

55  
56  
57  
58  
59  
60  
61  
62  
63

## Siemens

# Siemens Streamlines Static Code Analysis with SonarQube

### company profile

---

**company**  
Siemens

---

**company size**  
Large Enterprise

---

**industry**  
Technology

---

### the challenge

Until 2015, Siemens Digital Factory was using multiple tools for software code quality analysis across its various business units. Project managers would use language-specific tools like FXCop (C#) and PC-Lint (C/C++) to identify and report issues with software code. Apart from these tools, they would also refer to other third-party tools to obtain information about technical debt, redundancies and potential errors.

This approach was not ideal for many reasons. Different tools produced different metrics, which made it difficult for managers to obtain a holistic view of code quality. Multiple tools also increased the learning curve for managers, requiring them to learn the details of each tool before becoming productive with them. Finally, each tool produced its own reports, resulting in multiple points of entry to any discussion of code quality.

### the solution

In 2015, the Siemens Digital Factory technical management team decided to harmonize the process and adopt a common tool for software code analysis across business units and development teams. Integration with existing Microsoft tools and technologies was identified as a key requirement, as were security and scalability.

The project team identified a number of criteria for the tool evaluation process:

- Support for multiple programming languages
- Support for various C++ flavors
- Microsoft Azure DevOps Server build integration
- IDE integration
- Support for unit test coverage results
- Security and scalability

76  
76  
77

## key results

---

Consistent, unified reporting

---

Smooth integration with Microsoft Azure DevOps suite

---

Large numbers of code analyzers

---

Powerful dashboards personalized to user role and context

---

Shorter learning curve due to intuitive, browser-based interface

---

Security and scalability

---

Based on these criteria, the project team identified various tools which were further evaluated through real-world usage with selected development projects. Emphasis was placed on tool usability and user metrics presentation and capabilities across programming languages. System security and unit test coverage were also verified during this prototyping phase. After considering the results, the team selected SonarQube as their platform of choice.

## the results

Today, SonarQube is tightly integrated with the Microsoft Azure DevOps Server build process for C/C++ and C# projects along with unit/integration test results from Visual Studio Test Platform (VSTest.Console) and NUnit.

Apart from this integration, Siemens SonarQube Service has developed a unified project configuration methodology for projects using special programming languages and custom build environments. These projects are now also analyzed with SonarQube, with results appearing in SonarQube's project dashboard.

Siemens works closely with SonarSource support and product development teams. This helps drive innovation by providing regular feedback on various product functionality such as the Sonar C & C++ analyzer and the Azure DevOps Server integration.

In the future, Siemens plans to re-implement its SonarQube server infrastructure using virtual machines or containers. This will help to better manage workloads and ensure SonarQube is easily accessible for development teams across the company.