

## ***Adaptable QA Puts Quality Back Into the Software Rapid Development Environment***

By Brendan Joyce, Director of Quality Assurance  
Critical Path Software, Inc.

As with most exciting new methods that promise better-faster-cheaper innovation, the choice to implement the Rapid Development Environment (RDE) approach for new software product development is often at the sacrifice of quality. However, the fast pace of development conducted in the spirit of market gains is not the only culprit for a steep decline in software quality. The software quality assurance (QA) and test modus operandi itself is to blame. Software QA and test fail in the RDE due to rigid QA methodologies that haven't evolved to match the pace of change in the development process. A new era of QA in software will be built upon a foundation of *adaptability*. There are three characteristics embodied by just such an adaptive QA process: creativity, flexibility and efficiency. Just as those descriptions might seem obvious, adaptive quality assurance challenges the team to approach test plan and QA processes with a focus on everything but the obvious.

This paper will characterize each of these elements of an RDE-like quality assurance and test environment and point out where testers are taken for granted. The author will demonstrate a path to the restoration of quality in a hyper time-sensitive software development market.

### **Creativity: Worldly Guru Testers are Real People, Too**

Creativity may not be the typical association made with quality assurance and test professionals, but testers themselves might consider it their best asset and the trait that ranks them among the best in the business. A team that can adapt lends an authentic sense of caring and ownership to the product; an attitude rare in a testing team, whether internal or outsourced. Sacrificing software quality for time-to-market pressure can be frustrating and demoralizing for a passionate QA team. Turning this into a challenge while mitigating the negative effect is not easy. It's obvious that by providing a creative test team with exceptional equipment, training, and an active approach to the task at hand, one can keep the focus on nimbleness and open-minded thinking. The result is a pure,

unadulterated critique of how the products stack up along with meaningful usability and design input.

Adaptive QA contests that the people are the most important component of an adaptable approach to quality. Allowing exceptional people to develop in a creative and highly motivating atmosphere is critical for retaining the level of talent required to test software in a fast-paced Rapid Development Environment (RDE). What is less obvious are the methods best used to foster the passion and integrity that is the cornerstone an adaptive approach to QA.

Developing a testing environment that encourages creative approaches to testing diverse types of software requires a firm commitment to the goals and motivations of each member of a QA department. Motivating the team to aggressively improve the testing of each individual product must outweigh the negative aspects of time-to-market pressure. By challenging exceptional individuals to use their abilities to adapt the process to the environment, they develop positive attitudes toward their involvement in the process and ultimately to the products that they test. As this passion grows the adaptations to each successive project matures as the team further heightens its attention to various adaptations.

Preventing distraction from turnover among the team by providing an exceptional testing environment makes for the ultimate in adaptability with a team that is mobilized like an expert recon team. They focus on developing their skills in key areas and apply those skills to the project at hand. The payoff is a long-term relationship between the team and product with exponentially reduced ramp-up time at each new start.

Tailored educational opportunities enhance tester's creativity, resulting in a well-educated staff of testers that understand the products, target audience and what's going on outside their cube in the lab. Using that knowledge, they can then identify areas that are critical for testing based on their understanding of the products and the nature of competitive products in the same category. For instance, having a significant amount of exposure to the unique characteristics of various authors of, versions of and user experiences with web browsers leads the team members to tailor test cases to the most common areas of incompatibility. Using that knowledge to recommend more emphasis on some areas and less on others is one way of balancing time constraints and budget

constraints with the amount of coverage to apply to a product or a component of that product.

An aggressive, well-targeted approach to testing software is imperative when faced with a seemingly impossible schedule and is equally effective with adequate testing time. To help deal with the short schedule, one can get a jump on the testing by assigning a creative testing staff to pre-Alpha projects, applying system knowledge and wide range of expertise to test completed components before the product hits the Alpha milestone without a significant impact on the development schedule. The effective fostering and on-going credence paid to that team's creativity allows all these steps to move rapidly.

### **Flexibility: Flexing and Toning the Test Plan and Process**

Flexing to the needs of the project requires an osmosis-like integration of the adaptive approach into the test plans and bug tracking systems within the baseline QA process. As a fluid testing environment, adaptive quality assurance process extends beyond testing plans and procedures. On top of the basic skills the right people possess the ability to shift gears as needed to accommodate the needs of the client, of the test, or of the testing department involved.

The desired outcome is to:

- Adapt to a seemingly impossible schedule;
- Test the product as early as possible;
- Allow intelligent utilization of the product type to determine the most critical test areas; and,
- yield all of the information required to improve the end-user products.

Since *time* is the catalyst driving the opportunity for success or failure, it is considered the bane of flexibility in a QA and test process. Developing test plans can be time-consuming and unfortunately time-wasteful. If the test plan development is not adapted to the evolution of the software product then the potential for waste is tremendous. A test plan developed for a feature from a specification that is subsequently changed prior to the feature development is frequently worthless.

This situation is avoided when the QA manager or QA lead work closely with the product manager to ensure they understand what areas of the product are likely to change.

