SIEMENS
Ingenuity for life

Siemens Digital Industries Software

Creating highest-quality embedded software

Software quality specification, rich architecture specification and quality monitoring using Capital Software Designer and Sigrid

Executive summary

Siemens and Software Improvement Group (SIG) partner to provide best-in-class software and guidelines to control and improve the architecture and technical quality of embedded software for the automotive industry. Siemens Capital® Software Designer provides uniquely rich and front-loaded software architecture modeling and analyses, whereas Software Improvement Group brings 20 years of experience in measuring and enhancing software quality through automated monitoring and consulting services. This whitepaper outlines how the combined solutions enable you to create high-quality automotive software deliverables.

Jan Richter
Siemens Digital Industries Software

Michel van Dorp
Software Improvement Group

siemens.com/software

Automotive software engineering: A-SPICE and software quality challenges

Automotive industry market trends point toward connected, increasingly autonomous, highly customized, electric and networked vehicles that are perceived by the younger generation more as "tablets on wheels" than as classical vehicles. These vehicles are expected to be extensible over their lifecycles through app purchases and installations, and to offer added-value services that are based on networks to their passengers. In addition, time-to-market pressure on vehicle manufacturers is mounting, and product complexity frequently leads to defects detected in a late stage that are expensive to fix and diminish the company's profit.

In the past, automotive software was running subordinate, low-level embedded control and entertainment functions. Today, software perceives and categorizes its environment, coordinates the driving process in advanced driver assistance functions, provides telemetry data to its manufacturer, receives over-the-air updates, and obtains high levels of authority over route planning, engine, gear, brakes and steering. In other words, the software takes more and more responsibility for the driving process. The traditional clear-cut boundaries between infotainment software and vehicle operating software are blurring.

Consequently, a collection of completely new functions increases the volume of onboard software by orders of magnitude. A large portion of this software is safety-critical. Connecting vehicles to the internet opens it up to security threats compromising vehicle integrity and passenger confidentiality and safety. Due to constraints regarding weight, power, heat transfer and cabinet space, it is not possible to continue adding functions by adding more electronic control units (ECUs). Thus, the industry will see a consolidation of software functions on large, multi-core ECUs; dynamic updates and extensions require a departure from static ECU images as specified by the Automotive Open System Architecture (AUTOSAR) partnership toward a dynamic architecture more resembling general-purpose IT systems, as specified by the AUTOSAR Adaptive Platform.[1]

As a result, software quality is more important than ever, but also more challenging to achieve due to exploding code complexity. What is software quality? The international standard ISO/IEC 25010 offers a clear yet qualitative view on software quality and breaks it down into maintainability, reliability, security, usability, performance efficiency, portability, functional suitability and compatibility.

Non-functioning software has a significant impact on the customer experience of end customers. Due to the impact the software quality and security should have high priority during the product development process.

Siemens' Capital Software Designer and Software Improvement Group's Sigrid platform together cover the full scope of software quality according to ISO 25010.

Classical methods for ensuring software quality – code reviews, long QC cycles, qualitative quality notions – are misaligned with modern agile development methodologies and unable to deliver the required quality. It is necessary to define the needed quality levels for all kinds of software early in the software requirements analysis and design stage, to continuously measure and monitor the software quality achieved in reality, and to immediately feed back emerging problems into the design process. This approach benefits from these elements:

