Parametric Test Optimization

Shishank Gupta

Abstract:

In software, the narrowest sense of software product[1] quality is commonly recognized as lack of 'bugs' in the product. This definition is usually expressed as either defect rate or in terms of reliability of the software. With changing times, many product organizations have realized that this definition of quality isn't enough to highlight their software's competitive edge or to gauge customer satisfaction. They are now moving towards a Quality of Service (QoS) based approach to the process of software product development and testing. This approach helps the organizations in providing a well-defined and granular definition of *quality* for its software products, to its customers. Using this approach the Product companies can aim for quicker releases without compromising on *quality*. This paper examines how the *parametric approach* can be used to optimize the software testing process.

Paper:

The evolution of testing

Over the years, testing has evolved from a simple, mundane task to an extremely complex and probably the most critical and challenging activity in the product lifecycle. Some of the factors contributing towards this change include

- Shorter release cycles
- Complex functionality of the products being built.
- Specialized infrastructure and resource requirements for testing performance, usability etc.

The newer trends in testing have resulted in a change in people's outlook towards it. QoS is the latest buzzword in town. Many product companies have already mandated Reliability, Data Integrity, Performance, Scalability, and Usability testing before every product release.

[1] The term 'Product' in the document can also be applied to IS application in addition to commercial products.

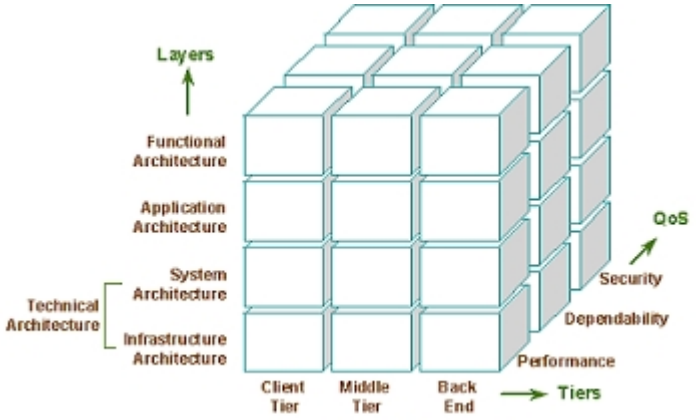

Fig 1 – Product architecture complexities

Fig 1 depicts the complexity of a typical Enterprise Architecture Framework that is organized across multiple dimensions. This complexity in architecture often translates into a complex test planning, execution and analysis process. Given this challenge, how does one ensure a cost and time effective testing process?

Optimization holds the Key

Quality can be classified into process quality and end-product quality. The product development lifecycle is complex and involves multiple stages. The output of the previous stage acts as an input to the next stage. Each intermediate deliverable has certain quality attributes that affect the quality of the end product. If each stage of product development cycle meets the requirements of the next stage, the end product thus developed will meet the specified requirements of the end-customer. Therefore a highly matured process quality is required in order to achieve a predictable end-product quality. Testing process, being the last stage before release of the product to the customer, holds the key to success of the product in the market. Considering the fact that time is always at a premium, the need for having an optimized testing process is therefore very essential.

Depending on the type of product and customers, there may be different weight factors for each *quality parameters*. While *quality parameters* like performance and reliability may be important in some cases, usability and maintainability may be critical for some others. To increase overall customer satisfaction and maintain the market edge, it is imperative that the requirements for these *quality parameters* are also considered along with the functionality of the product in the planning, design, implementation and testing process of the product. Though a number of optimization processes and methodologies are available for the development lifecycle, a few discuss testing. The *Parametric approach* addresses the optimization aspect for both the development and testing lifecycles. During testing, the approach helps the Test Managers in optimizing the testing process by prioritizing the test scenarios based on relative priority of each *quality parameter*.

Identifying and Prioritizing requirements

The first step in the *parametric approach* applied to testing is the identification of the different types of testing that has to be performed on the product. This is essential in order to plan the infrastructure and specialized resource requirements that might be needed for each type of testing. Once all the testing types are identified, the next step is to identify the specific requirements and the relative priorities of each of the *quality parameters* based on business drivers, the relative benefits and the risks associated with each of them. With the above information the Test Manager can identify the right set of activities and schedule them in the right sequence based on their relative priorities. In view of these discussions, it is not surprising that the *parametric approach* is applied at the very beginning of the development cycle, i.e. Requirements stage. The process of nailing down the functional requirements is complemented with identifying the quality profile of the product. This process may include anything from user research (usability) to discussions with technical experts (maintainability / security). Clearly spelling out the requirements for the *quality parameters* serves multiple causes for the product company. It helps

- Manage the customer's perception of the quality of the product
- Aid the development team in factoring these parameters in the product design, and most importantly
- Provide a reference for the test team to identify the right test scenarios and test cases in order to certify adherence to the product specifications.

The following is the suggested list of *quality parameters* that need to be considered in the planning and decision making process.

- Performance
- Usability
- Reliability and Integrity
- Portability and Interoperability
- Maintainability and Reusability
- Security

Fig 2 identifies a sample set of requirements for usability. Similar requirements should be collected for all the *quality parameters* and factored into the planning process.

