2008

PACIFIC NW
SOFTWARE QUALITY CONFERENCE

Collaborative Quality

October 13-15, 2008

Conference Paper excerpt
From the

CONFERENCE PROCEEDINGS

Title: Is your Testing Effective and Efficient?

Author: Bhushan B. Gupta
Affiliation: Hewlett-Packard Company, 18110 SE 34th Avenue, Vancouver, WA 98683
Email: Bhushan.gupta@hp.com
Biographical Sketch:
Bhushan Gupta has 23 years of experience in software engineering, 13 of which have been in the software industry. Currently a Program Manager/Test Lead in the RPS group, at Hewlett-Packard, he joined the company as a software quality engineer in 1997. Since then he has led his groups in product development lifecycles, development methodology and execution processes, and software metrics for quality and software productivity.

As a change agent, Bhushan Gupta volunteers his time and energy for organizations that promote software quality. He has been a Vice President, a Program Co-Chair, and a Board Member of the Pacific Northwest Software Quality Conference. He offers a workshop titled "Engineering Software Quality" at the Center for Professional Development, OHSU, for software quality practitioners.

Bhushan Gupta has a MS degree in Computer Science from the New Mexico Institute of Mining and Technology, Socorro, New Mexico, 1985.

Abstract

Adequate testing is much more difficult when your product involves multiple facets such as software, hardware, Other Equipment Manufacturer (OEM) components, and industry compliance including safety and environment. Test coordination is more complex as there are multiple teams engaged in testing the product. It becomes increasingly difficult to ensure that all facets of the product are tested and there is no unintended test effort duplication.

The Retail Photo Solutions (RPS) group at Hewlett-Packard has developed a test planning method termed "Test Landscape" that assures a high level of test effectiveness and efficiency and yields a high quality product. It defines testing scope, identifies test ownership, and tracks test coverage and status across multiple development stages and quality attributes. The method involves identifying the quality attributes, such as Functionality, Usability, Reliability, Installation/Deployment, Safety, and Regulatory, that must be tested. These attributes form the horizontal test vector. To make sure that the product components are adequately tested before they are integrated, a vertical test vector representing development stages including Unit, Module, Component, System, and Solution and Beyond is also established. The two vectors combined yield a test matrix – the Test Landscape. The method is being used by the RPS group and has made test management simpler and efficient, while also enabling management to have more confidence in the testing process.

Introduction - Background and Challenges

The Retail Photo Solutions group at Hewlett-Packard develops solutions that enable the consumer to produce memorabilia such as calendars, posters, and albums from their own photographic work. The solution is comprised of:

- A software component that assembles images into the desired memorabilia format using image input devices (scanners, kiosks, memory cards etc.)
- Printing devices both HP and non-HP
- Production equipment such as CD/DVD for archival.

This is a global business so the solution must be localized and internationalized. Being global, it also has to comply with each country's regulatory requirements. The solution includes the OEM software and hardware components that are subject to the same standards of quality as the in-house components. The software and hardware development is co-located on multiple HP sites in North America and Europe.

This complex nature of the product makes its testing increasingly difficult. In particular the group faces the following challenges:

- Avoiding testing an integrated system before its individual components are sufficiently tested and stable
- Covering both the customer and the international regulatory perspective
- Optimization of overall testing to avoid test duplication
- Conducting the right testing at the right time in the development lifecycle, across the various stages of integration form component to solution.

Relevant Definitions

The following definitions are relevant for the foundation of this work:
Effective Testing: The test plan and its execution assures a minimal high priority defects found in the field
Efficient Testing: There is no unintended duplication of testing efforts

There are situations where some test duplication may be unavoidable. For example, a user interface is included in two different platforms supported by the solution with a slight variation. Both platforms will need testing, resulting in some duplication. We came up with the test landscape concept and used it as the primary method for organizing and communicating our test planning among the multiple involved groups.

Framework for the Test Landscape

The two vectors that define the test landscape are the product quality attributes and the different levels at which these attributes must be tested during product development. The quality attributes are the product characteristics in addition to functionality that a product must posses to provide value to its users. These characteristics include but are not limited to installation, usability, performance and form a sound basis for the product quality. Gupta and Beckman [1] have discussed the prominent software quality attributes. The

following table lists the important attributes and their definition from the sources highlighted in the table which the team used for this methodology:

Attribute	Definition	Source
Functionality	The capacity of a solution to provide its required functions under stated conditions for a specified period of time	Webster Dictionary
Usability	The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use	ISO 9241-11
Reliability	The ability of a solution or component to perform its required functions under stated conditions for a specified period of time	IEEE Standard Computer Dictionary, 1990
Installation	The capability of the software product to be installed in a specified environment	http://www.isi.edu/natural-language/mteval/html/222.html
Localization	Means of adapting for non-native environments, especially other nations and cultures	http://en.wikipedia.org/wiki/Internationalization_and_localization
Regulatory	Legal restrictions promulgated by government authority	http://en.wikipedia.org/wiki/Internationalization_and_localization
Security	Condition of being protected against danger or loss	http://en.wikipedia.org/wiki/Internationalization_and_localization
Compatibility	Exist or function in the same system or environment without mutual interference	http://en.wikipedia.org/wiki/Internationalization_and_localization

