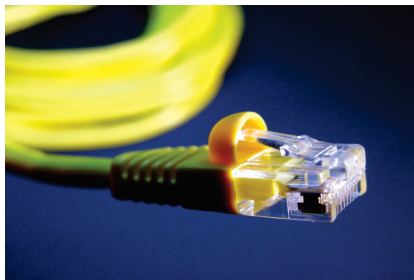


## SECTOR IT & Telco

# Case Study



Marconi Communications



### ABOUT THE COMPANY

Marconi Communications, a division of the UK's GEC company, is one of the world's major suppliers of telecommunications equipment to telecoms network service providers. Its Data IP & Voice Networks Business is very active in the provision of Intelligent Network (IN) products which enable customers, such as British Telecom (BT), to operate, maintain and extend services to millions of domestic and commercial subscribers.

Marconi Communications operates several UK development sites including a large site at Poole on the South coast.

***"We found errors using Cantata that we know we would otherwise have missed until later"***

### Background

The 'IN' products of which Marconi Communications are particularly proud are GAIN INventor and GAIN INjector. The INventor is a Service Creation Environment that allows telecommunications providers to prototype new service ideas such as Personal or Freephone numbers. The INjector is a Service Control Point which enables those services to be implemented in the telecommunications network. These products are very software intensive, so a constant theme of Marconi Communications' software process improvement strategy, is the search for 'better' ways of producing large volumes of reliable C++ code, in a cost-effective way.



### Two Schools of Testing

Marconi Communications were long-standing owners of Cantata licences but the use of the tool was relatively patchy. This was largely due to some misplaced expectations about what a test tool could achieve. There were also two contrasting schools of thought on how to go about testing with Cantata. One school was in favour of writing code, debugging it informally, and then writing a test harness to show that it worked. The other preferred to create and maintain tests parallel to code production, so that software was tested as it was developed.

Motivation for a fresh look at the coding process came when new the releases of INventor and INjector were planned to provide new IN functionality. A continuous improvement strategy is part of Marconi Communications' business process and here, it was specifically targeted at identifying faults earlier - so as to reduce the time and cost of the work.

The contrasting approaches were studied, and it was concluded that the second approach led to better results and provided better long-term maintainability. This conclusion was based on the analysis of previous projects and the observation that testing needs to serve twin purposes: one purpose is to ensure that code does what it is required to do, and the other is to ensure that changes made to previously working code do not introduce errors.



## Why Unit and Integration Test?

To ensure that both these objectives are met, tests need to be designed and be in place at the start of coding, and to allow for frequent re-running. Cantata met this general aim well, due its scripted approach with stubbing.

### Cantata Beta Trial Participants

Two engineers, Dave Parker and James Legg, attended the beta trial training and conducted an initial evaluation of Cantata. According to Parker, *"Cantata worked well, and any problems found were quickly dealt with."* The software team made the decision to stay with the new tool. Interestingly, at this point, another Marconi Communications team producing further INventor facilities came on board with Cantata.

### So What Has Cantata Done For Them?

The volume of new code created and tested with Cantata had reached approximately 10,000 LOC. At the peak, ten developers were using Cantata. In terms of the main aim, which was to reduce development time by finding faults earlier, there is no doubt that Cantata was very useful. Dave Parker noted, *"We found errors using Cantata that we know we would otherwise have missed until later."* He also added the useful observation that, the very process of devising the tests revealed errors without even using the tool! This occurred because test planners were obliged to run the test scenarios 'mentally', thus they would identify design errors before the code was even written. On the secondary objective, to check the continued correctness of pre-existing code, Cantata provided for test automation; this made regression testing at unit level (class and/or domain) more simple.

In terms of specific effective Cantata functionality, Parker mentioned two areas. One was the Cantata 'White Box' testing feature. This opened up the private components of a class and enabled the IN team to carry out useful checks on the internal states of the code. This type of checks would otherwise have been very difficult or even impossible. The other feature to have found favour is 'Wrapping', which Parker described as: *"Very useful, especially for checking the messages sent from one process to another, and also for simulating responses from external processes to the process under test."* At times, the effort of setting these up could seem an overhead (mainly because of the structure of the Marconi Communications software build environment) but once implemented, the repeatability proved to be a significant bonus.

The test coverage side of things was less used, mainly because it was available rather late. Parker said that he found: *"Statement coverage (at 100%) good as a general measure of confidence but that decision coverage was better at spotting unexpected scenarios."*

Companies in the IT & Telco sector perform unit and integration tests to make their testing more efficient.

### Reduce Commercial Risk

- > Increase software quality by thorough unit testing to prevent impacts on corporate and brand reputation
- > Use state of the art testing to prevent against fitness for purpose litigation
- > Systems are too complex to thoroughly test, so they are decomposed into testable units

***"Cantata worked well and any problems found were quickly dealt with"***

### Minimize Overall Testing Cost

- > Testing code earlier is cheaper (less re-work costs)
- > Most effective use of testing resources (unit tests do not have to wait for system builds)
- > Increases overall software quality (more thorough testing is possible at unit level)

All case study text has been approved by the relevant customer.

QA Systems acquired the Cantata business taking over all development, support and sales from IPL in March 2012. Cantata is the extension of the Cantata++ tool.