«File Searcher» Project

Test Plan

Background: Estimations, schedule, strategy, and metrics are needed to organize the testing process efficiently**.**

Purpose: To organize the testing process effective and efficient during the whole project period

Scope: Testing process description, metrics, schedule, resources.

Audience: Management staff, QA team, project team.

Contents

1. Project scope and main goals

Efficiently automate the search and retrieval of files across various directories based on specific user criteria, with performance significantly surpassing manual file search capabilities.

2. Requirements to be tested

- **2.1.** **UR-1**.: **File Search by Directory and Depth**(Critical Path test.)
- **2.2.** **UR-2**.: **File Types for Search** (audio, video, office files) (Critical Path test.)
- **2.3.** **UR-3, L-1**: **Displaying Information for Found Files** (name, path, size, creation date, first frame screenshot) (Functional Test, Critical Path Test.)
- **2.4.** **BR-1,SC-1**:**Performance**(500files per second) (Performance Test.)
- **2.5.** **BR-2, SC-2**: **Operation Time Limit** (1 hour) Endurance Test.
- **2.6.** **UR-4, BR-3**: **Multilingual Support** (Russian and English) (Smoke Test, Functional Test)
- **2.7.** **QA-1**.: **Logging** (logging operations, stopping at 1 MB, displaying the current directory)
- **2.8.** **BR-4**.: **Support for Windows and UNIX File Systems** (Compatibility Test)
- **2.9.** **BR-5**.: **Network Support** (Integration Test)

3. Requirements NOT to be tested

- **3.1.** **L-2, SC-3**: Option to Add New Languages
- **3.2.** **L-3.:** First Frame Screenshot for Video Files
- **3.3.** **DS-1.:** Support for Windows XP and Windows 7

4. Test strategy and approach

4.1. General approach

The application is designed to assist users in efficiently locating specific files across various directories, supporting a wide range of file types and file systems. Configuration and initial setup will be performed by experienced users, while general users will primarily interact with the search functionality. As a result, usability and security are not within the main scope of this testing effort, focusing instead on core functionality and performance.

4.2. Functional testing levels

Smoke Test: Automated testing across basic search scenarios and directory structures on supported operating systems (Windows and UNIX). Verifies that the application initializes and performs basic file searches without critical issues.

Critical Path Test: Executed manually with detailed checks on essential functions such as directory depth, file type filtering, and display of search results. This level ensures that core user scenarios are fully operational.

Functional Test: Conducted to verify that the application correctly handles all specified functionalities, including file type filtering, display of file attributes, and logging. This test ensures the accuracy and completeness of the core functions as outlined in the requirements

Integration Test: Focuses on validating the application's ability to perform file searches over a network, as well as ensuring that it integrates correctly with different operating systems and file systems. This test verifies smooth operation across networked

environments and platform compatibility.

Performance Test: Manual and automated tests designed to verify the search speed and application responsiveness when handling large data volumes. This ensures compliance with the performance requirement of 500 files per second.

Compatibility Test: Validation across various file systems supported by Windows and UNIX, ensuring that the application performs reliably on all intended platforms.

5. Criteria

Acceptance criteria:
Smoke Test: 100% success rate for all smoke test cases to ensure basic stability.
Critical Path Test: 90% success rate, with 100% of critical and major bugs fixed.
Functional Test: 90% of functional test cases must pass, covering all essential application functions.
Integration Test: Successful interaction with network components and compatibility across various operating systems.
Performance Test: The application must meet or exceed the performance requirement of 500 files per second.
Compatibility Test: The application should operate without issues on all specified operating systems and file systems.

Final requirements coverage by tests should be at least 85%.

Testing start criteria:
Availability of a new build that meets initial configuration, functional, and performance expectations.

Testing pause criteria: Critical Path Test and Functional Test should only begin after 100% success in smoke tests.
The testing process may be paused if, after executing at least 25% of test cases, the failure rate for any test type (e.g., integration or compatibility) reaches or exceeds 50%.

Testing resumption criteria: At least 50% of the bugs identified during the previous iteration must be fixed before resuming testing

Testing finish criteria: More than 80% of the planned test cases for the current iteration across all six test types should be executed.

6. Resources

6.1. **Software:** Two virtual machines with Windows XP and Windows 7 (to match application compatibility requirements).

Delphi 7 IDE for any necessary debugging or adjustments.

File management and network simulation tools for testing file search functionality over networked drives.

