"PROCESS" IS NOT A DIRTY WORD!

On many software development projects today, testers and developers are increasingly being caught between two imperatives – to participate in the development and delivery of high quality software, and to do so in the shortest possible time. Of these two imperatives, the deadline is the one that usually wins out.

Even projects that start out with the best of intentions tend to revert to anarchy as the deadline nears. They feel compelled to focus on the release date and hope that the quality looks after itself.

THE PROBLEM WITH "PROCESS"

Most testers and developers understand that a formalised method of working is the most efficient and effective way to run a project. Certainly when projects grow to be more than just a few people, everyone knows (deep down) that some sort of structure needs to be put in place to manage tasks, facilitate communication, and standardise the approach.

At the same time, however, many project members meet the prospect of a formalised process with fear and loathing. They see this structure as unnecessary bureaucracy that is designed to slow them down and burden them with paperwork. The process becomes the scapegoat in the perceived conflict between quality and efficiency.

Unfortunately, some of these perceptions are not without reason. Many of us have been on projects where development standards have been slavishly applied, with little or no opportunity for tailoring the required level of documentation and technical rigour to suit the nature of the application and proposed schedule.

A few bad experiences in the past are no excuse for going to the opposite extreme, however. If designed thoughtfully, a formalised process does not have to stifle your creativity, slow down your work, or burden you with unnecessary paperwork. In fact, it should have the opposite effect.

WHAT MAKES A GOOD PROCESS?

There are a number of factors that should be considered when designing a good process, including the nature of the application, the assessed level of technical risk, the development schedule, the size and complexity of the project team, the contractual requirements, and the maturity level of the organisation.

- *A good process relieves you of the need to keep reinventing basic tasks, freeing you to spend your efforts on the unique and creative aspects of your work.*

  For example, starting with a standard document template means that you do not have to establish what is required of a particular document each time one is produced. Furthermore, the organisation only needs to review and argue about its structure once, rather than every time one of these documents is written.

- *A good process only requires documentation that has a well-defined use.*

  When people do not understand why they are generating paperwork, it feels like a waste of time. Because of this, the task is often not done well, and so it actually *is* a waste of time. The key is to understand the purpose of each document you write, and the intended audience. Do not write a long document when a short one will suffice. Furthermore, if you cannot think of why someone would ever want to refer to a document or record again, then it probably is not necessary.

- *A good process encourages uniformity but provides sufficient flexibility to move outside the constraints when the circumstances dictate.*

  Do not allow your quality system to stifle creativity. Although a good process provides a predictable framework within which to work, it should permit moving outside of that framework when there are justifiable reasons for doing so.

INTRODUCING A PROCESS

Having designed a good process, do not undo your efforts by a heavy-handed introduction process. To introduce a process with a minimum of fuss, keep a few simple principles in mind:

- Introduce key process areas (eg, requirements management, documentation techniques, unit testing) one at a time.

- Implement each key process area in a basic form first, allowing for the refinement and complexity of the process to be added later, as the maturity level of the organisation improves.

- Achieve buy-in from project staff, maybe through the use of workshops.

CONCLUSION

With a good process, the conflict between quality and efficiency largely goes away. Rather than hindering your software development project, a well considered development and test process can be a powerful tool for improving the efficiency and effectiveness of the project. A simple, yet consistent framework, can streamline project tasks, reduce the variability of documents and other deliverables, and facilitate communication on the project. The visibility that a good process provides will contribute to more predictable product quality and a provide insight into project progress against the schedule. If done well, it does not have to slow you down.