Cantata++

Critical Software
Critical Care

Case Study: GE Medical Systems

Critical Care Monitor in Use

This Case Study features Datex-Ohmeda, now part of GE Medical Systems, whose business is systems, equipment and services for anaesthesia and critical care. Their products include patient monitors, anaesthesia machines and ventilators.

Patient monitors are used to measure and monitor several clinically important patient parameters, such as heart rate, blood pressure, oxygen saturation of the blood, body temperature, inhaled and exhaled gas concentrations, ECG waveform, EEG waveform, and depth of anaesthesia. Anaesthesia machines administer a selected mixture of gases and anaesthesia agents to the gas stream that the patient inhales helped by an inbuilt ventilator.

All Datex-Ohmeda's products are required to meet the strict quality requirements imposed on software in safety-related medical devices, by the US Federal Drug Administration (FDA). This means that a detailed risk assessment exercise is conducted, in line with the FDA's General Principles of Software Validation; Final Guidance for Industry and FDA Staff [1]. This process identifies safety-critical components that require thorough unit testing, and those that can be less rigorously tested.

Datex-Ohmeda has developed a software platform for a new generation of medical devices. This platform will allow shorter development cycles for new products and new features, such as new measurement parameters. We examine how Datex-Ohmeda selected and used Cantata++ for the platform project and how its testing strategies have developed.

SAFETY CRITICAL BUT FLEXIBLE TESTING REQUIREMENTS

Most of Datex-Ohmeda's products are classified at the highest level defined in the FDA guidance for software contained in medical devices (Major Level of Concern). For example in patient monitors, a failure to raise an alarm can cause harm to a patient. The software platform underwent a risk assessment that identified approximately 10% of the software as implementing safety-critical functions. Full white-box unit testing of this code was therefore required, in addition to functional black-box testing of all of the code at sub-system level. The software was developed in C++ on a Windows NT host using the Microsoft Visual C++ compiler, and cross-compiled using the Diab Data and CAD-UL C++ compilers. The target system is a Mpc823e PowerPC target running the Nucleus Plus RTOS. Risto Repo the Software Process Improvement and Verification Officer of Datex-Ohmeda said "**We needed a C++ testing tool that was flexible enough for testing both object-oriented C++ components and whole sub-systems in both host and target environments, and also across different sites**".

SELECTION

Risto Repo had known about Cantata++ from his previous job with Space Systems Finland. In December 2000 he requested a competitive evaluation of Cantata++ against another tool on two sites: Helsinki and Louisville, USA. The evaluation resulted in early 2001 in the selection of Cantata++. Risto Repo explained the reason: "**Put simply, Cantata++ did what it promised for C++**".

Datex-Ohmeda and IPL then worked closely together undertaking a detailed capability investigation of cross-compilers. At the time, these were far from being ISO C++ compliant. A decision was taken to carry out testing with Microsoft Visual C++ and the Diab Data cross-compilers, despite the latter compiler's limited support for templates (at that time). Datex-Ohmeda purchased Cantata++ in June 2001 for use at Helsinki, and Madison and Louiseville in the USA.

CANTATA++ IN ACTION

Cantata++ was put to use in white-box component testing of the safety-critical code in both host and target environments. These components were first unit tested on the host (Windows NT) and then the same tests were re-run on the PowerPC target. All tests required 100% coverage of entry-points, statements and decisions. The non-safety-critical code was subjected to black-box functional tests at sub-systems level, using Cantata++ coverage analysis. Sub-systems typically contained between 10 and 80 classes.

Engineers at the three Datex-Ohmeda sites were soon up and running, as Jutta Luosta, a leading test developer in Helsinki put it "**Cantata++ was very easy to get to know**" adding that "development testing of subsystems was really quite fast with Cantata++". Tuomas Jyrske who attended IPL training courses said "Working for a time with Cantata++ before coming to the training courses was better, as I brought lots of questions and was able to get really good answers at the training."

