

# Ensuring software quality

As the software component in new products explodes, how can companies ensure code quality when they are faced by time-to-market challenges? By **Neil Tyler**.

Over the past 30 years, the use of tools that help engineers to simplify and automate the design, development and integration process has propelled the development and manufacture of electronics and embedded systems. Today, when the concept of quality is becoming critical to all aspects of business, we are seeing the same developments when it comes to ensuring quality software development.

“Across all industries, software content is exploding and it’s not just the amount, but also the criticality of the software being deployed,” explains Jeff Fortin, head of product management at Vector Software. “More companies are looking to use software to differentiate the value being provided in embedded devices.”

The importance of software in delivering key features and systems, including safety critical functions across many different pieces of equipment and systems, means the traditional ‘artisan’ approach to software development is being abandoned.

In that approach, software development was viewed as more of a creative process. Today, software needs to be constructed with the precision and quality typically associated with modern-day manufacturing.

Regular news reports of product recalls due to software bugs, show that issues around testing remain. When investigated, many of these recalls can be attributed to issues that should have been picked up in



quality testing but ‘time-to-market’ pressures meant the software didn’t receive sufficient testing before being deployed.

“Customers are under pressure to develop products faster to meet the needs of new business models,” according to Fortin. “Succumbing to market pressure to develop software quickly is full of risks. Poor quality software leads to failures and

expensive product recalls, as well as reputational damage.”

Embedded systems commentator Jack Ganssle notes: “Eighty percent of embedded systems are delivered late and new code has 50 to 100 bugs per thousand lines.”

“We’ve seen examples of test code being left in place and then being deployed in the real system – suggesting a lack of time to test the

final configuration,” Fortin warns.

That raises the question of how do you address the needs of new business models while maintaining quality using existing tests?

“Do they need to run all the safety tests if they are looking to upgrade software in the field? How do we do the testing efficiently, whilst not impacting on the quality of that testing?” Fortin asks.

### Maintaining quality

Engineers and developers need tools that afford them the visibility of testing completeness as well as the generation of test cases for code that are not covered by existing test cases.

But, as experience shows, tests do not guarantee bug-free code at the end of the day.

“The majority of problems with an end product are usually caused by inefficient and incomplete software testing,” suggests Fortin. “It’s the efficiency or agility of this part of the development process which defines how quickly products are developed, assembled and then released – and it’s this part of the process that is under the greatest pressure to deliver.”

Without proper testing, quality issues will not be detected, which means that reworking and retesting will result in a slowing down of the speed of delivery.

“We are seeing more companies trying to achieve a six-week release cycle when, realistically, it will take three months to carry out every test that is required,” says Fortin.

Engineers need to be able to run tests on a simulation to verify the system’s behaviour and compliance with the requirements set out.

Furthermore, as products develop over their lifecycle with upgrades or availability of new features, changes will need to be made and implemented. These changes have code implications and engineers will need to understand where source

code changes will have knock-on effects and be able to create new tests to test the new code.

A designated software testing platform allows engineers to overlay intelligence that understands the smallest number of tests to be re-run by a change to the source code.

So, what can development teams do to ensure they have the agility, process and tools that allow them to ensure development tasks can be done in the fastest time and to the highest quality standards?

Various toolsets are available and they provide designers with a comprehensive range of automated software testing tools that enable the implementation of a complete and automated test infrastructure.

“VectorCAST, for example, provides users with the ability to check the health of their codes by using heat maps, dynamic dashboards along with in-depth analytics that provide actionable data for improving code quality,” explains Fortin.

“VectorCAST2018, our first major release since 2015, looks to consolidate many new features in a fully integrated package that offers embedded software development teams the functionality to create their own bespoke systems test and QA environment,” says Fortin.

“This new integrated environment can run whatever tests are required to ensure software meets the requirements specified, satisfies market demand and, in doing so, protects the brand’s reputation.”

Ease of use has been an issue, so this latest update has sought to simplify the user interface and looks at enabling more people to access the testing process.

“More people running tests means a greater chance of finding problems,” Fortin argues. “We wanted to make the testing process as easy to use as possible. Testing is complicated and we wanted to make the tools a little ‘softer’ for people



**“Looking to the future, a vast amount of test generation and reporting will be automated, the trickier part will be ensuring that the tests being run are the correct ones.”**  
Jeff Fortin

to use, as well as using automation more efficiently.

“We didn’t want to isolate the testing function to a few people, so with a simple click, all system level and unit level testing is available to users.”

A key component of VectorCAST 2018 is support for greater parallelisation in terms of both systems and unit level tests.

“We wanted to be able to run these tests efficiently and be able to test pieces of the code in parallel,” says Fortin.

Fortin believes that tools like VectorCAST will, in time, provide more of the background calculations testing, while providing continuous testing and integration. While these improvements should make the testing environment easier, users will still have to contend with a regulatory environment.

“We have to accept that regulators don’t move at the pace which the market demands, creating a stress between where the market wants to go and where regulations stand. But what we can do is make the work undertaken by engineers as efficient as possible.

“Looking to the future, a vast amount of test generation and reporting will be automated, but the trickier part will be ensuring that the tests being run are the correct ones, especially when new capabilities such as artificial intelligence and virtual machines become more common. The challenge then will be to create the test capabilities capable of working in those environments.”

Fortin believes customers now have a much greater awareness of the importance of getting software right.

“More people understand that quality has to be built in at the beginning of the software development process and maintained when it comes to ensuring the in-field lifecycle of embedded devices,” he concludes.