

My learning as Testing Professional from my peer group, organization, and industry

Manish Singhal
Solidcore Inc
manish_singhal@solidcore.com

Abstract:

Being in the industry for over 11 years, this topic excites me because important factor is to look back and observe what I have learnt over past 10 yrs.

Being worked on domains like ERP, CRM, Desktop publication, Telecommunication, e-learning, financial compliance and finally, security and chance control, worked as developer and then in testing group both in project and product based companies, there is so much I have learned. Success and failures were part of life.

It is so important that we learn from peer managers. Any company whether a project or product brings new challenges with time. With IT industry going again into recession, India being no longer an IT outsourcing advantage due to high cost, rising inflation and reducing dollar price, achieving success is getting tougher & tougher.

On this paper, I mainly want to talk about few of the learning's which I have experienced in the past 11 years

Keywords:

Testing, Attrition, Risk, Mitigation, Requirements, Trainee, Communication, Test Environment

Introduction:

Learning'sAhaa... We as human being should keep learning new things everyday. Whether it is at our home or it may be office. It can be even on the way driving back home or listening to radio. It might be our peer or manager or may be trainee. We see so many things, observe it. Sometimes, we analyze, and most of the times, we do not. Sometimes, they might be useful immediately or sometimes it might give results in long term.

Here, I have taken very few highlights of some important topics which we come across in daily like. Some of them may be direct result of my real time experiences while some are just observation. Most of this can be applied in our day-to-day life as well.

Communication:

Let's start with one of the most important aspects of the testing success is effective Communication. At all stages of life cycle, we face numerous issues which involves discussion between relevant stakeholders. Right from the Requirement gathering stage till the final packaging where the final customer support is being done, the activities which are done invite attention from the management in the form of discussion and approval.

I will take an example of performance testing results. Let's say for a particular release, the testing team has done performance testing on the modules. There were no benchmarks which were suggested by the Product Management or in the requirements. QA team did the setup as suggested in the deployment guide and sent the results to Dev Team and Product Management. Since there was no requirements, so Dev team also was happy with the results. Product management, since were busy, just ignored the mail. Release was done and was made live at the customers site where the bottleneck's related to performance issues were found. Whose fault now? QA or Dev or PM. In fact this is no one's mistake or it is mistake of all OR it is failure of **Communication**.

Any activity which is done as part of release testing, whether it is Test Strategy Creation, Test Case / Scenario creation , Test Setup or Test Results, all should be baseline.

For the testing results, it should be the ownership of all the stake holders to ensure that all the delivers are identified in the strategy document it and they should be tied with milestones.

In the above example, for a 6 week release cycle, performance testing can be scheduled for 4th week and hence, in the weekly checkpoint meeting where all stakeholders are to be part, should discuss the results, agree on the them and baseline. This will have dual benefit of documenting the output in product documentation and also, will enforce on:

- ✚ Effective communication between al stakeholders
- ✚ Importance of Weekly checkpoint meeting
- ✚ Effective project plan where deliverables are tied with milestones

Risk Management:

Risk Management is also a part of effective communication. It involves:

- ✚ Risk Identification
- ✚ Risk Assessment
- ✚ Risk Mitigation
- ✚ **Risk Monitoring**

Few people might think that Risks are to be identified at the very beginning of any release cycle or any project. Hence, in the Process term, Risks are documented in Project Management Plan or Risk Management Worksheet, discuss it with stakeholders, identify the mitigation activities, plan for them and then forget.

This can be biggest mistake because risks sometimes can be like changing requirements. The Severity / Impact of any risk can change every week.

e.g. Non-availability of hardware required for testing. At the initial stages, risk for this can be less as the project is of 4 months duration and testing is going to start only at 9th week. Hence, the impact of this risk at the beginning is less. As the project progresses, then somewhere near the 4th week, impact of this risk gets bigger and if not solved, then, in the 7th week, it will be blocker for testing as the setup can not be created.

As mitigation, planning might be done initially that by 3rd or 4th week, hardware will be procured and will be made available to QA team. Hence, mitigation is done. But, planning mitigation has to be followed with Risk Monitoring.

Monitoring of risk in the above case would ensure that in the 4th week, red flag will be raised and procurement will be done on highest priority.

Similar example can be related to man power resources. At the initiation, planned resources might include a resource that will be made available on x week. As part of monitoring, Project Manager has to be in touch with HR / Project office to ensure that the resource will be made available on the requested date or else, some other mitigation has to be thought on. Not monitoring it on scheduled basis can lead later to compromise in quality which can be either in terms of features, testing time.

Hence, Risk monitoring can be the most important activity in the Risk Management life cycle.

Test Environment Setup

Most important among all the testing activities is setup of an efficient test environment. Why I am using the efficient word here?

Take an example. You have a product which is supported on multiple Operating systems and multiple databases. It is not only multiple OS, but it is different versions of the OS as well.

Very simple setup will be created where for each OS, we will have one machine and for each version, we can have a machine i.e. if testing has to be done for all windows platforms, we will have 1 machine each for Windows Me, Windows XP, Windows 2k3, Windows NT, Windows Vista and windows 2k8.

