The PET Project Summary

Prevention of Errors through Experience-driven Test Efforts

The PET project was set up to improve software quality through improved testing. It was funded by the Commission of the European Communities (CEC) as an Application Experiment under the ESSI programme: European System and Software Initiative. The PET consortium consists of two large Danish companies both of which have subsidiaries in many countries around the world, and export the major part of their products.

Through a rigorous analysis of problem reports from previous projects the companies behind the PET project achieved a step change in the testing process of embedded real-time software. The objectives were to reduce the number of bugs reported after release by 50%, and reduce the hours of test effort per bug found by 40%. Both of these goals were met. The actual numbers achieved were **75% less bugs reported**, and a **46% improvement in test efficiency.**

The problem reports were analysed and bugs in them categorised according to Boris Beizer's categorisation scheme [BB]. It was found that bugs in embedded real-time software follow the same pattern as other types of software reported by Boris Beizer.

It was also found that the major cause of bugs reported (36%) are directly related to requirements, or can be derived from problems with requirements.

The second largest cause of bugs (22%) stems from lack of systematic unit testing, allegedly because of the lack of tools for an embedded environment. It was found that tools do exist to assist this activity, but their application requires some customisation. A unit testing environment was introduced based on EPROM emulators enabling the use of symbolic debuggers and test coverage tools for systematic unit testing using **LDRA Testbed.** The unit testing methods employed were: **static and dynamic analysis**.

It was demonstrated that the number of bugs that can be found by static and dynamic analysis is quite large, even in code that has been released. The results that were found are applicable to the software community in general, not only to embedded real-time software, because the methods and tools are generally available. Finally a cost/benefit analysis of the results with static and dynamic analysis indicates that there could be an immediate payback on tools and training already on the first project.

The efficiency of **static analysis** to find bugs was very high (**only 1.6 hours/bug**). **Dynamic analysis** was found to be less efficient (**9.2 hours/bug**), but still represented a significant improvement over finding bugs after release (14 hours/bug). A **test coverage** (branch coverage) of **85%** was achieved for all units in the product, which is considered best-practice for most software, e.g. non-safety critical software.

LDRA Ltd. Portside, Monks
Wirral, CH41 5LH, UK.
t : +44 (0)151 649 9300
f : +44 (0)151 649 9666
w : www.ldra.com
e : info@ldra.com