The test plan development schedule can then be adapted to help ensure that the testing documentation is an effective resource for the testing staff. Inherent to the adaptive QA process when used by the outsource test team work with existing QA teams, sharing testing load, assisting with coordination, planning the distribution of labor, and avoiding conflict. Outsource can be most beneficial when the company leverages outside experiences with new leading edge technologies, new programming languages and methodologies, and new hardware and software for early entry into or competition with new product areas.

To complement a staff of creative people, the lab must be flexible enough for quick adaptation to the project at hand. Move a cable or two and test through a proxy environment. Disconnect a cable and the test machine is using a dial-up connection to the Internet. Work on lab machines setup and ready to go in the latest configurations to testing speeds ranging from the now rare 386 to the latest Pentium processor with the latest 3D accelerated video cards or latest configurations. Major OS versions are available to load on each machine in a matter of minutes. The latest browsers, and the current OS and major product patches are available on a file server. Search access from any machine using a web browser to locate anything stored in our software library. These may all be obvious components of a QA lab, but the critical elements must not only be for show but for rapid access and use without cumbersome “sign-up” and “check-out” procedures or hierarchical roadblocks.

### **Efficiency: QA’s Evolution to True Support the RDE**

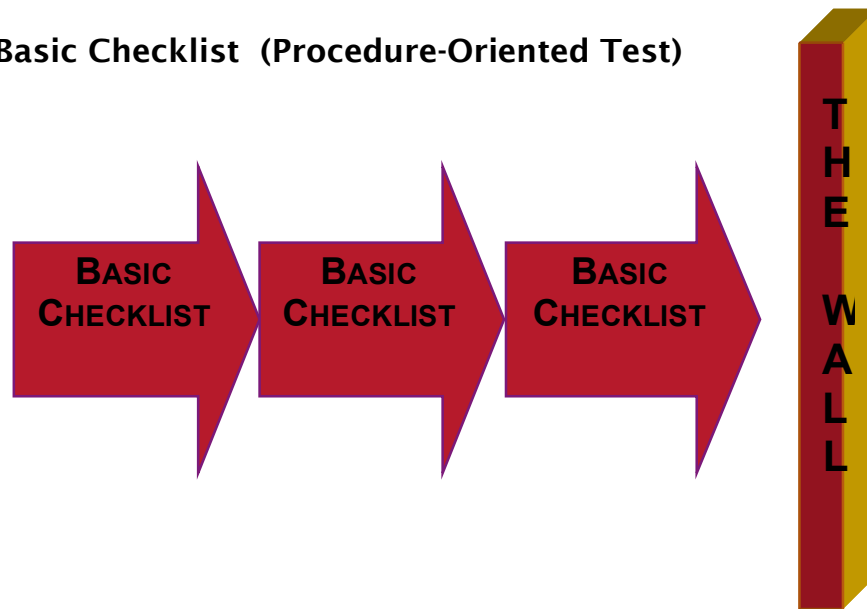
At the core of the productivity issue surrounding methodic QA is the trap of internal test departments that follow a basic test plan for each build of a product. Perhaps this involves testing for one week, waiting a week for the fixes, retesting the same items the second week, and so on. Sometimes it means hiring a contractor for a week and sitting them in front of a checklist that serves as the only guideline for test coverage for that product.

The problem with this method is that it is easy to overlook critical areas that need to be tested. If the person isn’t comfortable with the product or doesn’t have a thorough understanding of the industry the product is geared toward, they are likely to miss the

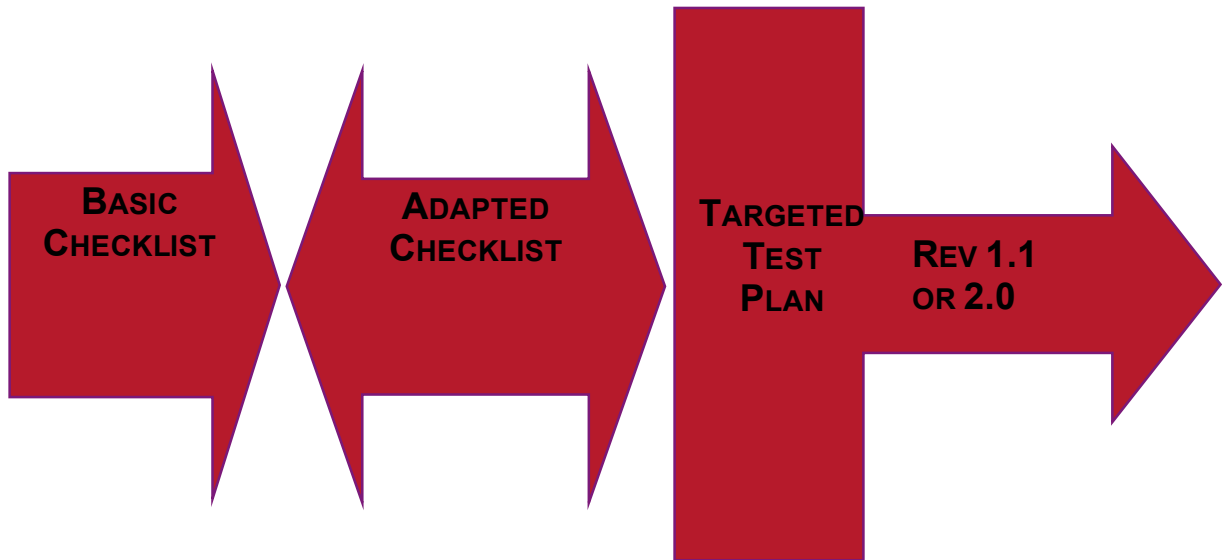
most important areas that the target customer depends on. If the project changes or significant new technologies are incorporated in a release, the basic checklist may not do enough to stress the areas that have changed the most from release to release.

