Software Quality Assurance

Test Strategies

An Effective Handbook for Implementing Software Test Strategies

By Shiva Kumar.

For additional information or assistance please contact Shivakumar at (+91)422-320-606. **www.thinkbn.com** • 697A, Trichy Road, Coimbatore, TN 641045.

Table of Contents

An Effective Handbook for Implementing Test Strategies

Software Quality Assurance

Introduction

Strategies are simple but executing them is not!

A strategy in its synonym is a preparation for long-term battle plans or making plans to achieve a goal. It might involve giving up a destructive habit or addiction or it might be a matter of recognizing a self-defeating pattern of behavior and somehow seeing how to change it.

For example the bulk of testing hours of a tester goes in to testing the product again and again for different releases in the same pattern. At a certain phase it leads to frustration of the testers for doing things repeatedly. It happens sometimes that when someone is frustrated he may sound something like, “I know its possible to see this (or experience) differently”. And often, this awareness or even hope that there is another way of experiencing your dilemma, or problem opens the door for it to occur. This is where strategies come in.

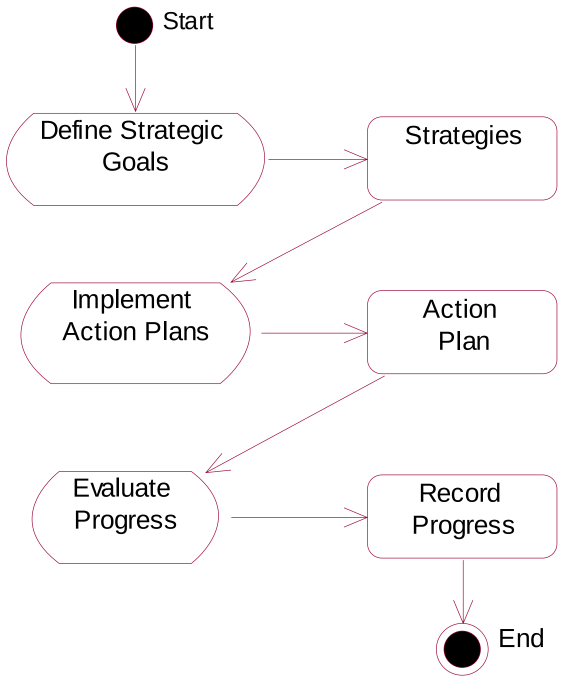
Test Strategy

The Concept

Strategic Planning is a process to guide the members to envision the future and develop the necessary procedures and operations to achieve that future.

The purpose of Test strategies to:

- Provide a framework and a focus for improvement efforts
- Provide a means for assessing progress.



This strategic approach to testing will forecast the action plans which includes the different types of testing that would be followed in the testing life cycle, identifying risk issues etc. earlier so that progress can be evaluated more precisely. The development of a test strategy is a means of communication with the customer on the organization of testing and the strategic choices that go with it.

The test strategy indicates how testing is to be carried out. This will clearly indicate where special emphasis on various aspects of the system has to be given so that best possible use of resource and time can be made use of. And also the test strategy forms an important basis for a structured approach to testing which makes the testing process manageable.

Why do we need a Test Strategy?

The Problem

When test cases are written for testing the whole of the product, which is intended to unearth all defects, why do we need a test strategy? These test cases will say what will be tested and a test strategy will say how this will be tested. Hence a test strategy aims at finding the most important errors at any early stage involving minimum costs. Consider a project scenario which is under tight schedule and where there are no test strategies, the project when comes to testing life cycle will undergo repeated prototypic testing to adhere to release dates and schedule. At the end of the project

1. Not all defects are unearthed
2. Not enough quality in testing, as the prime focus would be in sticking to the schedule
3. Frustration amongst testers for doing things repeatedly
4. No room for improvement

The Solution

As an alternative now consider a scenario where the testing team has identified a test strategy, which addresses most of the risk and the different types of testing that has to be subjected to the product are identified. At the end of the project

1. A organized approach would have been established in testing
2. The most important problems will be found earlier
3. The problem that requires much amount of rework will be identified
4. Efficient use of resources are made
5. Involvement in testing would be more since different types of testing would be followed in different phases.
6. Testing will no more be a monotonous job thereby eliminating frustration

Analysis of this type of testing would give a deep insight of unusual test approach which is risk based. This risk based test strategy will be fruitful, if all the risks are identified at an early stage. This means that assessing the damages due to the defects, which are left undetected prior to operation and during operation. Consider the following,

```
Risk = Chance of Failure X Damage
```

So when the chance of failure is high and when the damage is more, risk is definitely high. The chance of failure can be attributed to frequency of use and the chance of error. Broadly classifying into their respective areas would give a good perception of all possible chances of failure

Frequency of Use

The frequency of use determines the amount of risk because if a function is used many number of times a day has a bigger chance of failure than with a function which is used occasionally.

Chance of Error

The chance of error can be due to

- Complex functions involved
- Completely new functions, which the testing team is not aware of.
- Functions for which certain tools or techniques were employed for the first time
- Developers who lack domain expertise
- Insufficient Quality Assurance

These are some factors not all, which determine the chance of an error. Naturally this amounts to the high chance of failure, which thereby involves high risk.

