HOW TO IMPROVE YOUR TESTING PROCESS

"There is always room for improvement", this is true to most aspects of your work and definitely to the way you manage your testing process.

The problem is that improving means changing the things we do, and most of us are not eager to modify what we are already used to doing.

Having said that, there are always small things and gradual changes that will help us improve our work in a way that is botheasier to do and with smaller risks of failing.

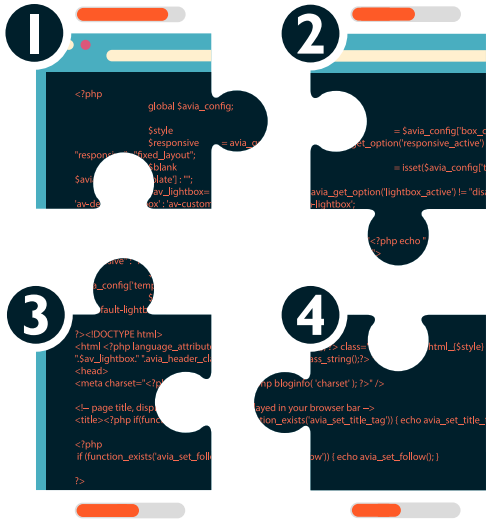
The Following is a list of ideas you should consider that will help you improve the way you manage your testing

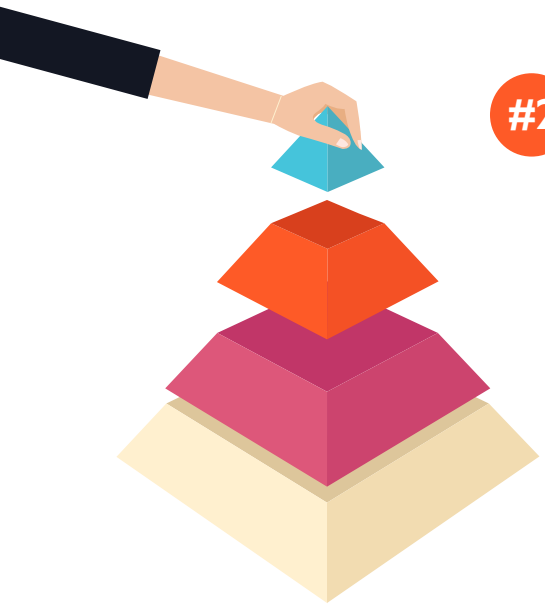
#1 DIVIDE AND CONQUER

Break your system down into features, modules or some type of smaller pieces that will allow you to approach the big project as a set of smaller testing units.

To do this you can use the topology of the system. If you are working Agile then User Stories and Epics will help you.

Many times the pieces won't be uniform or homogeneous, this is not a problem but try not having features that take half or 1/3 of the version under test.





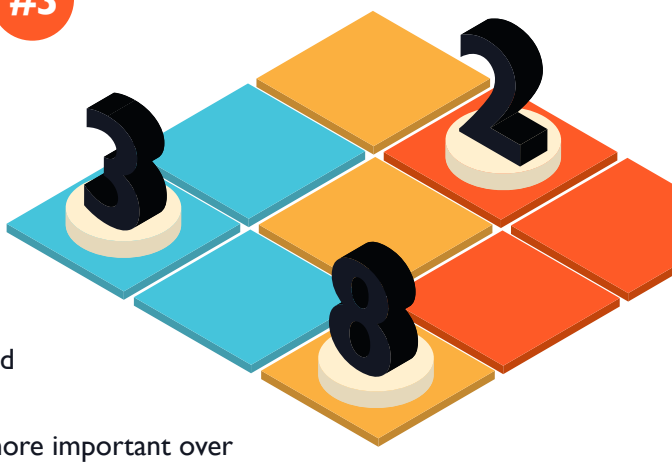
#2 UNDERSTAND THE RELATIONSHIPS AND DEPENDENCIES BETWEEN TESTING UNITS

Dependencies can be logical such as "you can not use A if you don't have B", or technical for example, "in order to test the GUI you need to have a back end ready and working".

Sometimes these dependencies can be "worked-around", for example if you can test the GUI without a backend given that someone provides with a proper a stub, but it won't be exactly the real thing and you will need to test some of it again later.

Many times the plan of the development will not overlap with your plan, but then you should not take development as the only one defining what needs to be delivered and when. It is enough that you say that you won't have time to test it and report bugs on time and you will see how the constraints of the developers start shifting.

MAP THE RISKS OF YOUR PRODUCT AND GRADE THEM BASED ON A SCALE OF HIGH, MED AND LOW #3



Risks can be technological, for example if whenever there are changes on a certain part of your system you end up having tons of bugs all over the place.

Risks can also be project-wise, for example if the developer assigned to a task is overloaded and she won't have time to finish on time all the bug fixes.

Risks are always subjective and so you need to see most of the risks together in front of you and only then will you be able to arrange and classify them.

