

Role of the test function when outsourcing software development

By Shafiq Ahmed



Introduction

From experience companies that outsource their software development to an external vendor, I have seen the test function only used towards the end of a project. When this happens, testers end up finding issues with requirements, software and the design. Due to cumulative errors building at each stage, going back to fix issues require considerable resources and time. You can reduce the errors by involving the test function at each stage starting at the contract stage. It's easier and cheaper.

The focus of this article is to outline how your organisation should use the test function when outsourcing development work to an external company.

Assumptions

I will be making the following assumptions:

- Due diligence has taken place.
- You have access to professional test engineers.
- Wherever the test function/team is mentioned, it could apply to the vendor or your business.
- I will be using example with one vendor. You might have multiple vendors that will be supplying you with system components, software, etc.
- I have split the stages into clearly defined activities,

however, the real world is not as simple as there will be overlapping and different naming conventions used.

- My focus is on the test function's activities, but other functions may need to be included in the relevant stages.
- Test function could mean 1 test person or more.
- Feasibility study has been carried out.

Problem statement and contract stage

Involving the test function at the problem statement stage allows the test team member(s) to ask questions about any ambiguities in the statement. Test engineers can pick holes where there is more than one interpretation of the problem statement. It also helps the test function to understand the context of the problem and build knowledge slowly allowing more time to think about how they will approach the testing at later stages.

Deliverables from the vendor need to be identified such as requirement specification, architecture design, test plan, test documents, test report, code, release notes, etc.

A mechanism for issue reporting should be implemented to allow you to report issues against documentation and software to the vendor.

Requirements stage

There are two categories of requirements; they are explicit and implicit.

Explicit requirements are easy to measure and can be written down. Implicit requirements are not written

down and cover usability, fitness for purpose, etc.

The vendor can meet all the requirements and still deliver a product that is not fit for purpose. Utilise the test engineers critical thinking ability and their technical skills to pick out ambiguities in requirements. They will also be able to help you define the acceptance criteria to deal with implicit requirements.

Note that the requirements are subject to change at later stages once new information is available at the later stages.

Design stage

The design stage could be user interface diagrams, user stories, architecture diagram, functional specification, etc.

Using test engineers to review and provide input at the design stages helps test engineers identify risk areas for testing. Areas of the design that are not clear to the test function can be clarified.



The test engineers should identify if there is a need for test automation and any special software hooks to allow test engineers to test the software.

Reviewing design documents can help testers build a better understanding of the software they will be testing and come up with better test cases/scenarios/charters.

Requirements can be updated if there is a change identified in the design documentation.

Test and development planning stages

Appropriate development and test methodologies need to be selected, with the input from all disciplines including test.



The vendor should agree to the test plan/strategy and identify who is going to be doing the testing and at what points. Test and development planning need to be taken into account together, otherwise, you'll end up with a disjointed and confusing plan with development and test functions not talking to each other and doing their own thing.

If it's decided your organisation will carry out the testing, it will be beneficial to get the vendor to do some testing, such as unit testing, functional testing and sanity checking before releasing the software to your test team. You will avoid wasting your time as it is quicker and cheaper to fix errors because the bug fix cycle is tight and lean.

If specialist testing is needed, such as penetration testing or certification to meet regulatory requirements, they will need to be identified along with equipment needed.

The acceptance criteria shall need to

be described including requirements verification testing and validation testing.

Other in-between stages

I will leave out describing the test document creation, testing and development because there will be different according to the methodology used and will complicate the picture. The code shall need to be tested by developers and the test engineers during the development phase. Regular monitoring should be in place and issues found will need to be reported to agreed reporting mechanism.

The final stage

To ensure the project is complete check the deliverables are delivered, the requirements verified and the validation testing successful.

You will need to ensure you are happy with the bug fixes and any outstanding non-priority bugs.



Additional points to consider

- If the work involves integrating multiple vendor systems or software, it's likely to increase testing time because there are a lot of failure points and the possibility of incompatible code.

- Avoid a situation where the vendor "throws code over the wall" for your test function to test. Ensure they are doing some of the testing themselves before releasing the software to you.
- If you are going to be using existing solutions tailored for your business, such as using existing software libraries or a technology platform for developing an embedded device, for example, a smart speaker, it might be worth asking for evidence of testing and also get your team to test the features as part of evaluating the solution.
- You will need to provide quick feedback to the vendor when there are issues that you need to report to the vendor. What is likely to happen is that key people on your project at the vendor might leave or get pulled on to other projects. A delay also means that if the same person comes back to work on your code, it will take them a bit longer for them to get into the frame of mind.
- When choosing the vendor to work with you, ensure they understand the importance of testing.
- When dealing with test matters with the vendor make sure that you are dealing with a test professional.
- Make sure that the test terminology used is agreed on, for example, regression testing, smoke testing, exploratory testing, ad-hoc testing mean different things to people in other organisations.
- Before signing the contract,

ensure that the test engineers review the test section of the agreement.



To contact me visit:
www.shaf.com/contact

© Shafiq Ahmed 2019
www.shaf.com