. Damit können alle Disziplinen effizient, zeitgleich und sicher an einem gemeinsamen Automatisierungsprojekt arbeiten.

## Evolution der Software-Entwicklung

Im Automatisierungs-Engineering fehlen oft die Voraussetzungen für testgetriebene Entwicklungen, kontinuierliche Software-Anpassungen (Refactoring) oder schnelle Code-Reviews. In der IT sind Methoden der agilen Software-Entwicklung dagegen längst verbreitet und werden gerne eingesetzt, wenn Transparenz und Flexibilität wichtig sind, um die Software bei hohen Qualitätsstandards dennoch rasch einsetzen zu können. Dabei sind Konzepte der agilen Entwicklung, wie sie in der IT zur Anwendung kommen, auch in der Automatisierung möglich: Dank der Anbindung an leistungsfähige Tools für eine Versionskontrolle und der Standardisierung von Software-Modulen lässt sich die Versionierung des Quellcodes sinnvoll und effizient einsetzen. Eine Versionsverwaltung dient zur Erfassung von Änderungen an Software-Objekten und Dokumenten entlang des gesamten Maschinenlebenszyklus. Alle Versionen werden mit Zeitstempel und Benutzerkennung versehen. Der Anwender kann jede Version problemlos wiederherstellen. Zu den wichtigsten Vorteilen der Versionsverwaltung für Automatisierer gehören die



Anbindung von Versionierungssystemen an das TIA Portal mit dem Version Control Interface (kurz VCI).

Bilder: Siemens

Koordinierung des gemeinsamen Zugriffs von mehreren Entwicklern auf die Objekte, die Transparenz über Änderungen und die gleichzeitige Bearbeitung mit mehreren Entwicklungszweigen („Branches“) eines Projektes. Bekannte Beispiele für Versionsverwaltungs-Software sind Git, Subversion (SVN) und Tfcv (TFS).

## Testen und berichten: Ein Job für den Job

Der Quellcode auf dem Versionsverwaltungssystem lässt sich nun für die Erstellung eines testfähigen Builds – ein TIA-Portal-Projekt mit notwendigen Software-Elementen – nutzen. Ziel ist es, das Generieren von Testprojekten nicht von Hand auszuführen, sondern automatisiert abzuwickeln, indem der Programmierer einen entsprechenden Testfall auf dem Continuous-Integration-Server erstellt. Dieser Job startet den Build auf dem Automatisierungs-PC und soll auch den automatisierten Funktionstest und die Prüfung von Programmierregeln starten. Anschließend wird der Job die Logs und Testergebnisse einsammeln sowie die Exit-Codes analysieren und das Ergebnis dem Programmierer zur Verfügung stellen. Die für die Detaildiagnose notwendigen Informationen wie Logs und Daten sind Bestandteil des Berichts.

## TIA Portal als Schaltstelle für Continuous Integration

Das TIA Portal als Engineering-Framework enthält ein umfangreiches Bibliothekskonzept zur Unterstützung der Standardisierung und ermöglicht die Parallelisierung der Workflows. TIA-Portal-Funktionen wie Multiuser Engineering, Test Suite und Version Control Interface liefern Antwort

ten auf die genannten Anforderungen zur Realisierung von Continuous Integration. Doch Technologie allein genügt nicht. Das Unternehmen muss die Bereitschaft zur Veränderung mitbringen und die unterschiedlichen Disziplinen zusammenführen, so dass Mechanik, Elektrik und Automatisierung gemeinsam und zeitgleich an einem Projekt arbeiten können. Die IT-Abteilung des Unternehmens muss ebenfalls eingebunden werden und die nötigen Server und Tools bereitstellen.

## In kleinen Schritten zur digitalen Transformation

Die Vorteile von Continuous Integration liegen auf der Hand: Der Kreislauf aus raschen Änderungen und automatisierten Funktionstests mindert das Fehlerrisiko, steigert die Software-Qualität und schafft einen transparenten Entwicklungsprozess. Zudem sorgen automatisierte Abläufe für Konsistenz und reduzieren den Zeitaufwand für das Engineering. Continuous Integration ist ein Schritt von vielen auf der Reise durch die digitale Transformation. Mit Standardisierung als Basis der Digitalisierung können schon heute Automatisierungs-Teams diverse Vorteile nutzen und ihre Erfahrungen schrittweise ins ganze Unternehmen übernehmen. Die Standardisierung ist damit der ideale Einstiegspunkt, um schrittweise einen vollständig kontinuierlichen Entwicklungsprozess umzusetzen. Darauf aufbauend lässt sich Continuous Integration Schritt für Schritt umsetzen und gleichermaßen für kleine wie auch sehr umfangreiche Projekte anwenden.

RT ◀

**Stefan Kläber** arbeitet im Marketing S7 & TIA Portal bei der Siemens AG.