- Tight integration with rich embedded software architecture specification and analysis
- Early definition and specification of the software components' required quality levels
- Continuous quantitative software quality assessment and monitoring of metrics
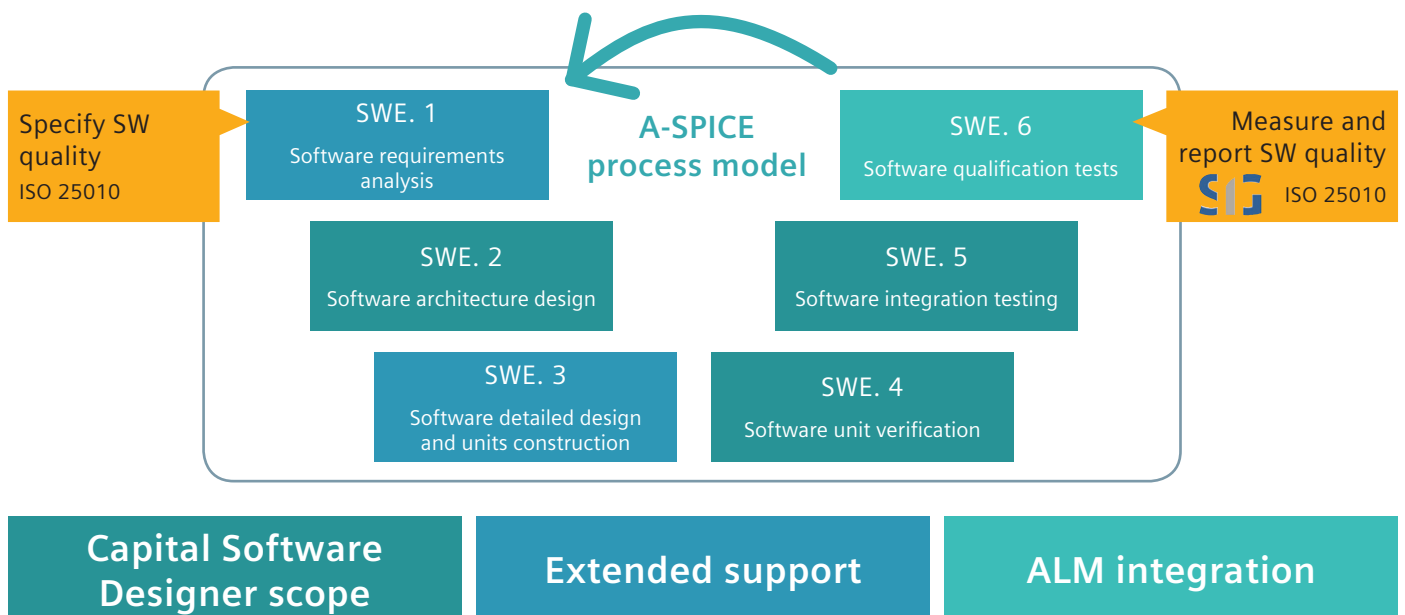- Feedback of software quality issues to inform architecture refactoring

Siemens' Capital Software Designer and Software Improvement Group's Sigrid® for Capital Software Designer together cover the entire picture. Capital Software Designer focuses on establishing a first-time correct software architecture and software component specification by capturing and analyzing software components, their interfaces, runnable functions, expected timing properties, software component internal

behavior, and variability. Capital Software Designer also addresses functional correctness both during pre- and post-implementation stages through test case design and execution of software unit tests, virtual simulation-based integration tests, and formal verification of implemented software.

Sigrid for Capital Software Designer integration continuously checks implemented software component code for relevant maintainability issues and suggests actions to improve software quality.

The integration between Capital Software Designer and Sigrid makes it easy to obtain an integrated quality view, and to turn insights from Sigrid into architectural enhancements.

From a process viewpoint, the Automotive Software Process Improvement and Capability Determination (A-SPICE) process model includes a software engineering process group that specifies a V-cycle for software development. Software quality management must be meshed into all process steps to be effective and efficient. Along with Polarion ALM™ software, Capital Software Designer addresses software quality in all process steps from software requirements capture to functional software verification. Software Improvement Group complements the process by measuring software quality after implementation, and by providing feedback to embedded software architecture design.

Getting the architecture right through rich specifications and analysis

Architecture specification and analysis using Capital Software Designer

Capital Software Designer supports your embedded software design flows starting from upstream engineering input in AUTOSAR or Systems Modeling Language (SysML) format, and enables you to capture legacy code and architecture assets.

