



## 10 tips for successful testing!

by Dominic Maes and Steven Mertens

Software development is often a balanced exercise in time and quality management. For most companies it is a constant struggle to complete tasks on time and within budget, usually resulting in quality being neglected. When applications are finally released into production, users are often confronted with only basic functionalities or even the complete omission of functionalities due to the lack of development time.

How do you set up or improve your software testing process to reduce the business risk of poor application implementation?

Here are a few simple tips to get you ahead in quality, saving you time and money:

### 1. Start early and be prepared!

The key to a successful software development project is preparation. Projects often start as simple ideas with enthusiastic developers committing hours of effort prior to in-depth investigation, this results in unforeseen problems causing delays and project overrun. Testing should be introduced from the outset, this way you have an objective look at the project feasibility and its risks prior to committing valuable effort. In addition the test or quality department should be involved at an early stage to review requirements and design documentation. Reviewing helps to highlight defects and if conducted early in the process will save money by reducing the cost of expensive repairs or corrections. Defects found in the requirements phase are on average 100 times cheaper to correct than the same defects found in production. Preparing your test execution is paramount to getting your project completed in the shortest possible timescale. Valuable time can be saved during the test execution phase by specifying your test scripts and creating a test scenario before the application under test is delivered.

### 2. Select a testing method which fits your needs

Iterative development, e.g. V-Model, Agile, Extreme Development... Each Software development process needs its own adapted test process; select the testing method correct for your application. You and your management team will benefit from a process that not only reveals the business/project risks but provides the comfort of knowing that the testing team has mitigated these risks. Pay attention to your project specific characteristics, there is no such thing as a 'universal test approach'. Processes that work for one company or project may be totally inappropriate for another.

- Do you need to react quickly and make lots of changes?
- Is the goal of the project to add functionality to an existing application?

Many different aspects of your development process can influence the correct choice of the testing approach. No matter which direction you choose, ensure that your test process is structured and started at an early stage.

### 3. Be objective

The test team should have an autonomous role within the organization as it is often difficult to work within the project itself. Some project managers tend to neglect the test teams advice when their project deadline is at stake and quality is no longer important. When there is a delay in a development, many project leaders also have a tendency to use the time allocated for test execution to perform further development activities, hence testing is usually skipped and not just postponed. This can have serious consequences for the overall quality of the project which often ends-up in rework when the application is placed in production. The additional work results in delays and further cost to the overall project. When your organi-

zation is large enough (i.e. multiple IT projects per year) you can even consider setting up a test factory, a separate department conducting independent and uniform testing of all your existing applications and new projects.

### 4. Review, review, review, review ...

Testing should start early in the development process, you cannot afford to wait for code to start testing your application. Make sure you start reviewing your documents right from the outset; as well reviewed documents tend to result in 65% less defects being carried forward into later phases of the development cycle. This can amount to huge saving of time and money. Don't 'just send' your documents asking colleagues to review them without explanation or preparation, instead give individuals specific assignment on the documents that you want them to review. Have someone check the technical aspects, another, the documents on a higher level and yet another against the layout standards used within the company. This way you can avoid holding review meetings where only grammar mistakes have been highlighted. Reviewing is an exhausting task, so only reviews for short periods, one hour maximum, at a review rate of 10 pages for the period, make sure you are not disturbed. Avoid doing reviews in a crowded or noisy office environment, close your e-mail.... Make sure you become an expert in your reviewing methods. It takes time to learn 'how' to review and code in a professional and organized way.

### 5. Prioritize

It is generally accepted that testing everything is impossible. In a normal application or system, there are far too many functionalities and too many possibilities to test every single one of them. An application can take several man-years to test so it is important to test those

aspects of your application where the risk of damage or frequency of use is expected to be the highest. A thorough risk assessment should be performed before setting up the test strategy. Select test techniques according to the risks you want to reduce. Build a strategy based on the risks that are involved in the project. Remember: No Risk, No Test.

## 6. Measure!

Two important questions which should be posed.

- What do you know about your testing process?
- What do you want to know about your testing process?

Most of the time, it's easy to establish the cost to solve a defect in your application (at different stages), but what is the cost and how efficient is your team in finding these defects?

The key to answering these questions is: - start measuring.

Gather data to monitor your test process, efficiency and effectiveness of your test department.

Utilize metrics which provide the information you need to know about your process, i.e. test scenario density, defect density, number of defects found in the first month of production, etc...

The question is often asked 'When is the best time to start measuring?'

In our experience, NOW is a good time to establish metrics, the sooner you start collecting data the faster you can start improving.

## 7. Know when to stop testing.

When can you suspend test execution?

As a rule, you should stop 'Bug Hunting' when the cost of finding an additional defect is no longer in balance with the risk you're exposed to. Under normal circumstances, you need to set exit criteria to decide in advance when testing should stop, for example, define a criteria that will stop testing when 90% of the available test scripts has been executed with a maximum of 5 open defects. Make sure your project criteria are clearly defined. Discuss your exit criteria at the start of the test project and incorporate them within your test plan, this will avoid discussions at a later date. When no exit criteria are defined, testers tend to continue testing until all defects have been found or solved.

## 8. Preserve.

The strength of software testing partially lies in the structured preservation of testware. On completion of testing, take some time to evaluate and assess your deliverables and preserve them for future use. Test cases should be stored in such a way that they can be reused in later projects- or even during the maintenance

phase. You should always have your 'old' test cases ready for immediate re-use, this way you can save 30% to 50% in overall test effort.

## 9. Communicate

In many companies there is an 'It is not MY responsibility' attitude. Communication is poor between the end-users, architects, developers, testers and management. Too often this kind of behavior leads to misinterpretation in requirements, unwanted functionalities and unsatisfied end-users. Communication is key to 'quality'.

Listen to the end-users and get involvement of the complete team to obtain the best results.

Be diplomatic and have constructive discussions.

Software development is constantly balancing budget, quality and scope.

## 10. Last but not least: keep it simple!

Often companies become lost in administrative procedures when considering testing. An efficient test process should excel in its simplicity, no mountains of red tape, nor tedious ways of giving your quality team and developers a hard time. Choose an approach that fits your project or company and does the job: assuring quality for your applications.

Good luck!

# Biography



Dominic is a senior test consultant at TOBIUS NV. Dominic has a master degree in commercial and financial sciences. Since 1997 Dominic is working as a professional test consultant. He started from junior tester over test coordinator and test manager to an expert level, specialized in both test processes and tools. Dominic gained experience various industries and sectors, such as retail, telecom, manufacturing, finance and government in Belgium and the Netherlands. Dominic has excellent knowledge of TMap® and TMap Next®, and is both ISEB Practitioner and TMap® Certified. He is a requested speaker at colleges, universities and seminars.



Steven is Manager of the Competence Center Testing at TOBIUS NV. Steven has a bachelor in management and insurances. Since 1998 Steven gained experience in various industries such as the pharmaceutical industry, gas & electricity, social security, insurance, banking, retail, telecom, industry and government in Belgium, the Netherlands and Luxembourg. He has an excellent knowledge of testing methodologies such as TMap®, STBox™ and Test Process Improvement frameworks. Steven is one of the three creators of STBox™. He has experience in coaching people, giving training, test management and team leadership, determining test strategies, writing test plans and practical implementation of test methodologies. Steven is ISEB Practitioner Certified since 2005. He is a frequent speaker at colleges, universities and seminars.