A discipline finding its identity

As schedule pressures are applied to software development projects, quality is getting squeezed out. Efficient verification and validation techniques can help solve this problem, says Donna O'Neill

Donna O'Neill is the managing director of IV&V Australia. She has more than 15 years' experience in software testing and software IV&V, in a wide range of application areas. In her consulting, she evaluates clients' testing strategies, recommends improvements, and provides training and management to project teams. She is the program co-chair of the AsiaSTAR 2001 software-testing conference.
E-mail: donnao@ivvaust.com.au

Software testing in Australia is a discipline searching to find its identity. It is widely accepted as a necessary activity on development projects; however, its role is often ill defined and its potential is greatly underestimated.

In all parts of the ICT (Information and Communications Technology) industry, testers find themselves caught in the quality versus schedule conflict, where they have been tasked with "ensuring that we deliver a quality product", yet they are not given sufficient time, support or resources to effectively achieve this goal. The extent to which this occurs varies somewhat from one market sector to another, and with the maturity level of an organisation's development process. However, inefficiency and frustration are still the norm.

Where does testing fit in?

Although there are pockets of excellence, across the ICT industry as a whole, few projects integrate software-testing activities into their development life-cycle as well as they could.

Testing is generally seen as a separate task, as an adjunct to development, which occurs after development has been largely completed. While this may be appropriate for some types of testing (e.g. user acceptance testing, performance/load testing), these projects are missing out on much of the value that a broader testing strategy could offer.

This attitude can be seen in the way project schedules are defined. On most projects, the development team has a fairly clear idea of their own life-cycle process, and produces detailed schedules with quite complete work breakdown structures. However, the corresponding test schedules are often very different.

At the lower end of the maturity spectrum, we see schedules with a task labelled "test" that sits somewhere near the end of the life-cycle, before release. This task does not provide any indication of the planning, test design, test data collection, test environment set-up, technical reviews and reporting activities that will (or should) occur during the "test".

Even on more mature projects, where more complete test schedules are defined, the test and development schedules are often separate, with few indications of the actual dependencies between the teams. Few projects formally define a project-wide test strategy (e.g. in a master test plan) that incorporates a multilevel approach that targets test effectiveness:

- The use of "white-box" testing techniques at the unit test level, by the developers, to ensure the code is robust and reflects the intended design.
- The use of "black-box" testing techniques at the functional and system test levels, by independent testers, to verify during functional testing that the stated requirements have been implemented, and to validate during system testing that the product meets its intended purpose.

Certainly, it is rare for testers to be involved with reviewing requirements for "testability" (and to be able to recommend changes to requirements), even though these requirements form the basis of their tests. Without clear and testable requirements, testers can only ever show that the system "does what it does" rather than fulfil their real and vital role of ensuring that the system "does what it is supposed to do".

The most critical repercussion

of the lack of integration between development and testing is that projects end up doing less testing.

Development tasks slip, squeezing testing into a decreasing time-window before release. Testers have to be fast on their feet, quickly abandoning all "non-essential" tasks and tests, focusing their efforts on key and critical functionality. They are forced to react to a rapidly changing set of inputs rather than complete their jobs in a planned and systematic manner. They run fewer tests and the project ships more bugs.

Where do testers fit in?

It takes an experienced test manager to be able to keep this situation from degrading into chaos. In many sectors of the ICT industry, however, there is a shortage of experienced testers and test managers. By and large, this is because testing is seen as a less glamorous job than development. This is possibly because testers are not the "creators" of the product. However, it is more likely to be because the complexity of testing at its fullest potential (technically and managerially) is not well understood.

Testing is often treated as an entry-level position for new graduates or as a temporary position until a more desirable job as a developer or business analyst comes along. Many organisations will have at most one or two experienced testers, with the average level of testing experience on a test team being far lower than the corresponding development experience on the development team.

As testers gain in experience, they sometimes become frustrated by their organisation's lack of appreciation of their role and its difficulties. Many of them respond to this by becoming freelance contractors. This is widely seen in all parts of the ICT industry, with some sectors staffing test teams exclusively with experienced contractors.

This has also given rise to a growth in the number of specialist software-testing consultancies. Test consultants are experienced test managers who have a strong interest in test strategies and test process improvement.

In Australia, organisations providing these services range from larger development organisations and management consultancies that have broadened their scope into testing, down to an increasing number of small consultancies who specialise in test management and test processes.

Some tool vendors are also entering this arena to provide test process support to facilitate the introduction of their tool sets.

Fully outsourced developmental testing is still uncommon in Australia (as opposed to Europe, where it is very popular). Where testing is outsourced here, it is often to test labs specialising in specific types of testing, such as usability, compatibility and performance/load testing.

What could testing contribute?

Most projects understand that they need to test their software to help ensure that a quality product is delivered. What they do not always understand is how to apply testing efficiently and effectively.

The most controllable projects are the ones that define a test strategy using a variety of test levels at various times throughout the development life-cycle. They use testing (as well as other verification and validation techniques such as reviews and inspections), to reduce the likelihood of defects being put into the system in the first place, and to ensure that the defects that do get through are found and fixed as efficiently as possible. They do this through the use of a systematic end-to-end testing strategy:

- For every development activity, there is a corresponding verification activity;
- Product requirements are reviewed to ensure they are defined well enough to support the goals of stakeholders, including testers. This will reduce the incidence of defects related to functionality and fitness for purpose;
- Design-related defects and robustness issues are removed by developers through unit testing prior to hand-over of software to the testers for functional testing;
- Independent functional and system testing is done as early in the life-cycle as possible. The feedback from these activities is used to prioritise development activities and reduce the risk (or impact) of running out of time before the product is ready for release;
- The dependencies between teams and the responsibilities they have to each other are clearly understood.

When included in a project at the early planning stages and applied throughout the development life-cycle, software testing can be a powerful risk management tool. In addition to contributing to more predictable product quality, it can also provide visibility into the quality of the development process and progress against the schedule.

As projects become increasingly aware of the benefits of a good test process, software testing is finding its identity and taking its place as a respected partner in the development life-cycle. ■