The data flow diagrams are at the heart of embedded application software architecture. They capture precise interface specifications, adding physical unit systems to the data type system. Data dictionaries help to ensure data range and parameter consistency across the entire project. Software product lines reflect product variability on the embedded software level. Timing requirements capture the timing specifications of execution, and test cases capture the acceptance conditions as part of the embedded software specification. Behavior contracts expressed as pre- and post-conditions define acceptable boundaries for software component behaviors.
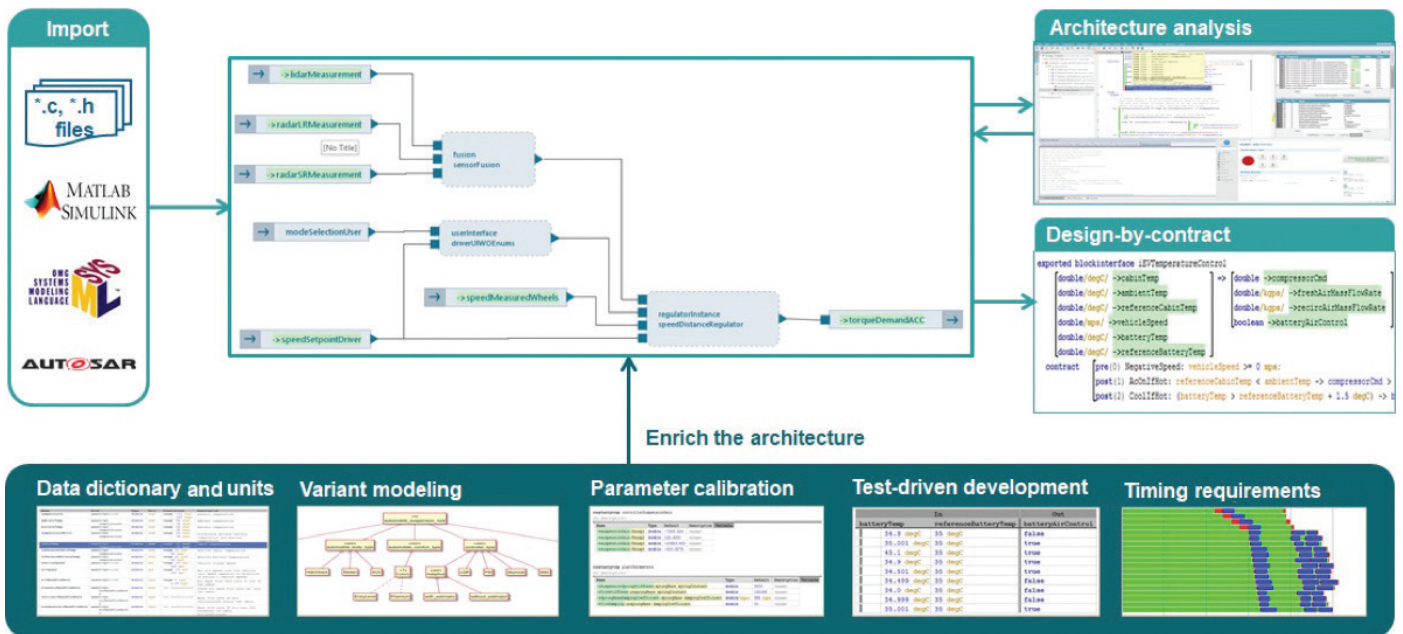
All of these model aspects become part of an embedded software specification, and they are analyzed for consistency before the first line of code is written.

The specification is completed by code shell templates generated from Capital Software Designer, also known as code frames. The software component specifications are then packaged and readied for shipping to suppliers, whether they are in-house development departments or external contractor companies.

Post-implementation code asset checking

Capital Software Designer supports three paradigms for verifying embedded software after implementation, focusing on correct integration and functional correctness:

Interface integration ensures that the different software components actually fit together and focuses on verifying the software interfaces.

Unit testing applies test conveniently, using tables to software components and groups of software components within a single control unit. Every unit test case consists of stimuli and expected outcomes, a single table line. Batches of software unit tests are grouped and batch-executable from test suites.