Table 1. Quality Attributes Used to Define the Landscape for a Photo Kiosk

This work provided the initial framework for the landscape and we also added Safety since it is relevant to the hardware devices. Generally the set of attributes that should be included in defining this vector will vary from product to product and business needs and should be carefully selected to get an optimized set. Wiegers [2] has provided a list of non-functional software quality attributes with the usage guidelines.

The second vector in the Test Landscape is the time during the product development when the testing should be conducted. This vector may vary depending upon the type of product, software vs. hardware, development methodology, iterative vs. sequential, and the specific shop practices followed by an organization The following diagram describes the main elements of this vector for a typical waterfall software development lifecycle:

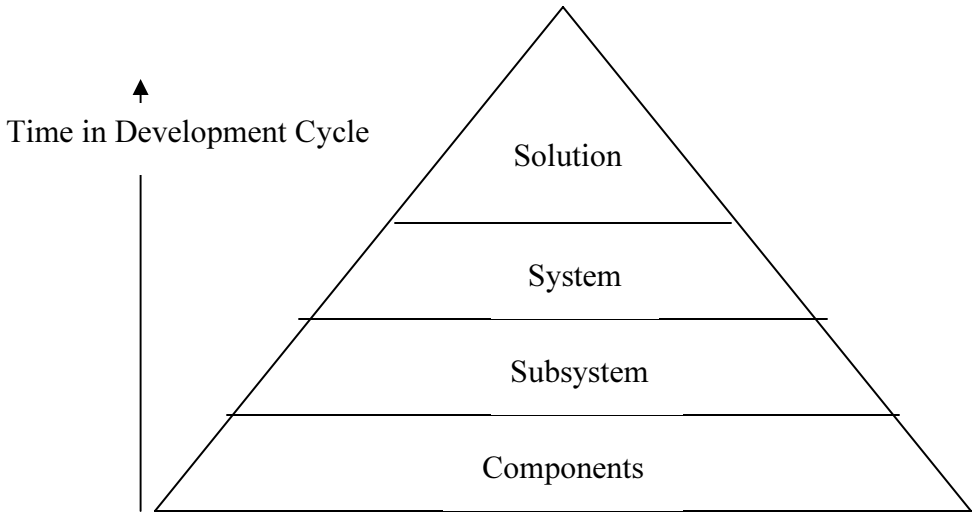

Figure 1. Test Types during the Product Development Stages

In this model integration testing happens at multiple levels as the development proceeds. The development stages can be customized for a particular environment to make the qualification more granular if desired. Craig and Jaskiel [3] and Kaner et al. [4] have discussed various testing stages during software development. The granularity does come at an added cost of qualification for the extra stages.

The two vectors, when combined together, result into the following Test Landscape:

	Functionality	Usability	Installation	Localization
Components	✓	✓		
Subsystem	✓		✓	
System	✓		✓	
Solution	✓	✓	✓	

Table 2. A Simple Test Landscape Showing Horizontal and Vertical Vectors

Our Experience

As discussed earlier, the RPS has a very complex product that includes software, hardware, and the OEM products. In addition, the development is based in the USA, Germany, and UK with each location having its own test team. Since the product is marketed internationally it is important to qualify it against the regulatory requirements. The development is primarily waterfall with multiple test-fix cycles after the "functionality complete" milestone has been reached.

The solution components are tested by the individual development teams at different levels i.e. subcomponent or unit, subsystem or module and system. Printing devices are also individually qualified for performance, reliability, and regulatory as needed. Since the teams are globally dispersed the testing is carried out in multiple places. A high level of coordination is essential for a successful overall solution testing.

The Quality and TCE (Total Customer Experience) director organized a taskforce to develop a test strategy to assure that:

- The testing was effective with no high priority defects found in the field meaning no test escape
- The testing was efficient to optimize the qualification cost
- The product had the intended quality measured against the release criteria.

The taskforce included stakeholders from development, quality assurance, custom product engineering, service and support, regulatory, and human factors engineering. Since the group consisted of development and test managers and leads from multiple test areas a general discussion started around what attributes should be tested and who should own what level of testing.

HP has well established test attributes and it was easy to create a basic attribute list that included Functionality, Localization, Usability, Reliability, and Performance as listed in Table 1. The subject matter experts from Service and Support and Regulatory brought in their perspectives which led to the creation of a broader well rounded list of attributes. An organization can build its own list of attributes that adequately characterizes the product quality.

