



“The most skilled developers will think of anomalies and if they build those cases, the chances are you will have a much better outcome.”

John Paliotta

John Paliotta is chief technology officer of Vector Software, a company he co founded in 1990. In 1994, the company built the first version of its VectorCAST product range. Called VectorCAST/Ada, the product was initially sold to customers building avionics, military and space applications. As well as serving as chief technology officer, Paliotta oversees all engineering and QA activities within the company.

Rethink test to drive quality

Chris Edwards talks with John Paliotta about the changing world of embedded software development.

Vector Software's CTO John Paliotta sees big changes coming in embedded software development that may overturn conventional wisdom on how the most skilled people on the team should be deployed. He also sees a push for quality emerging among the embedded systems community as consumers become jaded about the number of often baffling bugs in the products they buy.

"I don't care about the processor inside my phone, but I do care about my user experience."

That quest for quality is likely to lead to more organisations looking at automated test, Paliotta says. "Most of our business is still in high-criticality systems. However, we are getting enquiries from organisations who are developing an internal mandate for improving quality."

Paliotta draws a parallel with the hardware industry. "Manufacturing went through a whole renaissance with the Deming thing (Deming was an engineer focused on quality management). That was about finding the right cause of problems."

Before their adoption of quality driven processes, car makers found they had regularly occurring defects in parts such as brake rotors. "They would get thrown in a bucket and melted down, but why were 3% of them failing? After investigating, they might find they should recalibrate the lathes once a day. It's a quantifiable process and it leads to money.

"That kind of analysis has not been done with software. It's not being treated like a manufacturing process. People are not yet digging into those cost components."

Part of the problem of improving the development process is that organisations have not structured their processes to cope. Many manufacturers have slowly incorporated embedded systems into primarily mechanical systems to take advantage of better flexibility and control. But key parts of software development are often still thought of as being outside the organisation's skill set.

"In the early days, traditional manufacturers would outsource software development because it was sort of a nuisance. Companies have, over the past 10 years, brought software back in house, but still outsource test. Testing was always seen as a point solution and so was done in the final stage. Only then does full product testing happen for the first time.

"Now, those companies have larger code bases and distributed development teams, which exacerbates problems, and many are on a monthly release cycle. Things are changed and things fail. If developers are only performing a small proportion of the total testing, it's no surprise that things are breaking late in the cycle."

When developers are involved in the test process together with automation, tests can be run as soon as things are changed to pick up problems that break old code.

Bringing developers into the test process has further ramifications as it puts more emphasis on unit testing. "Unit testing is funny; it gets a bad rap a lot of times. People often treat it as dumping data into a function, with the goal of getting all the instructions in the function executed," says Paliotta, "but that misses the meaning of the idea of code units. A unit should be something that abstracts detail."

Some units can be tested in isolation. Others need to be grouped before meaningful tests can be applied, as the purpose of the unit test is to determine the unit's readiness for integration. Paliotta says: "It's about deciding what a unit is. To some extent, the line is blurred between unit and integration testing."

The ability to decide what a unit is points to a shift towards the greater use of requirements, which Paliotta hopes will be the case. But the requirements need to be expressed in a less clumsy way than has often been the case. "Historically, requirements have been text based. Why does someone have to read them at each stage? Why do I need need to write three paragraphs of text that someone else is going to skim through to see which parts are relevant?"

Instead, those high level requirements should be decomposed into useful unit test definitions, as well as used to drive integration tests. "This gets to another point that I've been talking about; if you think of C or C++ as languages, generally people will build interface level headers that define methods and data types. They will create charts to show interactions.

"The valuable artefacts that are normally put under configuration control are the code, not the test artefacts," Paliotta says. Yet the tests encapsulate the job each unit and subsystem is meant to perform. "It's one of the ideas behind Agile development – you define interfaces and build test cases for them. The important elements are the interfaces.

"The logic of the function is kind of irrelevant – as long as it works. If I have my two best guys defining good interfaces, I can have the two least experienced doing the implementation," Paliotta argues. "In my career, it's always been the smartest guys doing the implementation. Testing is less valued and people who are less skilled are put into defining interfaces. We need to turn the industry on its head to say best people will design the interfaces and the test cases.

"The good developer who today builds code will think of edge cases; the less skilled developer will most often not. They will think in terms of nominal. The most skilled developers will think of anomalies and if they build those cases, the chances are you will have a much better outcome."

The result could be teams with very different structures to those used today, but the consequences could be similar advances in overall quality seen by the car industry when it examined its processes.