GETTING READY FOR INDEPENDENT TESTING

One of the biggest problems facing independent software testers is that there never seems to be enough time to run all their tests.

Sometimes projects do not allocate enough time in the schedule to start with. Sometimes the development schedule slips. Sometimes the user requirements change, causing more work to be done. If these projects have fixed and immovable deadlines, the usual consequence is that the testing effort gets squeezed.

To maximise the amount of time spent testing, the project should plan to release working threads of operation to the testers as early and as often as possible. Testers should make sure they are ready to run tests as soon as this software is available from Development.

This means that the project should remove the obstacles that traditionally impede the start of testing and waste that all-too-short testing time.

1. Make sure the software is ready for testing

In order for testers to conduct effective functional and system testing of the product requirements, the system must be relatively stable.

Developers can ensure that they hand over a stable system to testers by doing unit testing on their code to find any design and code-level defects prior to hand-over. They should also make sure that the system is still robust following a round of bug fixes.

The focus for developers should not be just “software to test”, but rather “software able to be meaningfully tested”. If the testers find themselves unable to sensibly run their functional tests, they should consider passing the software back to the development team, to avoid wasting any further time.

2. Make sure you understand what to test (and not test)

Testers should be told which functionality is ready for testing in the release. They should also be told whether any areas of functionality are still unstable or require major work, so that they do not spend time trying to test functionality that the developers know is not ready.

This information should be identified in a Release Note (or equivalent), which is prepared by the development team and discussed with the test team. The Release Note should identify (at least):

- The released software (executable name, version number, date baselined, implemented functionality)
- Dependencies (associated files and libraries, compatible executables, associated tools)
- Constraints (limitations, incomplete functionality)
- Problem Report status (PRs fixed in the release, known PRs still outstanding)

3. Make sure the tests are ready to be run

It is important that testers coordinate their test preparation activities with the development team’s schedule for implementing the functionality. Testers should be given advance warning of which functionality is being developed and released in each software drop, so that they can prepare the relevant test procedures and test data in advance of receiving the software.

4. Make sure the test environment is ready

Setting up the test environment can be a very time-consuming activity, and often requires coordinated input (eg, effort, software, hardware, tools) from a variety of sources.

This activity, if left until the last minute, will consume the time that the testers could have used to run tests and provide feedback to the rest of the project. Projects should ensure that the test environment is established by the time test running is scheduled to start, at the very latest.

In conclusion:

To focus the software hand-over activities, identify a set of “test readiness” criteria in a checklist. A checklist will help to increase the visibility of the hand-over process and the responsibilities that project members have to the process.