IBM Rational Test RealTime Simplifies Compliance with DO-178B Regulations for Embedded Avionics Software

Overview

- ***The Challenge***
  To achieve FAA approval, embedded software for avionics systems must comply with DO-178B guidelines for requirements-based testing and code coverage analysis. Because manual code coverage analysis is tedious, time-consuming, costly and error-prone, automating this process represents a significant opportunity for improved efficiency.

- ***The Solution***
  IBM® Rational Test RealTime™ automates code coverage analysis of embedded software by highlighting portions of code that have not been tested. IBM Rational® RequisitePro® for requirements management and IBM Rational ClearCase® and IBM Rational ClearQuest® software configuration management tools provide complementary capabilities to meet D0-178B requirements for requirements-based testing.

- ***The Benefit***
  Using an integrated solution including Rational Test RealTime and other IBM Rational tools, software development organizations have streamlined the FAA certification process, while improving efficiency, time-to-market, and cost-effectiveness. One such organization completed its certification four months ahead of schedule, and enabled one engineer to complete tests in one month, something that typically required three months by a team of four.

Early in 1993, the Federal Aviation Administration (FAA) recognized DO-178B “Software Considerations in Airborne Systems and Equipment Certification” as its preferred guideline document for the development of airborne computer software. This document, published by the Radio Technology Commission for Aeronautics, Inc. (RTCA), has become the standard used by development organizations to achieve FAA approval for the embedded avionics software they build.

Section 6.4.4 of DO-178B requires two specific forms of test coverage analysis: requirements-based test coverage analysis and structural coverage analysis. Requirements coverage analysis is used to assess the ability of requirements-based testing to verify that software requirements have been implemented, and links test cases to requirements. Structural coverage analysis is used to identify what source code was executed during the requirements-based tests, and links test cases to the source code.

Many avionics development teams building DO-178B compliant software spend a significant amount — in some cases more that 50 percent — of their budget and time on testing, because they rely primarily on manual testing methods. The ability to automate many of these activities not only helps these teams deliver certifiable software more quickly, it enables their organization to reallocate valuable testing resources when and where they are needed.

Tom Sawyer, Supervisor of Software Development for Moog Inc., notes, “Manual structural coverage is very time-consuming, expensive, and laborious. It is almost an impossible task for large complex systems. Many of the projects we work on are DO-178B contracts for safety-critical airborne systems. We test levels A, B, and C code, including the highest safety critical software up to Modified Condition/Decision Coverage (MC/DC). These projects include traditional requirements-based testing, but they also have a structural coverage requirement, which greatly increases the scope of what you need to do for testing."

Applying IBM Rational Test RealTime

To address the challenges inherent in DO-178B development for FAA compliance, Sawyer began using IBM Rational Test RealTime, a cross-platform solution for component testing and runtime analysis designed specifically for those who write code for embedded, real-time, and other types of cross-platform software products. Rational Test RealTime provides automated code coverage analysis capabilities that enable easy and effective structural coverage analysis by identifying which portions of a program have been tested.

Sawyer developed a system for converting existing test inputs and expected results into Rational Test RealTime scripts. This system enables testers to receive structural coverage, testing credit without having to rewrite existing unit test scripts. "One of the first benefits of Rational Test RealTime was that it allowed us to get credit for what we had already done. The other major benefit was that it allowed us to perform tests that could not be accomplished in our lab environment. For example, in the lab it is very difficult to test analog signals at a specified boundary. We wrote unit tests in Rational Test RealTime to simulate and test values exactly at each boundary and for other conditions that cannot be tested in the lab," says Sawyer.

Sawyer and his colleagues are using Rational Test RealTime with IBM Rational RequisitePro for requirements management, IBM Rational ClearQuest for defect tracking and change management, and IBM Rational ClearCase for software asset management. Together these tools provide not only the means to satisfy code coverage requirements, but also establish clear traceability from testing to requirements, code and

defects — enabling the team to effectively manage the FAA audit process and rapidly achieve FAA certification for their software. "By using the tools we were able to keep costs in check and complete certification in record time, about four months ahead of schedule," Sawyer reports.