Simulation places the embedded software into the context of a virtual version of its physical environment, where Capital Software Designer integrates with Simcenter™ Amesim™ software and MathWorks' Simulink® as simulation engines. Capital Software Designer owns test harnesses from model- (MiL) and software-in-the-loop (SiL) paradigms, sets up the simulation, executes it, and collects and displays test results. An XiL harness consists of a stimulus model, a system under test, a plant model, and a monitor representing pass/fail criteria.

Formal verification helps you circumvent the limitations of finite test coverage. You apply the best-in-class C language model checking methods to source code, thereby detecting functional errors such as contract violations, as well as generic software defects such as array index errors. Furthermore, formal methods enable you to generate clever test cases that guarantee branch, location, condition and modified condition/decision coverage (MC/DC).
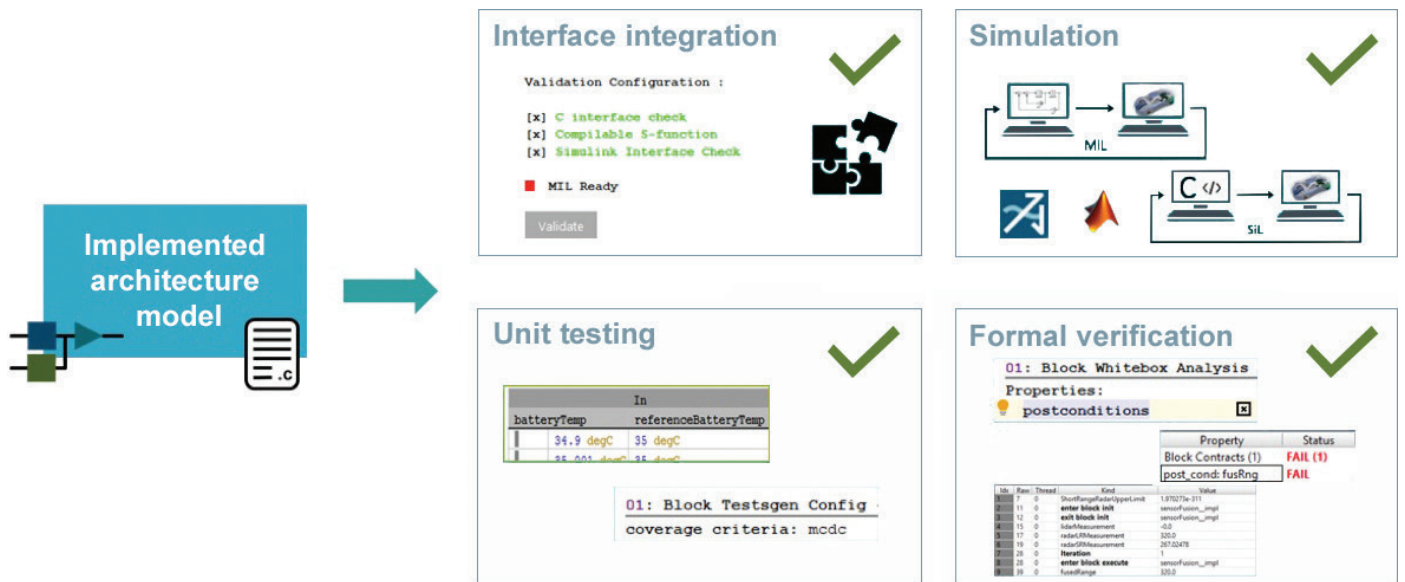
As described in the introduction, software quality has many dimensions: maintainability and its manifold aspects, reliability, security, usability, performance efficiency and portability.

Integrated software quality specification and analysis

The proper definition of software quality and adherence to this quality is a key factor in successful software development projects. Software Improvement Group's software assurance platform, Sigrid, has been developed to provide best-in class software quality analysis based on the international ISO/IEC 25010 standard for software quality. Sigrid analyzes source code according to the standard and compares it against Software Improvement Group's software quality benchmark. The result is immediate insight into the relative quality of the software. Sigrid subsequently provides guidance for improving the source code to achieve the expected quality.

