SIEMENS
Ingenuity for life

Reducing engineering costs with Continuous Integration

Turbocharging software development

Quality up, COSTS DOWN
The merging of IT and OT is a frequently mentioned goal of progressive digitalization. One actual example of its application is Continuous Integration: This process for improving the quality of software helps to reduce the engineering costs of automation.
By Stefan Kläber

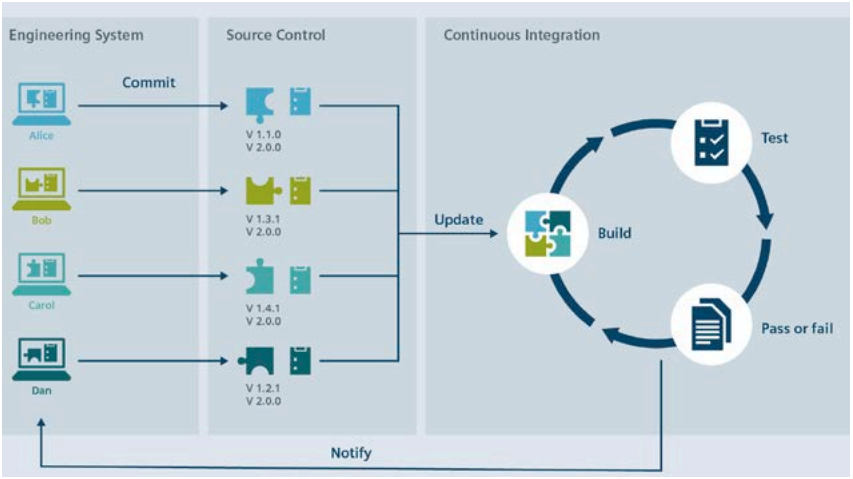
Engineering is not only a major part of electrical engineering, but also one of its greatest time-consuming activities. Whereas hardware costs are tending to stagnate, the time and money required for engineering increase on account of the demand to design ever more productive and complex machines. Modern engineering tools such as the Totally Integrated Automation Portal (TIA Portal) from Siemens provide a multitude of tools and user-friendly functions to support the the programming of programmable logic controllers (PLC). But that is not all. There are also many methods and processes in the IT fields for not only optimizing software development but also improving the quality of the results. One such process is Continuous Integration, which offers many advantages and can also be applied to the engineering in automation projects.

Turbocharging the software development

But what is Continuous Integration actually? It is a process that enables developers to integrate their modifications as often as needed into the basic common source code, and generate and test them automatically. The aim is to detect integration problems at an early stage. In practice, this means that traditional methods of software development are being replaced by agile, interdisciplinary teamworking. Multiple programmers work together as a team on TIA Portal projects. Each development engineer works on the assigned software modules on the basis of a standardized programming guideline and integrates the results in a version control system. A Continuous Integration server can automatically create a software build from the individual software modules and automatically test the functions of the software. The results of the test are conveyed as a report to the programmers so that they can make any necessary corrections. The continuous cycle of build, test and feedback improves the quality of the software and saves development time. At the same time, the search for and correction of errors are simplified and (partially) automated on the basis of clear, logically coherent program segments.

Standardization eliminates isolated solutions

Certain requirements have to be met before Continuous Integration can be used in automation. Keyword standardization: Nowadays, the control system is still frequently programmed separately from other working steps in machine manufacturing, and the development engineers are placed at the end of the process of developing machines. Consequently, the PLC code is often not checked or cannot be checked until late



development phases of a machine. However, the later an error is detected, the more it costs to correct it. When several development engineers are working on a project, there is often a lack of transparency in the changes because there is a lack of necessary tools. And each development engineer likes to follow his own style in respect of naming convention, structure and the like.

The evolution of software development

Automation engineering often lacks the prerequisites for test-driven developments, continuous software adaptations (refactoring), and quick code reviews. On the other hand, in IT, methods of agile software development have long been widespread and are used preferably if transparency and flexibility are needed to use the software quickly while maintaining high quality standards. Agile development concepts, as used in IT, can also be applied to automation: The use of powerful tools for version control and the standardization of software modules allows the versioning of the source codes to be used advantageously and efficiently. Version control is used to record changes to software objects and documents throughout the entire machine life cycle. All versions have a time stamp and user ID. The user can easily restore every version. The most important benefits of version control for automation engineers include the coordination of shared access for multiple developers to the objects, the transparency of changes, and the simultaneous working with multiple branches of the project development. Familiar examples of version control software are Git, Subversion (SVN) and Tfvc (TFS).

Testing and reporting: a job for the job

The source code in the version control system can now be used to create a testable build – a TIA Portal project with the essential software elements. The aim is to avoid generating test projects manually, but for the programmer to create a corresponding test case on the Continuous Integration server to automate their execution. This job starts the build on the automation PC and should also start the automated function test and check of programming rules. The job will then collect the logs and test results, analyze the exit codes, and make the results available to the programmer. The report contains the information required for detailed diagnostics, such as logs and data.

TIA Portal as the center for Continuous Integration

As an engineering framework, TIA Portal contains an comprehensive library concept for supporting standardization and allow work flows to take place simultaneously. TIA Portal functions – such as Multiuser Engineering, Test Suite and Version Control Interface – provide answers to the stated requirements for achieving Continuous Integration. But technology by itself is not enough. The company has to be prepared to make changes and bring together the separate disciplines so that mechanical, electrical and automation engineers can work together on a project simultaneously. The company's IT department must also be involved and provide the necessary servers and tools.

In small steps to the digital transformation

The benefits of Continuous Integration are obvious: The cycle of rapid changes and automated function tests reduces the risk of errors, improves the quality of the software, and creates a transparent development process. Automated sequences provide consistency and reduce the time required for engineering. Continuous Integration is one of many steps on the way to digital transformation. With standardization as the basis for digitalization, automation teams can already exploit a range of advantages and transfer their experience step by step into the entire company. This makes standardization the ideal starting point for progressively implementing a completely continuous development process. Building on this, Continuous Integration can be implemented step by step, and applied to both small and very extensive projects.
RT

Herausgeber
Siemens AG 2019

Digital Factory
Postfach 4848
90026 Nürnberg, Deutschland

Artikel-Nr.: DFFA-B10476-00