Conversion of Test Assets

The process Sawyer used to automatically create Rational Test RealTime test scripts from existing test assets is embodied in a tool he developed called the AST or Auto-Script Tool ®. The tool takes test inputs and expected results already developed for the lab using National Instruments LabVIEW and generates Rational Test RealTime scripts, which are then executed to gain structural coverage credit. Sawyer explains, "In the lab, the test inputs are stored in Excel spreadsheets, which interface well with LabVIEW. There can be two to three hundred test inputs along with their initial values to start the test. These tests also require you to supply the number of cycles that you want to run during the test, and a set of expected results, which can also number in the hundreds. After the test run, each result is compared with the expected result to see if it is within tolerance, and is assigned a pass or a fail."

The Auto-Script Tool uses those same lab spreadsheets containing test inputs, number of frames and expected results, and converts them into Rational Test RealTime scripts. Each test input and expected output becomes a variable. The tool also transfers header information, as well as, the number of frame cycles the test will run. The script is then run in the Rational Test RealTime environment using stubbed code. Because the script is running on a PC, it does not have access to embedded hardware interfaces. Sawyer explains, "If the code needs RAM, I build my own RAM in Rational Test RealTime. So, instead of being hardware RAM, it is Rational Test RealTime RAM in my stubbed software."

Depending on the Rational Test RealTime Target Deployment Port (TDP), the tool compiles software with different target compilers, loads the specific debugger and generates an output stream that is captured into Rational Test RealTime to produce the pass/fail report. Sawyer notes, "We can compare the pass/fail results from Rational Test RealTime with those achieved in the lab. But the key benefit is that at the end of each run we have code coverage results unattainable in our lab. We now have a way to gain coverage credit for tests run in the lab. Whatever testing cannot be done with a spreadsheet conversion I accomplish by writing my own script in Rational Test RealTime. By merging those common assets we can get 98 to 99 percent coverage and very quickly get credit for structural coverage testing. The remaining one percent of code is usually something like a default in a case statement which will never be reached, because the inputs are pre-checked. That last one percent can be checked manually fairly easily."

Immediate Feedback Enables an Iterative Approach

Sawyer has implemented an iterative process in which he leverages code coverage results from Rational Test RealTime to identify and eliminate any gaps in the test portfolio. "In structural coverage testing you go down every path, and any code that has not been executed is dead code, deactivated, or reflects a missing requirement. Compliance with FAA requirements does not permit any of those situations," Sawyer notes.

He continues, "Today, as the software is under development, someone can be writing Rational Test RealTime test cases to unit test that software. So I immediately have coverage information. The unit tests are totally repeatable and the certification authority can see that we are doing a good job in our unit testing because we can show all of the test artifacts early in the software development cycle. In addition, once you have your first coverage information, you hold a structural coverage analysis review. At the review you look at every line of code — color coded in Rational Test RealTime — and see what code has not been touched during the testing. You decide right then, if you are missing requirements, need additional robustness testing, or if you have a dead code issue. This is a major step in improving what we do as testers. When you look at the actual code in Rational Test RealTime, if it is green that is great, because you know you have been down that path. If it is orange it means some of the path has been gone down, but not all of it. And when the code is red, that means it has not been touched. As soon as we get our software in place and we have some of the tests we run them through Rational Test RealTime and we are able to get a color coded representation of what the coverage is to date."

Obtaining valuable test results earlier in the development cycle helps teams find and address problems when they are less costly to repair. "As you start putting the modules together to build a Computer Software Component (CSC), and then build those CSCs into a Computer Software Configuration Item (CSCI), you are also building a collection of unit tests into a software integration test suite. All the unit test assets up to a CSCI level are then passed to the verification and validation test group. The goal is to use Rational Test RealTime upfront in the software development process rather than waiting until all the software is developed. When you wait until the end of the development cycle to test, any problems you find are much more expensive to go back and fix," Sawyer adds.

Qualification Kit an Essential Part of Certification

The FAA requires software being used on certified critical airborne systems to follow the DO-178B standard for all software verification tools, including IBM Rational Test RealTime. Tools used in the verification process have to be qualified for use. While the qualification process of tool verification is the responsibility of the organization developing the DO-178B application, IBM Rational assists in this process by providing qualification assets in the form of a Qualification Kit. "We used the Qualification Kit to qualify the tool. Without the kit we would have had to develop one on our own and risk schedule challenges getting to certification," confirms Sawyer.