When the test process *can* change as necessary to meet the evolving needs and schedule of the product development, this trap is foiled. By using the creativity and flexibility of AdaptiveQA™ to attack the problem as though it were a living, breathing entity the development process in a RDE becomes, by nature, an evolutionary one. Critical Path Software has formally designated AdaptiveQA as the means by which to properly cope with this evolution of the testing process.

**Table 1: Basic Checklist (Procedure-Oriented Test)**



**Table 2: AdaptiveQA™ Approach**



As the graphics above indicates by comparison, AdaptiveQA™ test coverage matures over time to create a continuous loop of improvement, while the Basic Checklist method repeats mutually exclusive, static tests. This does not mean AdaptiveQA requires spending additional time over the course of the project. Rather, it's an approach that yields a tidy, sequential set of documents that provide the next version of the product with a full-featured set of test plans. This provides the testing team a variety of documented options for attacking the bugs that are the side effect of the evolutionary development process. Additionally, it highlights modifications performed throughout the testing.

It obviously takes a different method to examine the caterpillar compared to the method it takes to examine the subsequent butterfly. What's not obvious is the special nuances that must be appreciated in each species. In the same way, it may take a modified or a unique method to test an application in the pre-Alpha, Alpha, or Beta stages. It remains most efficient to apply whatever limited resources available to each stage, in the best manner possible.

Meeting the specifications and product deadlines by making the best use of QA time requires an understanding of system-level architectures, common problems that are found in complex programming environments, and involvement by senior developers.

These assessments will help to determine what components will need the most testing time. Organizing the lab and developing procedures that lessen the overhead and keep the team focused on functional duties maximizes the time available.

Adaptability means working off test plan templates for developing test cases and test plans for a project, and expanding test cases as the testing progresses to ensure proper tracking of new learnings over the course of the project. More than just saving the reams of paper and scratched notes that track the changes, software testing should include organized and usable documentation as time goes on that evolves in to a suite of test plans, checklists, and test cases that form the basic checklist at the start of the next product. Whether adapting plans within a single turn of the software or adapting them for successive projects, the expectation is that the more targeted testing done, the better tested the project, and the more efficient the long-range test process.

For example, let's track the second version of an Internet-based product tested by an outsource team that provided the testing for the first version. There's only two weeks for both functional test as well as regression of defects on bugs found during the first pass. The test of the first version included a compatibility test on five major brands of PCs and four off brands from 486 to Pentium III.

The results show no hardware incompatibilities in the first version, but do point to several problems using the product with particular versions of supported browsers and one conflict with a software utility program. Under these circumstances the specification must be examined for any changes that might increase the need for hardware compatibility testing and project notes. If there isn't any evidence that hardware has become more critical, then the focus on hardware reduces significantly and a more and expanded focus is put toward the matrix of software testing. This test project adapted to a focus on what is learned with the first version. Rather than follow a rigid plan arbitrary to past learning or future product implications, the correct shift in coverage on a more critical aspect of the product conducted in quick time reduced the coverage on a less-risky area of the testing without increasing the time needed to test the product.

The continuous loop nature of the mature test plan is superior to the testing development that starts and ends with the specification initially provided by developers. There are numerous reasons why this is the case. Specifications may be put together at

the last minute and may not be completed until after development is in process. (We know this is not the ideal scenario, but we also know that it happens). In this case, basing a test plan entirely on the specification will likely miss numerous areas that were not specified or that were incomplete.

Having an exceptional group of experienced, individual thinkers is required at each stage of an adaptive QA and test process to match the speed and complexity of software RDE. The tester involved must be able to determine what needs more coverage and what to sacrifice in order to meet the schedule. The product may evolve due to responding to changes in the industry, changes in customer needs, or to react to new marketing requirements. Since reducing coverage in one area of the product does not mean eliminating coverage, the tester has a safety net in that the areas of reduced coverage still get some attention. Therefore, if this area becomes more problematic the tester discovers this as a normal part of the process and can again adapt to the problem.

The testing department must be capable of adapting to this ever-changing environment in order to effectively target testing at the critical areas of highest risk without sacrificing base coverage. The computer neophyte has taken a place next to the power user as a viable market for software products. Simultaneously, the time from product idea to product shipment has compressed dramatically. If your testing process can adapt without sacrificing quality with that pace of development, you have a good chance of succeeding in this challenging, and deft, environment.

###