

DO-178C + UML

Rikkert van Erp reports from Düsseldorf

The ignite 2010 conference in Germany featured several presentations about emerging standards



The **ignite** conference, the new name for what many PT readers will know as ICSTEST or SQC, is still the most important one for testers in Europe despite growing competition. It took place in Düsseldorf over three days at the end of April with nearly 70 presentations plus workshops and around 40 exhibitors.

Peter Heller of **Airbus Operations** and **Sven Nordhoff** of **SQS** discussed DO-178C, the soon to be published fourth version of DO-178 *Software Considerations in Airborne Systems and Equipment Certification*. PT readers may have heard of this standard on training courses: as one would expect, it is the result of an immense amount of painstaking work and considered a superb example. Many of the test design techniques used in all sectors were developed in the safety-critical industries and it's easy to argue that should be extended to life cycle models, activities and evidence for software certification. The structure of the long process of documentation and meetings being used to develop and reach consensus on the new version is interesting, if not particularly relevant to those who will use it or try to learn from it: more exciting to testers in general was to see how DO-178 relates to other standards that are less well known outside avionics. DO-278A *Guideline for Communication, Navigation,*

Surveillance, and Air Traffic Management Systems Software Integrity Assurance and ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment seem well worth a look.

The next presentation, by **Jan-Hendrik Boelens** of **Eurocopter Group**, was more technical and accessible. It introduced four technology-specific supplements to DO-178C: *Model-Based Development, Object-Oriented, Formal Methods* and *Tool Qualification*. The use of all of these is challenged and sensible constraints are placed upon it by the standard. For example: all elements in a model must be classified as defining, or not defining, requirements, and the former group must be subject to strict traceability just as requirements themselves are; all reused and inherited code must have defined requirements; a mathematical proof can

replace some testing provided the requirements it assures are defined mathematically; requirements of test automation tools must be defined in the same way as for the SUT. Would stipulations like this help to prevent expensive failures if enforced in testing of business systems?

The first name we at PT look for in a conference programme is **Professor Harry Sneed**. We've been privileged to report on his presentations twice before (issues 14 and 17) and they are everything we think a testing talk should be: inspiring, very practical and acidly funny. Prof Sneed's current position is at **Universität Regensburg**; we say "current" because he once related how developers refused to work while he was on the premises after he detected large blocks of very rarely executed code and we think it's likely testers will have felt the same way when he discovered the true coverage achieved by their tests. This time he demonstrated how to measure the size, complexity, completeness, consistency and quality of UML models, so Bavarian systems architects may already be sharpening knives ■



Peter Heller of Airbus **Jan-Hendrik Boelens** of Eurocopter Group **Klaus Kleemann** of SQS **Sven Nordhoff** of SQS

Some of the presentations from Düsseldorf plus many new ones can be seen at other forthcoming ignite conferences worldwide, including in London on 4th October 2010. Visit <http://ignite-conferences.com> for full details.