Test Plan Template

1. Document Information

1.1. Revision Information

Include a version number, the date modified, the person who made the changes, a summary of the sections that were changed and the reason for the revision.

1.2. Approvals

Before declaring Alpha, make sure that each member of the project leadership team reviews your test plan. Typically, you'll want to set up a meeting to review the test plan and get sign-off from the project manager, development manager(s), and marketing manager. The goal of this exercise is to validate your testing approach, get help on closing out any open issues, make sure that your basic assumptions are correct, and provide education on the QA process.

1.3. Executive Summary

2. Table of Contents

3. Overall Test Strategy

Start with a statement which summarizes the testing strategies you expect to use during the course of the project. List every test suite and tool you expect to use and whether it already exists, partially exists, or needs to be developed.

3.1. API Testing

3.2. User-level Testing

3.3. Compatibility

- **Platform Compatibility**
- **Application Compatibility**
- **Peripheral Compatibility**

3.4. Customer Seeding

4. Test Strategy By Component

List each component area. For each component summarize what testing method will be used, what features / APIs are already covered by existing test cases or tools, what features / APIs will have new test cases or tools developed, and most importantly what features / APIs will not be explicitly tested. It's also good to estimate how stable each area is expected to be from any previous versions of the code base. If possible, try to assign a risk priority to each area so you can focus your testing on the hottest areas during the course of the project.

5. Test Strategy By Configuration

List all hardware, software, network, and peripheral configuration that will be tested. For each class of configuration, estimate the risk associated with finding unique bugs as that configuration changes. For example, there may be a good chance of discovering a bug that's unique to the minimum supported memory configuration while the chances of finding a bug unique to a particular model of LaserWriter may be remote.

6. QA Roadmap

For each phase of the project, summarize the key activities and deliverables for QA. List what kinds of tests will be run, the level of quality you expect, what kind of seeding activity will take place, the specific milestone criteria, and what could put QA on the critical path for that phase of the project (typically this can occur prior to alpha acceptance and during each milestone acceptance period).

6.1. Pre-Alpha

6.2. **Alpha Candidate Testing**

6.3. **Alpha Testing**

6.4. **Beta Candidate Testing**

6.5. **Beta Testing**

6.6. **Final Candidate Testing**

6.7. **Final Testing**

6.8. **GM Testing**

6.9. **Patch Testing (if applicable)**

7. Bug Process

Explain internal and external (if it applies) bug reporting process, as well as how bugs are verified, closed and regressed (re-verify closed bugs toward the end of the project). In addition, it can be good to display the bug component tree in appendix.

8. Build Process

Detail the build process and, if it applies, when and which builds will be released to partners and / or licensees.

9. Roles and Responsibilities

List all QA-lead positions for the project, who is assigned, and when they're expected to start (try to use general milestone-terms here, project schedules change frequently). Also summarize the overall number of testers reporting to each lead and when they need to start. If any other testing groups are involved, list their specified responsibilities and what processes (teleconferences, email, co-location, etc.) will be used to coordinate testing efforts between the test teams.

10. Known Risks/Concerns/Contingencies

The main topics to cover here are the known project risks from QA's perspective (significant changes to the code base, API documentation is incomplete, not enough time to qualify candidate builds, limited availability of test equipment, etc). Then list how you intend to minimize each of these known risks.

11. Potential Risks/Concerns/Contingencies

The main topics to cover here are the potential project risks from QA's perspective (missed milestone candidates dates, other development groups not delivering as promised, testers get tied up finishing other projects, testing headcount is cut back, other QA groups fail to live up to their testing obligations, etc). Then list how you intend to deal with each of these potential threats should they occur.

12. Appendices

12.1. **Test Cases**

If applicable, keep a live list of known test cases in the this appendix and update it as the project progresses. If the list of test cases is too large, keep it in a database and refer to it in the "Related Documents" appendix.

12.2. **Open issues**

If applicable, keep a live list of known open issues in the this appendix and update it as the project progresses. If possible, try to assign a DRI (designated responsible individual) and a resolution date for each issue.

12.3. **Related Documents**

List the following documents along with their author and location: Engineering Requirements Specification (ERS), Marketing Requirements Document (MRD) and Licensing Requirements Document (LRD).