

- Rob van Steenbergen



# Execution & Reporting

## Best Practices of Software Testing

*Continued from the last edition of Testing Circus...*

After years of experience in the testing profession I have heard a lot about the 'best practices' for software testing. Sometimes people just want to share their best ideas, so you can use them too. Some stories I've overheard and noticed:

### **Test execution**

If you are lucky you can put other testers to work to execute your detailed test scripts. In this way the bugs will come to you automatically. Especially with a first version of a product, there is plenty to be found. Execute testing yourself on the basis of your test scripts is also good, because you don't really have to think anymore and this becomes a relaxed part of your job. This is because you've already done the hard work in writing the test scripts earlier. Now it is just a matter of executing the scripts one by one and record the bugs.

### **Test Reporting**

Reporting is the best part of our testing profession. This is because you have the power to reject a product. Use a beautiful picture on the front page of a traffic light that is in the colour red to emphasize that there is a no-go decision in the report.

Report on bugs that are still open, who is working on them and call these bugs product risks. Note: this is actually your product risk analysis. Every test method

tries at the beginning to capture product risks. But the real risks are the bugs that have not yet been resolved.

Add on some statistical data on the number of test cases executed what you already have done and how many of them are failed. Emphasize that a retest is required. If this is repeated often enough in reports you can ask for more budget and time. This because you've taken in account only one retest in you master test plan. Please do add this in your test plan if you haven't already.

### **In production**

Eventually you will be overruled by management, despite all the bugs that are still open. Prepare well for this by remembering a number of sentences that you can use in meetings: "This is going to cost a thousand times more than if we would solve these bugs now" or: "Will you be prepared to get in the news or other media with these bugs still in it?", "We still need to have at least one extra build to solve the worst showstoppers ". And the best one: "That one could have been solved a month ago".

If after this you are really overruled and the product is put into production, say to people repeatedly that this was not your idea and that management has taken the decision, even though you have rejected the product". This will build your reputation and makes your position stronger in your organization. People will come to trust you more and you will have good credibility.

## Conclusion

These are best practices that are very powerful to make sure that everyone keeps focussed on what is most important: solving bugs and ensure that everyone listens to the testers. Often the bugs that are found in production by users are not really big issues, because project managers almost always decide to put extra people in the helpdesk for the first weeks. The help desk will solve the problems with workarounds, or just keep people happy by saying we are working on it. So people will wait for the new version most of the time. If there is a bug in production that you have not discovered as a tester, mostly this is not directly related to testing, because you didn't create the bugs. (The developers did of course)

Eventually by using these best practices you contribute to a successful project. The testing profession may seem dull and boring, especially if you have to wait for specifications and repeat the same test scripts over and over again, but the satisfaction of a product finished is what makes this all rewarding.

### What did you just read?

Did you recognize parts of the stories above? Could you agree with them or even some parts of them? Please not that those parts will be the areas where you might want to study some more on testing. Maybe a colleague is telling some kind of story as you've read in this article. Run away or help him or her, because this kind of stuff is really killing our professionalism.

The message is: Keep critical of your own work and professionalism. Of course, everyone has a bad day or a project that is not going well. The main concern of the tester is to stay sharp and deliver valuable information about the quality to the stakeholders.



**Rob van Steenbergen** is an independent software test consultant from the Netherlands. He has over 15 years of experience in software testing with several organizations. Currently Rob is

working for ThiemeMeulenhoff, an educational book publisher. The domains he has experience in are banking, software houses, government, embedded software, infrastructure and public transport. Besides projects with clients, he is involved with a Dutch test magazine as an editor, runs a website about test events ([www.testevents.com](http://www.testevents.com)), has his own blog about testing and writes articles for national and international magazines, such as the Testing Circus. Website – [www.chickenwings.nl](http://www.chickenwings.nl) - Rob tweets at @rvansteenbergen