Similarly, just imagine the number of machines required for Linux, HPUNIX combinations if the product is supported on UNIX platforms? And not only have this, having these many machines brought challenges of space and on top of all this, power requirements. Big companies can manage the space problem but power?

Outsourcing was mainly based on man power availability at COMPETITIVE PRICING. If hardware cost goes up, the competitive pricing model will be broken which will have direct impact on the profit margins and hence, lost opportunity.

Solution: Efficient Test Environment: Use of Virtual machines, PXE Servers. Use VMware resources so that 1 high end machine with big RAM can serve as multiple machines. This setup can be easily done for Windows and Linux platforms. Use of virtual machines instances per real machine can be dependent on test type.

Key fact to remember here is that since VMware works on sharing resources, Performance testing should never be done on VMware environment.

Another solution is use of PXE server. PXE stands for Preboot Execution Environment. A PXE install server allows client computers to boot and install a Linux distribution over the network, without the need of burning Linux iso images onto a CD/DVD, boot floppy images, etc. This is handy if client computers don't have CD or floppy drives, or if we want to set up multiple computers at the same time (e.g. in a large enterprise)

These setups drastically improve upon the hardware need and most important, running regression tests over the weekend is an extra benefit which all points towards efficient test setups and testing ☺

Attrition:

One of the most important problems faced by managers is Attrition. Even though, there is no time when an employee will leave the organization, it can be tied with annual appraisals☺.

Attrition causes major damage when a critical resource decides to leave the organization and his / her last day is in mid of the release. Of course, majority of the companies have made the notice period as 2 / 3 months and have started enforcing it strictly, there has to be some end day for the outgoing employee.

While attrition is related to many issues, here, the agenda is not to stop attrition, but to mitigate the risk of attrition so that it does not affect the project timelines. The bottleneck for this will be the fact that it is difficult to judge as when an employee will resign.

Planning of Knowledge transfer sessions is a curative measure. Once any employee resigns, we either introduce a new engineer into the team from some other project OR we assign the tasks to any other team member who is from same team who will have some knowledge about the feature / product. While this is always done, as a manager, we should plan for preventive measures as well.

My approach is to have a Primary and Secondary owner for all features. The responsibility of primary and secondary owners is clearly defined and it is to ensure that across the multiple builds, multiple people have tested the feature.

Benefits:

- ✚ Dependency on any one resource is reduced because of unplanned leave
- ✚ Fresh perspective on any feature. It is always better to get a birds third eye ☺

In the case primary person departs, we already have a ready made resource available to take care of the feature.

Other solution is to have “Shadow Resource” available. The shadow resource is based on concept that whenever there is any unplanned manpower resource outage, if required, shadow resource can take care of the activity. This is not a solution for attrition as shadow resource will be shadowing multiple main resources at one time. They will not be master of our particular module but will be good to perform adhoc tasks.

While these measure will not completely eliminate the risks, but will in reducing the impact of attrition on release cycle.

Hiring Trainee's

Should we work with trainees? A topic, which is always brought on, always discussed among peer managers, discussion with senior management as well and then, put the idea into cold storage for re-visiting again ☺

Why just discuss and put into cold storage? Before we find the answer to this, main question is, why trainees? To evaluate, let us see pros and cons of this:

Pro:

- Low cost resources
- Willing to learn
- Fresh idea's, have tendency to look from a new eye
- Easy to find

Cons:

- Susceptible for attrition
- Training time can be longer

So, you keep weighing pros and cons and finally, if not able to draw a thick line, decide to go on with normal hiring and continue, hence, drop idea into a cold storage.

Here is a model we have derived to use trainee for Efficient testing. So, why I have decided to use trainee knowing that there are cons?

- ✚ Repetitive tasks: In any product based testing, the tasks become repetitive. You can not change the type of work / domain on monthly or quarterly basis. It is mainly regression testing, bug verification. There can be few feature additions but they are not enough to motivate the people. So, let's think about trainees. They can be given repetitive tasks like running same tests on different platforms, running automation and performance tests and since they are new, for them everything is learning ☺
- ✚ Will the quality of the trainees will be good? Well, this is on our hands. Market has good pool of people. It is we who decide the cut off marks, written tests and the interview.
- ✚ Training: We do not have to create separate training plan which is of 1 week or 1 month duration. Since, they have everything to learn, the best solution is on-job training and along with that, sessions can be organized on important features as the tests required. The key # to remember here is their work has to be closely monitored and should be given proper direction. So, Leads have to pay little more attention.

- ✚ Attrition: Trainee's are most susceptible to change frequent jobs. Since the initial salary will not be very attractive and giving them more salary will reduce their low cost advantage. The solution :
- ✚ Hire good trainees so that later they can be absorbed in permanent positions
- ✚ Show them their growth and career path in the organization, appraise them and give them regular feedback. Prove to them that even if there is little less money, the work in the current organization is good for career growth and if someone is exceptional, then bend the process to reward 😊

Conclusion:

I had so many thoughts in my mind. So, many things to share. After all, this is the way we learn right? From our daily activities, we see passing things and take them to improve our life (In fact, make things easy for all of us). This will be a never ending paper if all of us try adding experiences into to. But then, there is an end somewhere, so let it be now for my paper. Hope, it will help someone getting happy 😊