He then acted as the Helsinki technical lead, and passed on his experience adding “After the training course at IPL, I passed on my experience with help from the training materials”.

The C++ code became more object-oriented with the growth of source code generated by the project’s UML modelling tool. Cantata++ was shown to work well with both manually and automatically generated source code. Risto Repo points out that while each site started off with different approaches to testing, “We learned later from colleagues at different sites what conventions worked and what did not”.

EASY, FLEXIBLE, WELL SUPPORTED

One particular reason for the easy acceptance of Cantata++ by developers is well expressed by engineer Erno Muuranto “For testing with Cantata++, it is nice that I can stay in the same environment (Microsoft Visual C++) that we use for programming.” He adds that “**Cantata++ has a lot of powerful features, but the tester can use as much or as little as you need to get the job done**.” Risto Repo put it a different way: “A real benefit is that Cantata++ makes people write tests in a consistent way because it gives engineers a proper framework, but it does not require everyone to use it in the same way, so it is still flexible”.

For many of the developers the introduction of Cantata++ has also had other benefits that we have seen at many companies before. Tuomas Jyrske comments that “Cantata++ testing has had a positive impact on the way we write the code, reducing coupling between objects and making isolation testing with stubs easier.” Tuomas added that such a full-featured testing tool needed good support for developers to trust it, and he was pleased to say that “Cantata++ Technical Support always gave a quick response to enquiries”.

OUTSOURCING TESTING

From the very beginning Datex-Ohmeda, knew that even with three development sites they were going to need to bring in additional resources, especially for testing. One early problem that they faced was the lack of implementation detail in the System Requirements Specification. When implementing these requirements the detailed design specifications (from which test cases for the safety-critical classes were derived) lacked stable interface definitions. As Risto Repo explains this became a problem when outside companies were given batches of code to unit test: “The design of interfaces was fluid, and development changes were so rapid that component tests became quickly out of date. We were a little bit surprised at how much of a problem this was.”

Datex-Ohmeda actually used both Finnish and Indian companies for outsourced testing work on the software platform. The most instructive lessons came from the functional sub-system tests: as Jutta Luosta describes “Initially we just did not specify coverage requirements because there were so many things that were new being introduced”. After initial tests had been delivered to Datex-Ohmeda she re-ran the functional tests from both companies using Cantata++ coverage analysis. She says “We got very descriptive results from the comparison of subsystem level coverage figures”. The results actually ranged from 65% to 92% entry-point coverage; with one company’s tests producing much lower coverage than the others. Having also analysed in detail the statement and decision coverage metrics, Datex-Ohmeda then revised the terms and conditions of the outsourced testing contracts to require specific entry-point and statement coverage targets. Jutta Luosta comments, “Now, for upgrading the tests as new changes are made, we always set coverage targets”.

CONCLUSIONS

Since the initial phases of the project, Cantata++ use has been successfully extended to further phases such as New Patient Monitor targeting Renesas H8S, and other projects such as the safety-critical Non-Invasive Blood Pressure measurement unit. Risto Repo concludes by explaining Datex-Ohmeda’s positive views on using Cantata++ in the future: “**I expect that use of Cantata++ will continue to grow at GE Medical, as I hope that all new projects will consider using Cantata++**.”

IPL would like to thank GE Medical Systems for permission to report on their experiences with Cantata++ and take this opportunity to wish them well with their products and services.

The text for all IPL product case studies is agreed and approved by our customers.

Patient Monitoring and Anaesthesia

[1] General Principles of Software Validation; Final Guidance for Industry and FDA Staff issued 11 January 2002 available at http://www.fda.gov/cdrh/comp/guidance/938.pdf

FURTHER INFORMATION

Software Products Group
IPL Information Processing Ltd
Eveleigh House
Grove Street
Bath
BA1 5LR
United Kingdom

Telephone: +44(0) 1225 475000
Facsimile: +44 (0) 1225 444400
Email: tools@iplbath.com
Web: http://www.iplbath.com/tools

Certificate Number FM 01589