Usability

	No User Interface	Single language	Multilingual Similar languages (English, French, Spanish etc)	Multilingual Different languages (English, Arabic, Japanese etc)	
Supported User Language	○	○	○	◉	
	No help requirement	Online help	Multilingual Online help	Context sensitive help	Multilingual context sensitive help
Help / UserDocumentation	◉	○	○	○	○
	No User Interface	Text	GUI	Web	
Type of User Interface	○	○	○	◉	
	No	Yes			
UI Standards	○	◉			
	No	Yes			
Shortcuts, Autoskip, Function keys	○	◉			
	No	Yes			
Media - Sound, Animation	○	◉			
	No	Yes			
Personalization support	○	◉			

Fig 2 – Sample Requirements for Quality Parameters

Optimizing testing lifecycle

The increasing competition for higher market share has resulted in a need for shorter product release cycles. This in turn has resulted in a situation where the testing teams are under pressure to deliver the

highest-level of software quality, but in a shorter duration of time. To face up to this challenge the *parametric approach* provides the Test Manager with a mechanism to optimize the testing process. The following table illustrates the manner in which the activities can be optimized at every stage of the testing cycle based on the requirements and their relative priorities.

Stage	**Activities**	**Advantages**
Test Strategy preparation	• Identifying the types of testing (security, usability, performance etc) required for the product based on market feedback. • Identifying the order and amount of coverage of testing based on relative priority (scope). • Estimating testing effort required.	• Reduced risk of market failure as the product is tested for *quality parameters* that mean most for its success. • The Test Manager can estimate better with the help of detailed information available to him.
Test Planning and test Case preparation	• Planning the lead-time for infrastructure and environment setup for different types of testing. • Identifying the various test scenarios based on the *quality parameter* requirements. • Creating a product specific checklist based on the *quality*	• The Test Manager can plan for the different parallel test activities for different types of testing. • The checklist ensures that all the *quality parameters* are tested to the required extend and nothing is missed. • Optimize effort for test case

	parameter requirements. • Determining the depth of the test cases based on the *quality parameter* requirements and relative priority. • Determining the order of testing (scheduling) based on relative priority of quality parameter.	preparation and execution • In case of time constraint, the Test Manager can be assured that the most critical quality parameter is tested, thus reducing risk of failure.
Test result analysis	• Analyzing test results	• Since test scenarios are written for each *quality parameter* of the product, the test results obtained can be used for a focused analysis and determining the order in which the defects would be fixed.

Table 1- Optimizing testing process

Applying the approach – a Case Study

Before a product company decides to focus attention on a particular quality or business priority, it is recommended that they determine what's important to customers. The product companies need to know:

- What would the customers value most at this specific stage of the product life cycle?
- Are there features that would influence customers' decisions to buy or not buy the product? What are they?
- What aspects of the product do customers perceive as drivers of their success?

Let us consider a real life example to illustrate how *parametric approach* can be applied to optimize the available resources. *Freemail.com* wants to launch a free e-mail service over the Internet. It promises personalization support for its users and a whole lot of other user-friendly features. Using this USP, *Freemail.com* plans to capture Internet traffic from other existing mail service providers.

Step 1

Very clearly in the case of *Freemail.com* 'Usability' is topmost in the priority list. It probably is of little concern to *Freemail.com* if the web page takes a second longer to load (performance) or if the mail service may have occasional downtime (reliability). What we have just done is identifying the relative priorities of the different *quality parameters* for *Freemail.com*. The next step is defining the requirements for each of these *quality parameters*.

Step 2

Let us try and define what 'Usability' would mean in Freemail.com's context. This could include the ability to organize folders, set preferences for incoming and outgoing mails, color, font, sound and animation preferences, ability to set filters etc. Similarly Freemail.com would define the requirements for the other quality parameters. A point to be noted however is that these requirements would complement the functional requirements of the product being built and cannot be built in isolation.

Interestingly, Step 1 and 2 can be interchanged and their order would be decided on the available information on a case-to-case basis.

Step 3

The next step from Freemail.com's Test Manager's perspective would be to use the information captured in the above steps for optimizing the testing process. The test strategy would indicate that usability tests would be performed either in parallel or before the ones for performance or reliability. Also since clear requirements for each of these quality parameters are available with the Test Manager, it would be easier to estimate the time required for test case preparation, execution and analysis.

Step 4

After the test strategy is finalized, *Freemail.com*'s Test Engineers would arrive at the test scenarios based on the requirements for each of the *quality parameters*. For instance, instead of trying to test anything and everything Usability stands for, specific test scenarios would be written in line with the usability requirements identified in Step 2, thus optimizing on the effort spent on test case preparation. Similar approach would be adopted for all the other *quality parameters.*

Step5

Test Execution

Step 6

Since test scenarios were identified for each *quality parameter*, the test results would be detailed enough to perform analysis at each parameter level. This process of test result analysis by *Freemail.com*'s Test manager and senior management would help them answer the following critical questions before the release.

- Is the current product quality in line with the requirements identified in Step2?

- Is the product therefore ready for release?

Assuming that all the defects are of similar criticality, the priorities identified in Step1 by *Freemail.com* would be used to determine the order in which the defects would be fixed and retested. Thus at the end of the testing cycle, *Freemail.com* can be confident that its product meets the requirements that were identified at the start of the project and even if some portion of testing could not be completed due to time constraint, *Freemail.com* can take comfort in the fact that it was probably the most insignificant portion that was left untested.

Conclusion

The *parametric approach* ensures that the time available with the testing teams is utilized in the most optimized manner while achieving the product's ultimate quality goals. The *parametric approach*, with it ability to aid identification, scope definition and prioritization of requirements is an excellent Project Manager tool. Using the available information, the Product Manager can evaluate the different available alternatives to arrive at the quality profile of the product. The approach also helps minimize the risk of failure of the product as it facilitates factoring of all the critical *quality parameters* in the decision making process. With most software projects world over failing to see a successful end due to time and cost overruns, the *parametric approach* may just be the answer.