The Damage

Lets see what will be the damage when the error manifests itself. The damage will be the

- Cost of rework,
- Forgone income,
- Loss of customer confidence

Avoiding the Risk

It is impossible to asses risk in detail hence it is not the job of the test manager alone to asses risk. Therefore a large number of people should contribute to this- the customer, users, and development team etc. This not only increases the quality of the strategy but also has the advantage that the different parties are aware of the risks and the extent to which testing would contribute to make this testing manageable

Establish Test Techniques

The final step in a test strategy involves the selection of the appropriate test specification techniques that will be used to test the system. Choosing the techniques should take into account various factors, some of which are listed below:

Area of application

Some techniques are specifically suitable for testing the interaction (screens) between the system and the users; others are better in testing the performance of systems (Online shopping).

Use of resources

The application of a technique requires a specific amount of resources, in terms of both human and machine resources. For example the testing can be done on cross platforms like Windows, UNIX etc. And there are various applications in communication systems, which interact with certain devices like RDT (Remote Digital Terminal), USX1000 etc in two different platforms. Thus these resources have to be made available in advance prior to testing.

Required knowledge and skills

Not each tester is equipped for each technique. For the useful application of some techniques much business knowledge is needed. Therefore, the knowledge and skills of the test staff also influences the choice of the techniques.

The selection of the test specification techniques should be done in an early stage of the test process, for then the test team can equip themselves by getting trained in these techniques. Thus the idea behind in establishing a test technique is to have the most important tests take place as early as possible.

To put it in a nutshell the following are some of the key areas that should be addressed for establishing a well-defined test strategy:

Software risk issues

The purpose of discussing software risk is to determine what the primary focus of testing should be. Generally speaking, most organizations find that their resources are inadequate to test everything *in a given release. Outlining software risks helps the testers prioritize what to test, and allows them to concentrate on those areas that are likely to fail or have a large impact on the customer if they do fail.*

Features to Test

This is a listing of what will be tested from the user or customer point of view. For example if you were testing an Automated Teller Machine (ATM), feature to be tested will include Password Authentication, Deposit/Withdraw Money etc. This will give a brief overview to the testers of the number of major functionalities or features in a product, which increase the depth of testing on these features.

Hence pen down all the major features or functionalities that would have major setback on the customer when it is not properly tested.

Features not Tested

This section will record any features that will not be tested and why. There are many reasons that a particular feature may not be tested, for example you need to test a condition where the number of subscribers generating a call in a network would be 22,000. This may not always be feasible when the product is developed offshore where this infrastructure may be lacking. But whatever the reason a feature is listed in this section it all boils down the risk relatively.

Test Approach

The Test Strategy presents the recommended approach to testing of software applications. This approach should contain a description of how testing will be done and discuss any issues that have a major impact on the success of the testing and ultimately of the project. This section will throw light on various testing techniques and the objectives behind the use of this technique in testing.

For example:

- A **Sanity Testing** or a **Business Cycle Testing** validates basic activities the system needs to perform whenever each release is made
- A **User Interface Testing** will ensure that the User Interface provides the users with the appropriate access and navigation through the functions of the application. In addition this testing should ensure that the User Interface should conform to corporate or industry standards.
- **Performance Testing** will ensure that application is designed to test the run time performance of software within the context of an integrated system.

These are some examples and hence the appropriate testing techniques that would be best suited for the application has to be foreseen by the testing team so that a worth wile testing could be established. In addition, stating when these types of testing will be performed in the testing life cycle would give a clear perception to all those who are involved in the software development.

Another strategy issue that should probably be addressed is the use of tools and automation. Testing tools can be a boon to the development and testing staff, but they can also spell disaster if their use is not planned. Using some types of tools can actually require more time to learn, implement and run a test using these tools.

Training Needs

A testing team may involve mixture of testers who may or may not have adequate knowledge on the domain or to domain context. Hence training becomes essential and example of trainings might include use of test tools, Testing methodologies, Management systems like Defect Tracking, Configuration Management and use some support tools like Simulators, Network analyzers in case of communication systems will assist in the testing.

Responsibilities

This can be a matrix that quickly shows the major responsibilities such as establishment of test environment, Configuration Management etc. that are associated with the human resources who are nothing but the testing team. Prior to the assigning of responsibilities it would be worthy enough to consider the domain expertise of the test team.

Planning Risk Contingencies

Always an organization commitment is towards Quality and no one is against it but take a project where the schedule is at best ambitious and at most impossible. Your Quality Manager has promised the next release of your product on a certain date. The date seems aggressive with the available resources. Suppose a Programmer falls sick and thereby leaving a large gap in your knowledge base and now an ambitious schedule will now be a mission impossible. What are your choices?

- Alter the Schedule
- Reduce the scope
- Reduce the Quality (This means reducing testing thereby allowing more hidden defects)
- Add Resources (But it takes enough time to let him/her know about the project at this stage)

All the above choices seem bad. Hence this section will identify the risks and help us to focus and prioritize the testing to reduce them.

Conclusion

The testing of information systems should be based on the business risks, which the organization will experience in using these information systems. In practice, test managers should take the steps to come from risks and test the coverage in an intuitive manner. Good risk assessment is a part of these steps. In practice, the discussion of risks and related testing strategies in this way proves to be a real eye-opener for the concerned. It also enables negotiation of testing depth by having the customer decide which elements should be tested intensely. Thus the result of such a test strategy gives a better insight to everyone on how the testing will be done.

Establishing a Test Strategy is difficult but it should be borne in mind that all things are difficult before they are easy. Hence

"Do not follow where the path may lead. Go instead where there is no path, and leave a trail"

-Ralph Waldo Emerson

Thus this idea of establishing a test strategy is to come out of the conventional way of testing which is tedious and tiresome whereby a testing which has a strategy will not only be challenging but is worth doing.

Reference

I. Dr. Ingrid B. Ottevanger, "A Risk based Test Strategy", The Netherlands, 1999.