SoC (System-on-Chip) – driving the adoption of MISRA / QA·C throughout the supply chain - from the OEMs right down to the silicon

Overview
Socionext is a recently created enterprise, formed by merging the LSI (Large Scale Integration) capabilities of FUJITSU Limited and Panasonic Corporation. The company is headquartered in Yokohama, Japan and designs and develops innovative embedded SoC (System-on-Chip) technologies, delivering imaging and networking solutions to automotive, industrial and consumer markets globally.

Socionext’s SoC solutions are delivered on chips that are embedded deep inside their customers’ products. The reliability of Socionext’s software (and hardware) is therefore of paramount importance, especially as it is very difficult to rectify any issues once these products have shipped and are in the field.

Deployment overview:
- Application:
  - ADAS (Advanced Driver Assistance Solutions) for Automotive
  - SoC technology for surround view systems & graphics display controllers
- Development team/environment:
  - 16 developers
  - 2 sites (Shin-Yokohama, Sendai)
  - Typical embedded application 250,000 LoC
  - Development suite eBinder from eSOL
  - MISRA C:2012 coding standard
  - QA·C static analysis tool
The introduction of automated static analysis

Initially Socionext used manual code reviews as part of their software development workflow to help to ensure that they were delivering high quality software. However, as the number of developers increased and the team became split across two sites, it became increasingly difficult to perform these manual reviews in a consistent manner.

In an effort to improve the effectiveness of these code reviews, and indeed as part of a broader initiative to embrace best practices and improve the overall software development process, the company decided to introduce static analysis. At this stage the company also recognized the significance of the MISRA coding standard, and the fact that their customers - especially those developing safety-critical solutions – were increasingly requesting this standard.

Given the number of MISRA rules and the fact that each rule requires strict inspection, it would not be feasible to comply (and indeed verify compliance) without an automated static analysis tool.

In 2010 Socionext therefore adopted QA·C / MISRA commenting that, “The combination of QA·C / MISRA is recognized as the de-facto benchmark and has been adopted throughout the Japanese automotive industry, spanning the complete supply chain from the OEMs right down to the silicon level”.

Integrated HMI and Around View System for Automotive
Deployment

Socionext has identified a common set of development rules that are used across all projects. These rules reflect the company’s past experience, and their desire to avoid ways of coding that have historically lead to bugs. The precise choice of rules is fine-tuned to be project specific, depending on the customer requirements. Typically 150 coding rules are adopted, and Socionext mapped 1300 QA·C messages to help to cover these. In some projects more emphasis will be placed on the Socionext ruleset, while others will see the more rigorous adoption of the MISRA ruleset.

Results

Socionext has made the following observations, based on their experience over the past five years:

- they have been impressed by QA·C’s ability to detect important coding issues, specifically including; variables that were not appropriately initialized, incorrect #if - #endif statements and invariant code with conditional branching that resulted in unreachable code.
- code reviews have become more effective. Socionext’s process now requires analysis with QA·C prior to a code review. This stops the team from being distracted by careless or basic issues and enables the reviewers to focus on more important topics, such as logic errors. Socionext estimates that this has reduced the overall development time by 10%.
- higher quality code is now being passed to testing, also making dynamic testing more effective and reducing the cost (and time) for rework.
- QA·C has proved to be a valuable training tool for both inexperienced and experienced developers and has accelerated the broader adoption of coding best practices.
- and finally, Socionext noted that they are now producing code which is easier to maintain. The code is better structured and more easily understood (beyond the original creator) and updates are quicker, less costly and lower risk.
In conclusion, Socionext initially adopted QA·C/MISRA because this was a requirement from their automotive customers. Five years on they have experienced the benefits, and are now proactively extending the adoption to other projects across the organization.

About Socionext Inc.
Socionext is a new, innovative enterprise that designs, develops and delivers System-on-Chip products to customers worldwide. The company is focused on imaging, networking and other dynamic technologies that drive today’s leading-edge applications. Socionext combines world-class expertise, experience, and an extensive IP portfolio to provide exceptional solutions and ensure a better quality of experience for customers. Founded in 2015, Socionext Inc. is headquartered in Yokohama, and has offices in Japan, Asia, United States and Europe to lead its product development and sales activities.
For more information, visit socionext.com

QA Systems and Programming Research Ltd
QA Systems is an authorised reseller of the QA·C / QA·C++, QA·Verify static testing tools and their compliance module add-ons, which are owned by Programming Research Ltd. QA·C ®, QA·C++ ® and QA·Verify ® are registered trademarks of Programming Research Ltd. These tools and this document are the copyright © 2015 of Programming Research Ltd. Third party trademarks, logos and trade names appearing in this document are the trademarks and property of their respective owners.

QA·C, QA·C++ and QA·Verify, offer the closest possible examination of C and C++ code. All contain powerful, proprietary parsing engines combined with deep accurate dataflow which deliver high fidelity language analysis and comprehension. They identify problems caused by language usage that is dangerous, overly complex, non-portable or difficult to maintain. Plus, they provide a mechanism for coding standard enforcement.

Contact Us
For further information regarding QA·C, QA·C++ and QA·Verify and compliance module add-ons, please contact QA Systems at info@qa-systems.com where appropriate QA Systems will re-direct you to Programming Research Ltd.

socionext™

www.qa-systems.com