With the Sigrid for Capital Software Developer integration, Software Improvement Group leverages the innate capabilities of Sigrid to bring software quality analysis to embedded software development. The integration allows architects and testers to gain insight in the developed quality and to track quality improvements until the desired level is achieved.

The scores in this model are based on a comparison with other systems that are in the extensive SIG software analysis database – which contains software quality measurements that have been collected for 20 years. SIG quality measurements not only score the systems on software quality, but also tell how the software is doing compared to other systems in the market, which is a competitive advantage. The IT industry is constantly evolving and improving, which is why SIG performs a recalibration of the model every year. The definition of quality thresholds is done based on actual industry

averages, giving it more relevance and neutral reference.

Focus on product functionality and product quality

Customers and suppliers often evaluate embedded software from a purely functional point of view. This means they focus on whether the software does what it is supposed to do. But this approach covers only one aspect of a successful project. How software is designed and built is of equal importance. In comparison, hardware components are inspected and certified before they are put into use. Why should this be different for embedded software? By measuring the technical quality of the embedded software, you can identify and mitigate risks early on. These risks include, but are not limited to, security vulnerabilities, project delays and long-term maintenance issues.

Measuring software quality starts with the ISO/IEC 25010 standard and its implementation. This international standard for software quality is applicable to embedded software as well. The standard defines the core aspects of software quality that influence the maintainability of software in the long term. The advantage of using an international standard is that any programming language can be evaluated against a well-defined and standardized framework. The standard also ensures that technical code quality is analyzed in an objective, repeatable way. Therefore, the ISO/IEC 25010 standard goes far beyond other standards that either focus on documents and processes instead of the technical quality, or do not assess architectural quality metrics.

After measuring the software code quality using the ISO/IEC 25010 standard with Sigrid for Capital Software Designer, it is time to improve the quality of the analyzed system. A fast, iterative process is provided to enable you to build high-quality software from the start. It provides very concrete recommendations as to where to improve code, and only in areas that really require attention. This means no need for looking at long lists of unprioritized violations.

The Sigrid for Capital Software Designer integration will help developers and architects to achieve high code quality by:

Reducing code complexity

When you reduce the complexity of the code, your code will be easier to analyze, modify and test. Reduced complexity is accomplished by keeping your basic units (functions), simple and short, and by limiting the amount of logic each function contains.

Avoiding code duplication

Copy-pasted code is inefficient and error-prone, and frequently results in the need for later adjustments. In addition, duplication also increases the total source code volume, which has a further negative impact on the maintainability of a system. Avoid duplication by writing re-usable, generic code.

Guidance

Siemens and Software Improvement Group provide the best-in-industry software and guidelines to control and improve the quality of the architecture, maintainability and security of software developed with Capital Software Designer.