The group then started to discuss different stages in the in the development when these attributes should be tested. Since the solution is made up of hardware, software and OEM products, the levels had to represent all the stages involved in each development. The OEM products could only be tested at the system level while the hardware and software testing could begin as soon as a component development was complete. There was no clear consensus on the stage names or definitions and the team struggled in getting alignment on characterization of these stages. Finally an agreement was reached to use the simple notion of levels (Level 1, Level 2 etc.) to match the development stages. For example Level 1 represented the Unit/subcomponent, Level 2 the module/subassemblies and so on and so forth. The equivalent of levels is shown in our tables to avoid confusion.

The quality attributes and the test stages together provided the framework for the landscape. We used the landscape table to assign and agree upon ownership of testing for each attribute at each level. The group developed a landscape for each component especially hardware and an overall landscape at the product level to provide efficient test planning at all levels.

Table 3 shows a complete test landscape for a printing device that was a component of the solution.

	Functionality	**Reliability**	**Serviceability**	**Performance**	**Regulatory**	**Safety**	**Output Quality**
Subcomponents	Dev. Team	Dev. Team	CPE	Dev. Team	NA	Dev. Team	Dev. Team
Component	Dev. Team	QA	CPE	QA + Dev. Team	QA	QA	Dev. Team

System	Dev. Team + QA	QA	CPE	Dev. Team + QA	QA	QA	Dev. Team
Solution	QA	UNKNOWN	NA	CPE	NA	NA	CPE
Alpha	CPE	CPE	CPE	NA	NA	NA	NA
Beta	Retailer	Retailer	Supp	UNKNOWN	NA	NA	Supp
Acceptance	Retailer	Retailer	Supp	Supp	NA	NA	Supp

Table 3. Test Ownership of a Printing Device

The abbreviations used in the table are:

Dev: Product Development
QA: Quality Assurance
Supp: Customer Support
CPE: Custom Product Engineering
NA: Not Applicable

Both Beta and Acceptance test stages are focused on testing on the retailer site for the end customer use.

Some typical characteristics of the landscape during component development are:

- The Development Team has a heavy role to play in the beginning and their involvement decreases as the component/product development matures. At the same time, the involvement of other specialty teams increases as we move towards the final product. This is often the case as the components are assembled and the solution starts to exhibit end-product characteristics such as Usability, Performance that require testing by subject matter experts.
- There may be unresolved areas of testing that still need to be finalized. They are highlighted as UNKNOWN and can become potentially critical issues if not resolved early in the program.

The same test landscape can also be used to communicate test status, as shown in Table 4. Once again the component is a printing device (same as in Table 3).

	Reliability	**Performance**	**Regulatory**
Subcomponents			
Finisher	QA – behind schedule	Dev. Team + QA – behind schedule	QA – on track
Engine	QA – on track	Dev. Team + QA – behind schedule	QA – on track
Component			
Printer	QA – behind schedule	Dev. Team + QA – behind schedule	QA – on track

Table 4. Tracking Status for one of the Components of the Photo Kiosk Solution

Table 4 is a snapshot at a milestone in the product development lifecycle where the component was being evaluated. The information was used as a part of the dashboard to inform the upper management.

Table 5 represents the solution test landscape for the product.

	Functionality	Reliability	Serviceability	Performance	Regulatory	Security	Output Quality
Subcomponents	NA	NA	NA	NA	NA	NA	NA
Component	NA	NA	NA	NA	NA	NA	NA
System	NA	NA	NA	NA	NA	NA	NA
Solution	CPE	CPE	UNKNOWN	CPE	NA	UNKNOWN	CPE
Alpha	CPE	CPE	UNKNOWN	CPE	NA	NA	CPE
Beta	SUPP	SUPP	SUPP	SUPP	SUPP	SUPP	SUPP
Acceptance	Retailer	Retailer	Retailer	Retailer	Retailer	Retailer	Retailer

Table 5. Photo Kiosk Solution Test Landscape

The attributes at the levels prior to the solution level have been marked NA since testing at those levels had already taken place during the component development and system testing. For example, Table 4 shows that the Printing Device has been tested at all levels and is now being included in the solution. The Regulatory testing was not shown since the solution components subject to regulations have been tested at one more levels earlier in the development.

Once again the Test Landscape revealed that there were some areas with missing test ownership. The test landscape identified these gaps and raised awareness to the program management teams. This helped us focus on the critical business needs and achieve the desired level of quality and test effectiveness. Using the test landscape, we also discovered that there was a fair amount of overlap between the system and the solution testing. This was primarily due to the lack of clear definition of the two levels and lack of clarity of the roles and responsibilities of the two teams involved in qualification. The test landscape provided a clearly understandable framework which enabled the two groups to align on what the solution testing must accomplish which is different than the system testing. This led the solution team to consider typical use case scenarios such as "Busy Mom" which was portrayed as some one who did not have time to read the instructions and intuitively proceeded to produce her memorabilia. It resulted in an effective user scenario testing which was not being performed earlier. Solution reliability was also another area where testing improved.

