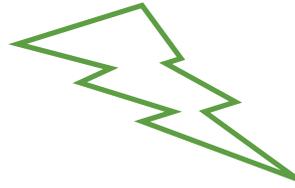


Promoting Software Testing in Your Organization



Testing Circus Regular Feature

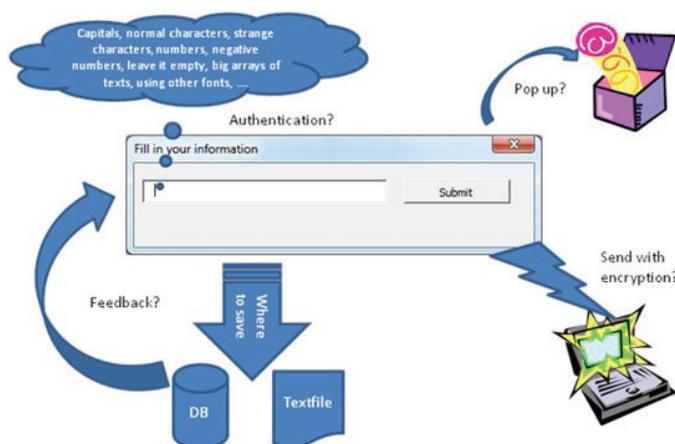
- Rob van Steenbergen



How to promote Testing in Your Organization

Test promotion: Do regular product risk analyses

As a tester you should know it. You can't test everything. Even a simple software program with one edit box can be tested and looked at from many perspectives. Capitals, normal characters, strange characters, numbers, negative numbers, leave it empty, big arrays of texts, using other fonts, Just think about it and you will have a big checklist when you're through. What should happen then, if you push the [Submit] button? Is the data from the edit box placed in a database or a text file? How does the database look? Should the data be saved secure or is a text file enough? If it is saved in a text file, how to deal with strange characters. Or is that of no importance to the user? Is it maybe more important that a textbox pops up with the text in it? Or should it be send to another device via mail or another communication protocol. And if it is sent to another device, what about security of the communication channel. Encryption? Is that important? Should we get feedback if something is sent?



If you don't know what a product should do, you don't know what to test. More important, if you don't know what's important for the user, on what part of the software should you focus? Security of saving or sending data, or should you focus on functionality and usability of the software?

Finding out the purpose of the software

If you don't know what to test, then you could try to test everything that comes up in your mind. But are you then efficient and effective in your testing if you do this? I would say NO is the answer. If you want to do testing in an efficient way, you would want to know the purpose of the software that you are testing. By knowing that and discuss this with the people who are going to use this software you get an idea on which areas to focus. For example: after talking through the functionality with the right people that sending of data is not important and feedback neither, than you can focus on saving the data and the database integrity and security. Even then you will have a lot to test. This still doesn't fire you of checking all kinds of input in the edit box.

Setting the scope

Even then. What happens with the data that is saved? How is it read and edited, which software does that, can it use strange characters? Is that even important? Does it matter that the user enters strange or weird texts? Or does the user already know he or she should only enter normal characters (a-z)? Is it even important to focus on that aspect of data entry?

Eventually you will have some test ideas for this simple input box, and an output screen, in combination with a database.

- Input of data in miscellaneous formats
- Check the data in the database.
- Check reading out of the database
- Login possibilities for security reasons.
- Speed of the database and how it handles big loads
- Check encryption of data in database

If you work out this test ideas and discuss them with designers, users, developers, support employees, and other stakeholders of the software you will get a lot of test cases, or charters for exploratory testing that will focus on the real purpose of the software.

That could still take a lot of your time. Maybe it is difficult to explain to your manager why you are working for a few weeks on an input box. "What? Three weeks of testing for only an input box? Why?" Explaining that could be very difficult. Because why should we test data encryption in the database, if the users have to log into the application with a username and password? That would maybe be enough security. "Why performance testing?" Manager: "That is not important, also it is not important what kind of characters the user will put into the edit box. I want to see your test report at the end of this week."



Is this it?

As I started this article, the things I've written down look already a bit like a risk analysis to get you focused on the purpose of the software,

but this is only purpose. What should you test within this scope you found? For that you should do a product risk analysis. To get focus on only the parts that probably have a big risk in failing or causing a lot of damage when

it gets in production. This means talking to the right persons to discuss the risks and the priority of those risks. No damage or low damage when something goes wrong? Little or no testing. Things could go wrong, but only once a year? But has big consequences and thus big damage. Lots of testing there! But as a tester you cannot always assume it yourself.

So doing a risk analysis is getting focus on the parts that matter for the people that will use the software or otherwise are involved in this software (the stakeholders).



Why is it also a test promotion tip?

Doing a risk analysis is not sitting behind your desk and thinking of risks on your own. It is about going into the organization and communication with the stakeholders for the software you are going to test. This is where lots of people will be introduced to what testing is. This will give people more insight and even control over testing. You could help them in this way, getting more understanding why you do test and why it is needed.

This is a big subject within testing. I realized that when I started writing this article. That's why I will write more articles about this in the Testing Circus magazine that will help you, the tester that is starting with product risks analysis or wants to start with this. I don't know if it I will write two or more articles, we will see ☺.

As for now an assignment for you.

- Who is using your software? Can you identify groups of people that will use the software?
- Who is paying for building the software and will lose money when there are problems?
- Who is going to profit finally when this software is going into production?
- Who is responsible for a good technical or functional design?
- Who is going to develop this system?
- Who is bringing this software into production
- Who will do maintenance on this software during the software lifecycle?
- Who will do support on this software?

Think about the persons you need for a product risk analysis in your project.

To be continued, thanks for reading, please don't hesitate to mail me with questions and suggestions.



Rob van Steenbergen is an independent software test consultant from The Netherlands.

In the last 5 years he has been involved in infrastructure projects and is now working at Leaseweb. He is working in agile projects, infrastructural and software products, coaching new testers and helping with setting up and improving the testprocesses for this company.

For more information visit – www.chickenwings.nl

Rob can be reached at
<http://twitter.com/rvansteenbergen>
 E-mail: rob@chickenwings.nl

Do you love Testing Circus ?

Show it to the World.

Show your love for Testing Circus. Send your testing team's photo to us. We will publish the photos here.

**We Love Testing Circus.
It Rocks!**

How to get your photo published?

1. Take printout in a paper where "We love Testing Circus. It rocks!" should be written. You can use hand written poster too.
2. Ask the team members to hold the papers while clicking the photo, so that the same is visible clearly in the photo.
3. Take 2-3 snaps of your team together and send to editor@testingcircus.com with Team Name, Organization and Location.