In SIG's book, *Building Maintainable Software*, consultants with the Software Improvement Group discuss 10 rules for improving software quality that are directly applicable for developers. In *Building Software Teams*, SIG consultants provide guidance on creating high-impact teams:
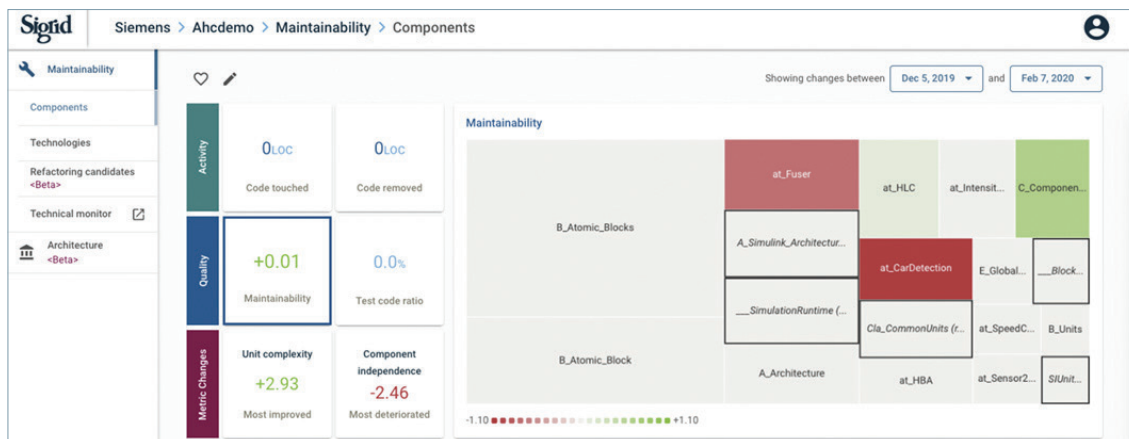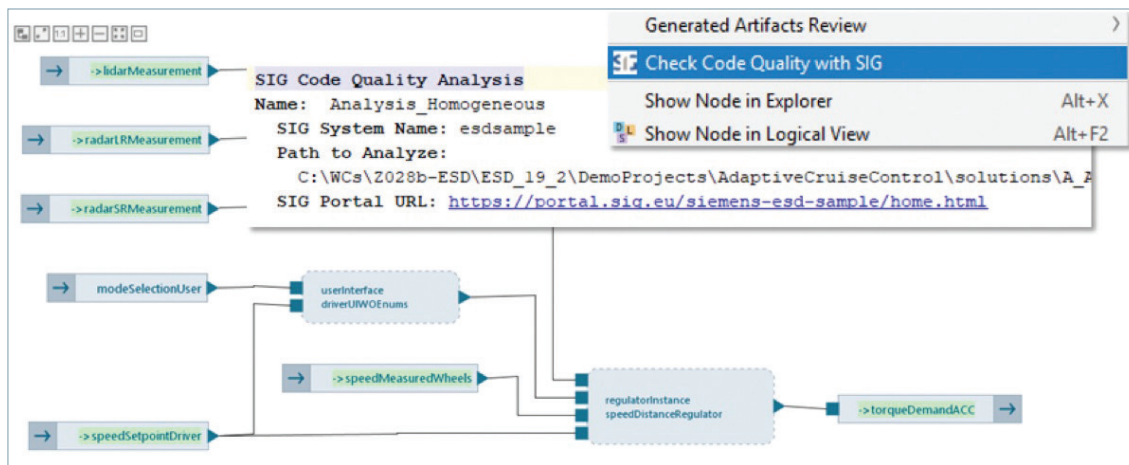
"Ten best practices for getting the development process right such that software is produced of consistent high quality. Our best practices do not just point a development team in the right direction. They are accompanied by a set of metrics that helps the team to consistently execute and monitor them."

Software quality analysis driven from Capital Software Designer

Siemens and Software Improvement Group have developed an integration between Capital Software Designer and Sigrid. This integration makes it very easy for verification and test engineers to trigger code quality analysis from Capital Software Designer, to navigate and inspect monitoring results and to feed those results back to the software architecture.

The integration associates the application sources with a SIG system, executes a local analysis in the customer's IT environment, reports the results to the Sigrid portal and provides a link to the monitor summary.

The monitor summary page below summarizes the aggregate software quality status of the embedded application. It provides entry points into details and violations and thereby gives clear advice for improving software quality on the levels of architecture and implementation.

Conclusion

In summary, the integration between Capital Software Designer and Sigrid for Capital Software Designer provides first-class insight and guidance into all aspects of software quality for your embedded software applications. Capital Software Designer addresses functional correctness and interfaces consistency during both pre- and post-implementation stages, whereas Sigrid addresses application maintainability in architecture and implementation.

In addition to out-of-the-box functionality in our solutions, Siemens Digital Industries Software and Software Improvement Group offer complementary services for preventing software decay, improving software quality and its architecture and moving legacy code towards the model level.

References

1. See *www.autosar.org*; Christine Jakobs and Peter Tröger, "Quo vadis, AUTOSAR?," INFORMATIK 2017, September 2017.

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software
Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

Contact
Siemens Digital Industries Software
Jan Richter
janrichter@siemens.com

Software Improvement Group
Michel van Dorp
m.vandorp@softwareimprovementgroup.com

siemens.com/software