Aligning Test Landscape with the Product Development:

To be effective, the test landscape must be designed very early in the development lifecycle. The quality attributes should be determined immediately after the product use cases have been established and the software system requirements are complete. This is equivalent to the requirement definition phase of the waterfall development or the release planning milestone of the agile development.

As the product development proceeds, the landscape must be reviewed and updated if necessary at the various checkpoints and milestones. Our experience was less than perfect with the landscape review. Some component teams proactively reviewed their test landscape while others had to be reminded to complete this activity. There were instances where the review was inadequate. By the time product was released, there was

a strong emphasis on the methodology and a better understanding of how it should be utilized.

Benefits of Test Landscape

The Test Landscape has multiple benefits that contribute to a quality product without any additional cost. The following section discusses these benefits in detail.

Test Effectiveness

The landscape builds the test effectiveness by making sure that each applicable quality attribute is tested at an appropriate time during product development thereby providing test coverage from the unit test to the solution test. An example would be to test the performance at component, subsystem, system, and solution level to achieve the intended solution performance. Early testing and defect removal leads to a lower development cost and a superior quality product as the longer a defect stays in the system the more expensive it becomes to fix it [5, 6].

Test Efficiency

Establishing early ownership and clear definition of each test area eliminates duplication of testing, establishes clear roles and responsibilities, and provides a mechanism where testing effort is well understood and is not an afterthought.

Test Coordination

The Test Landscape, after having determined the important critical test areas, can provide effective test coordination to balance resources, assigning testing tasks to appropriate teams, and placing mechanisms in place to analyze test progress and test results.

Scalability

The test landscape is scalable from the component to subsystem, to system and all the way up to the solution level. Depending upon the scope of the product, both the quality attributes and the testing stages can be altered to achieve effective testing.

Customization

The user of the landscape has the liberty of focusing on what is most important to their environment. At times, especially when breaking out into new markets, the functionality is of paramount importance while other quality attributes may not play such an important role. For RPS product it was important to provide excellent usability so that a novice user from the street can get his/her memorabilia while high performance was less critical.

Status Reporting

At every checkpoint or milestone during the product development, the landscape provides a mechanism to track the testing status and thus evaluate the product quality and any schedule risks. At the beginning of the product the landscape can be used to establish the ownership and then as the product development moves along, to evaluate if the intended testing has been performed or not. If, for some reason, the planned testing could not be achieved, a risk analysis can be carried out and the mitigation plans can be put into place.

Conclusion

It all comes down to product quality within the well known constraints – scope, schedule, and resources. Our experience shows that use of the Test Landscape in test planning contributes to the higher product quality, shortens the schedule and optimizes the testing

resources. The higher product quality is achieved by testing all the relevant product attributes based upon business needs at the right time in the product development. Identifying and removing unintended duplication contributes to both lower cost and shorter schedule. In most cases testing is the last activity in the product development and is on a critical path. Establishing the testing gaps early in the lifecycle and along the product development helps risk mitigation and potential schedule slip especially in the large organization where each group is focusing on a component or a subsystem.

Acknowledgement

The author would like to recognize the RPS Quality and TCE director, Lee Mason and the members of the taskforce, Amy Battles, David Boal, Tim Dummer, Bernardo Gutierrez, Rick Johnson, Shelly Reasoner, Todd Walker, and Wayne Westly for their enormous contribution in the development of the Test Landscape concept and their support for its adoption by the RPS Group at Hewlett-Packard.

References:

1. Gupta, Bhushan B. and Beckman, Orhan Ph. D., Quantifying Software Quality – Making Informed Decisions, Pacific Northwest Software Quality Conference, Portland, Oregon, 2006
2. Wiegers, Karl E., Software Requirements, 2nd Edition , Ch. 12, Page 216, Microsoft Press, ISBN 0-7356-1879-8
3. Craig, Rick D. and Jaskiel, Stefan P., Systematic Software Testing, Artech House, 2002, ISBN 1580535089, 9781580535083
4. Kaner, Cem, Falk, Jack and Nguyen, Hung Quoc, Testing Computer Software, 2nd Edition, ISBN: 0-442-01361-2
5. Boehm, B. and Basili, V., "Software Defect Reduction Top 10 List," IEEE Computer, IEEE Computer Society, Vol. 34, No. 1, January 2001, pp. 135-137.
6. Cigital, Case Study: Finding Defects Early Yields Enormous Savings, http://www.cigital.com/solutions/roi-cs2.php