Keep in mind that risks will change, they will become irrelevant or more important over time. Also new risks will constantly arise during the life of your project. For this reason risks need to be reviewed once in a while.

#4 USE THE DEPENDENCIES AND RISKS TO CONSTANTLY ADAPT YOUR PLANNING

Use all the inputs available (risks, specs, history of bugs, etc) to plan how you want to test the system. Speak up when the plan of the development is structured in a way that will not allow you to complete your work in a logical way.

Assuming you know what needs to be tested more and what will take more time to solve, plan your testing strategy accordingly.

Remember that as risks and your product change, you will need to rethink and modify how to approach your testing project, and so you should check your test plan and how it progresses over time, also how you need to adapt it in order to cope with the new reality and the challenges in your project.

#5 SET UP SMART SCHEDULING

The information you provide as part of your tests has an objective: to allow the project to be delivered on time and with the desired functionality. When all works fine and all your test pass (not finding issues) then all is good. But when you find issues, these will need to be solved by development and retested properly.

Work with development to understand what types of issues will take more time to solve and will be more risky to test and make sure to test these areas sooner rather than later.

When planning your testing approach and testing schedule, to take into account when you need to provide the information (feedback) from your testing so that it will still be on time to be valuable.

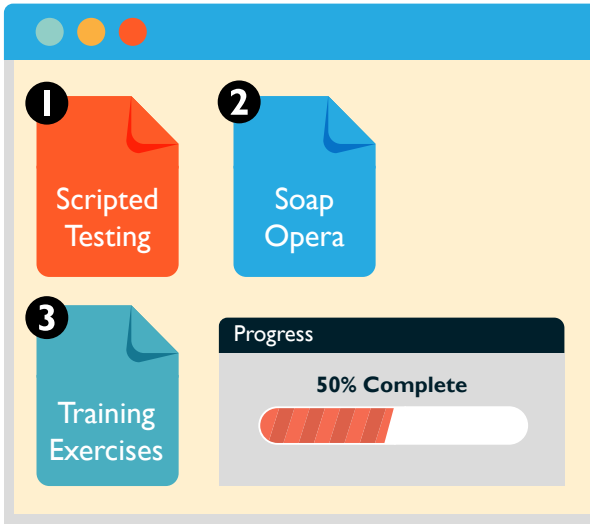
#6 MAKE SURE YOU'RE COVERED FROM ALL SIDES

When planning your testing process, remember to combine different testing types to cover each testing unit.

When you combine different techniques and approaches you increase the chance of finding more bugs. Use different approaches such as exploratory and scripted testing, or training exercises and Soap Opera scenarios.

Make sure that each approach is taken independently of the results of the previous one to ensure a better chance of finding most issues and not "falling" on the trap of incorrect assumptions.

Two sets of eyes always find more things that one alone, and no two heads think alike. For these reasons you should strive to have more than one person looking at your application. Pair testing and peer reviews are alternatives that may come in handy when you don't have the time or the resources of for two testers on each feature having to testers.



#7 DON'T REPEAT YOUR MISTAKES

Create a list of the last 10 or 20 mistakes you or your team did in the last releases and review your test plan and process against this list to make sure you will not re-incur on these same mistakes.

Since our memories are imperfect and we tend to forget bad things that happened to us in the past, make a list and share it with people outside the project so that they use it in their projects as well.

ADJUST AS YOU GO #8

Schedule time periodically (every week or every two weeks) to review the assumptions of your project together with your team and to make the necessary adjustments based on the feedback you gathered throughout the process.

Things will shift, requirements will change, and things will fall on you and your project from unexpected places.

If everything in your project changes, then how can your test plan remain static? Make adjustments when needed to ensure you are not testing the wrong things in your project. When you reach the decision that you will need more help or time to cope with the new reality it will be better to tell this as soon as possible.

GET AN OUTSIDERS' POINT OF VIEW #9

Be proactive in sharing with people outside the test team as much as you can from your testing process. Explain your assumptions and operations and try to gather feedback based on their knowledge and experience.

Many testers and leads don't like sharing information and try to make testing some sort of voodoo magic. This is both wrong and unwise, as you will also miss the possibility of getting feedback into your process and also of having people being aware of the hard work of your team.

By sharing as much information as possible with other people in the project you will help them to update you when anything changes that may affect your plans.

#10 DON'T IGNORE YOUR MISTAKES

Don't be afraid to accept that you make mistakes and can correct them. Most times you will be the first person to catch your mistakes, but many times we tend to ignore them in the hope that they will go away - and they seldom go away by themselves.

ONE FOR THE ROAD...

The dynamics of the testing process means that you will have to repeat these steps to keep up performances. However once best practices are in place it becomes part of the company's nature and thus less of an effort to implement.

DID YOU LIKE IT?
Why not share with others?

PractiTest

http://www.practitest.com/