

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

Website:	www.expresscomputeronline.com
Date:	27 July 2009
Link:	http://www.expresscomputeronline.com/20090727/technology02.shtml

Leave no stones unturned

Mihir Kamdar underlines some important tips for software testing



Every test manager's nightmare is to find out that not enough testing has been done. Moreover, the paradox lies in the fact that enough testing often comes with a heavy tag in terms of cost and efforts. If you under-test the product due to time and cost limitations, eventually, the cost of support and maintenance will increase. The challenge is to get the optimum quality at the lowest cost.

Let us understand the problem in depth. Consider the case of a Web-based product for which around 1,000 test cases have been written. If this product is supported on two database servers, three Operating Systems, two Web / app servers and two Web browsers, it makes 24 combinations of the supported platforms.

Executing 1,000 test cases on each of these combinations would give any test manager a shot of confidence about the quality of the product considering 360-degree vigilance. However, this is tantamount to efforts of around 22 man-months, (assuming an average tester can execute around 50 test cases a day) which is too high and practically impossible. It would further delay the time to hit the market.

As a solution, the first thing that comes to mind is Automation. Automation can surely solve the problem. However, creating the automated testing environment will also require time and efforts. In addition, if the product under test is not stable, automation will require lots of maintenance efforts. Therefore, it is not practical to employ automation for such products. Once the product is stable, automation is a must for regression testing. For the products that are not yet stable, the only option remaining is to take a risk, and in a calculated way. How do you do that?

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

- **First things first.** The manager has to lay down the goals for the exercise. The goal should be to discover ALMOST ALL the defects within the permissible time and cost limits. Here is an approach to resolve the problem.
- **Know the features of the application under test well** - This will help you to identify the dependency of each and every feature with the platform under test. For example, if a particular Webpage is static in nature, you do not require testing it for different databases or OS. You may choose to test it for different Web browsers though as different browsers may render such pages differently. You need to know what component of the supported platforms would affect what feature.
- **Know the supported components and their intricacies well** - This would lead towards identifying the probable defects due to change in the platform component. For example, if Windows XP and Red Hat Linux are two supported OS flavors, knowing the difference between the two OS would be beneficial to identify how these differences can affect the product under test. Yours tests can be aligned towards looking out how these changes affect your system under test. For e.g. one difference between these two OS is that the file names are case sensitive in Linux, while they are not in Windows. Let's say that your product uploads files to the server running one of these OS. Windows would either overwrite or prompt for duplication if the file name case has been altered. Linux would simply store it as another file.
- **Come up with the list of the most important platforms.** This means that you will require knowing what the most popular platforms with your customers are. Even the least severe defect visible to everyone will be more critical than the most severe defect that is visible to only few customers. Therefore, it is important to test maximum on the most popular platforms. Doing so will lead to greater quality assurance coupled with customer satisfaction and lesser product support/maintenance.
- **Wear your thinking hat** - Make intelligent decisions on what features need not be tested or need not be fully tested on what component. It would be wise to include the developers and other stakeholders in this crucial decision making process. It is in this stage where you are identifying the risks – the calculated risks. Depending on a variety of factors like the stability of the product, the kind of defect prevention measures followed during development and some historical data about the defect counts in such kind of projects in your team, you can make a decision on how much repetition of tests is necessary. For example, if the product is not very stable, you might want to test it more. If the project is about enhancing an already stable product, you might just want to test it once or twice before releasing it. Knowing the kind of defect prevention measures followed during development and the historical defect trends would enable you to forecast the number of defects that can be uncovered. This will help you to identify how much testing is necessary. More number of defects means more testing is required before declaring a product eligible for release.

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

- **Risk and Mitigation Plan** - It is also important to identify the risks and have a mitigation plan ready at this stage. The risks may include scenarios where in more defects than planned are found and under-testing of any component on a particular platform. It becomes easy to choose what risks you want to take before going ahead, and re-plan your strategies if required.
- **Based on the decisions taken on what to test and what not to test**, come up with a matrix which lists down all the supported platforms on one axis and the features on the other axis and mark what features NEED to be tested (fully or partially) on which platform. Using this approach, you can significantly reduce the testing time and cost, and minimize the risk of leaving many stones unturned.

The author is a QA Lead with around six years of experience in Software Testing and Development. Kamdar is working with ennovate Technologies—a Software Division of eInfochips, Ahmedabad