Rational RequisitePro Simplifies Audit Process

There are typically six to eight hundred requirements in the CSCIs that Sawyer tests. Each of these requirements is managed and tracked in IBM Rational RequisitePro to enable traceability of requirements throughout the project. Sawyer explains, "We build the requirements documents in Rational RequisitePro and we use that to trace from our system level requirements to our software requirements. When we put our requirement documents in Rational RequisitePro, each requirement is tagged. The next level design document and subsequent code references these tags. If you go to the code you will see a header or a comment that tells you what requirement that specific code is satisfying. Finally, we test and reference those same tags. Rational RequisitePro is an effective way to keep track of all that tracing."

This traceability plays a key role in meeting the demands of the FAA's audit process. Sawyer explains, "When I sit down with the Designated Engineering Representative (DER), representing the FAA, the first thing they want to do is trace requirements. They will say 'For these three requirements, I want to see where they are in your design and in your code. I want to see when they were tested, and I want to see the results of those tests.' It is very impressive when you can sit down with the FAA representative and show them all the processes are in place. With Rational ClearQuest, Rational RequisitePro and Rational Test RealTime we can show how we ensure that when a problem report is written that it was fixed and where it was retested. We can show the regression testing that we do. In many cases, because it is critical software, every time we make a new release of the software, it's entirely recompiled. And so we have to test the whole suite of code again."

On Sawyer's projects, requirements do not change frequently but when they do, Sawyer performs suspicion analysis in Rational RequisitePro to determine which requirements were affected. He notes, "After a safety-of-flight test, there can be requirements changes. And suspicion analysis is especially useful in a validation and verification review. Any requirement that is marked as suspicious by Rational RequisitePro has to be resolved before the software is released."

Streamlining Development with Activity-based Change Management

Seamless integration between IBM Rational tools helps eliminate communication problems by automating workflow and knowledge transfer. Sawyer notes, "I have found that problems are introduced wherever you have hand-offs in development. If you minimize the number of hand-offs, you minimize the problems. We keep our process as seamless as possible — not throwing anything over the wall — by utilizing the tools."

An example of this tool integration and automation is activity-based change management, which provides out-of-the-box process support for managing change with IBM Rational ClearCase and IBM Rational ClearQuest. "Rational ClearCase is used for version control of all assets, including test assets. To enable better communication between groups, Rational ClearQuest is used for problem reporting and tracking," says Sawyer.
With Rational ClearCase and Rational ClearQuest, an activity is automatically associated with its change set, which encapsulates all the correct versions of the project assets used to implement the activity. "Often if you leave an option open for a software developer to not follow check-in and check-out processes — if it is not built in as part of the process — they won't do it. Rational ClearCase and Rational ClearQuest help integrate the development process. You can only check-out code by referring to a problem report that gives you authorization to make a change. You are given a problem report, you log on, check-out the code, fix it, and check it back in. You follow the course of action using Rational ClearQuest to track each step of the activity request review process and sign off on the activity electronically," Sawyer explains.

Achieving Goals Months Ahead of Schedule

The ease of use, increased efficiency and productivity provided by IBM Rational Test RealTime and other Rational tools have enabled Sawyer to rapidly realize significant benefits in both requirements-based testing and structural coverage analysis. Sawyer concludes, "The use of these tools has given us a way of accomplishing what we needed to do. Rational tools, along with the process and Auto-script Tool I developed, enable us to get credit for work we had already done and for having a more seamless process. In one month with one person, I can do the work that required four people working three months. And the assets are reusable — for future regression testing, all I need to do is an overnight test run. It would also be much more expensive if we had manual structural coverage analysis. Manual coverage methods I have witnessed at other developer sites have taken them man-years for what I basically did by myself in a couple of months. As a result, we were able to receive certification from Transport

Canada and the FAA in record time, months ahead of schedule on our latest contract."

IBM Corporation
Route 100
Somers, NY 10589
U.S.A

Produced in the United States of America
02-05