

# Tips for Preparing for the Quality Assurance (QA) Phase

January 2007 - Pragmatic Software Newsletters

How can you ensure that your software is fully tested and has high quality, before moving to production? Here are some tips on preparing for Quality Assurance (QA). Next month, we will discuss how to measure the QA progress once QA begins.

## **Tip 1 - Quality Assurance begins during Specification Review**

The QA team should have the opportunity to review each specification and to find issues and make suggestions. The QA team should look for unclear requirements, requirements that are not defined well enough to create test cases around, and for requirements that might affect other areas of the software. Time spent in requirements review and correction can pay big dividends.

## **Tip 2 - Test Cases Should Be Developed as Coding is Progressing**

If you wait until the QA phase begins to create your test suite, test cases will be rushed and your team will not have time to fully review each suite of test cases for each requirements. Test case development should begin the day coding starts. Testers should be assigned to create test cases for specific requirements. Testers should remember to create test cases for:

- **Positive Testing** - These test cases ensure the software will work exactly as specified in the requirement.
- **Negative Testing** - These test cases are used to try to "trick" the software. For example, try entering in an invalid date in a date field. Try entering character data in a numeric field. Try entering in a date range where the "from date" is older than the "thru date". This also includes entering in data that contains apostrophes, as this tends to trip up SQL based systems.
- **Bounds Testing** - These test cases test the bounds of each field. For example, if a field is defined as 50 characters, try entering 60 characters.
- **Relational Testing** - These test cases test parent-child relationships. For example, if you have a parent-child feature you are testing (e.g. an invoice may have one or more invoice line items), try deleting the parent (invoice in this example), then ensure that all the child items (invoice line items in this example) were deleted.
- **Performance Testing** - These test cases ensure that the new release will perform as quickly (or quicker) than the past release. To test this, prior to the new release, try different actions (add a record, search for a record, update a record, delete a record, etc.) and record your timings in a spreadsheet. Once the new release is in the QA environment, try those same tests and record the new timings, this will tell you performance has improved or degraded.
- **Regression Testing** - Upon a successful test cycle, some of the test cases above should be marked as regression test cases so that they are run in future releases to ensure that existing features continue to work as designed.
- **Smoke Tests** - Once all the test cases are designed for all the requirements, a small set (10 to 30 test cases) of the positive test cases should be identified as Smoke Tests. These will be run prior to beginning the major testing effort. If any of these fail, they should be fixed immediately before testing begins so that time is well spent by the testing team.

Here is an example of a report that shows the Smoke Test Listing:

<http://www.pragmaticsw.com/newsletters/SmokeTest.pdf>

## **Tip 3 - Test Cases Should Be Reviewed By Team Members**

Once the tester completes all the test cases for a requirement, the tester should distribute a test case review report that shows each test case for the requirement. The test case review report should show the steps to be run, expected results, and what section of the requirement the test case pertains to. The test case report should be reviewed by the appropriate team members (project manager, programmers, and other testers). Reviewers should identify missing test cases and test cases that are unclear. The review meeting should only take about 30 minutes, as each reviewer should come to the meeting prepared, just to identify missing or unclear test cases. The tester will then make the changes and mark that requirement as Test Ready.

Here is an example of a Test Case Review report:

<http://www.pragmaticsw.com/newsletters/TestCaseReview.pdf>

#### **Tip 4 - Review Test Coverage**

Upon completing all test cases for all requirements (and having them reviewed), do a quick analysis of number of test cases by type (Positive, Negative, etc) by Specification to ensure you have good test coverage.

Here is an example of how that might look:

<http://www.pragmaticsw.com/newsletters/TestCaseCoverage.pdf>

## **Helpful Templates**

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines** - <http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf>
- **Functional Specifications** - <http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf>
- **Architectural Overview** - <http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf>
- **Detailed Design** - <http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf>
- **Strategic Planning Document** - <http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf>
- **Test Design** - <http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf>
- **Risk Assessment** - <http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf>
- **Weekly Status** - <http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf>
- **User Acceptance Test Release Report** - <http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf>
- **Post Mortem Report** - <http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf>
- **All Templates** - <http://www.PragmaticSW.com/Templates.htm>
- **Prior Newsletters** - <http://www.PragmaticSW.com/Newsletters.htm>
- **Software Planner** - <http://www.SoftwarePlanner.com/SoftwarePlannerPro.asp>
- **Defect Tracker** - <http://www.DefectTracker.com>
- **Remoteus (Remote Desktop Sharing)** - <http://www.PragmaticSW.com/Remoteus.asp>

### **About the Author**

Steve Miller is the President of **Pragmatic Software** (<http://www.PragmaticSW.com>). With over 21 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at <http://www.PragmaticSW.com/Newsletters.htm>. Steve's email is [steve.miller@PragmaticSW.com](mailto:steve.miller@PragmaticSW.com).

---

Pragmatic Software Co., Inc.  
383 Inverness Parkway  
Suite 280  
Englewood, CO 80112

Phone: 303.768.7480  
Fax: 303.768.7481  
Web site: <http://www.PragmaticSW.com>  
E-mail: [info@PragmaticSW.com](mailto:info@PragmaticSW.com)