6.2. **Hardware:** Two standard workstations with the following specifications: 8GB RAM, Intel i5 or equivalent, 3.2 GHz processor**.**

6.3. **Personnel:** One tester with expertise in functional and compatibility testing (100% workload for the entire project).
Role: Tester.

6.4. **Time:** Two workweeks (80 work hours).

6.5. **Finances**: According to the approved budget.

7. Schedule

25.10-29.10: Requirements analysis and clarification with the client.
1.11-5.11: Requirements finalization and documentation.
8.11-12.11: Initial test planning and creation of test cases.
15.11-19.11:Development of test scripts for automated testing.
22.11-26.11:Review and refinement of test cases and scripts.
29.11-3.12:Execution of smoke tests and defect logging
6.12-10.12:Execution of critical path tests and performance testing
13.12-17.12:Defect fixing and retesting
20.12-22.12:Project finalization, reporting, and discussion with the client

8. Roles and responsibilities

Tester: documentation creation, test-cases execution, participation in code-review.

9. Risk evaluation

9.1. **Performance Risks (high probability)**: If the application does not meet the required search speed (500 files per second) or struggles with network searches, this could impact user experience and overall application effectiveness. Additional testing and optimization may be necessary to ensure performance benchmarks are met.

9.2. **Compatibility Risks (medium probability)**: The application is designed to run on Windows XP and Windows 7. If additional platforms or environments are required by the client, there is a risk of incompatibility, which may increase testing scope and time.

9.3. **Additional Risks Due to Potential Changes (low probability)**: If the client foresees potential changes in requirements or scope, there is a risk of needing to revisit and modify tested functionalities. This could impact timelines and require extra resources to address evolving needs.

10. Documentation

Requirements Documentation. Responsible person — tester, deadline — Nov 5.

Test Plan and Test Cases. Responsible person — tester, creation period — Nov 8 - Nov 19.

Automated Test Scripts. Responsible person — tester, creation period — Nov 15 - Nov 26.

Smoke Test and Critical Path Test Results. Responsible person — tester, creation period — Nov 29 - Dec 10.

Defect Reports. Responsible person — tester, ongoing creation period — Nov 29 - Dec 17.

Performance Test Report. Responsible person — tester, deadline — Dec 10.

Final Test Report. Responsible person — tester, deadline — Dec 22.

11. Metrics

- **Test cases success percentage:**

$$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100\%$$

T^{SP}– percentage of successfully passed test cases,

$T^{SUCCESS}$– quantity of successfully passed test cases,
T^{TOTAL}– total quantity of executed test cases.

Minimally acceptable borders:
Beginning project phase: 10%.
Main project phase: 40%.
Final project phase: 80%

- **Overall defects fixed percentage:**

$$D_{Level}^{FTP} = \frac{D_{Level}^{Closed}}{D_{Level}^{Found}} \cdot 100\%$$

D_{Level}^{FTP}– overall defects fixation percentage by $Level$ during all project lifetime,

D_{Level}^{Closed} - quantity of defects of $Level$ fixed during all project lifetime,

D_{Level}^{Found} - quantity of defects of $Level$ found during all project lifetime.

Minimally acceptable borders:

		Defect severity			
		Minor	Medium	Major	Critical
Project phase	Beginning	10%	40%	50%	80%
	Main	15%	50%	75%	90%
	Final	20%	60%	100%	100%

Ongoing defects fixed percentage:

$$D_{Level}^{FCP} = \frac{D_{Level}^{Closed}}{D_{Level}^{Found}} \cdot 100\%$$

D_{Level}^{FCP} - defects fixation percentage by $Level$ (defects found in the previous build and fixed in the current build),

D_{Level}^{Closed} - quantity of defects of $Level$ fixed in the current build,

D_{Level}^{Found} - quantity of defects of $Level$ found in the previous build.

Minimally acceptable borders:

		Defect severity			
		Minor	Medium	Major	Critical
Project phase	Beginning	60%	60%	60%	60%
	Main	65%	70%	85%	90%
	Final	70%	80%	95%	100%

Stop-factor:

$$S = \begin{cases} Yes, T^E \geq 25\% \;\&\&\; T^{SP} < 50\% \\ No, T^E < 25\% \;||\; T^{SP} \geq 50\% \end{cases}$$

S – decision to pause the testing process,
T^E – current T^E value,
T^{SP} – current T^{SP} value

Test-cases execution percentage:

$$T^E = \frac{T^{Executed}}{T^{Planned}} \cdot 100\%$$

T^E – test-cases execution percentage,
$T^{EXECUTED}$ – quantity of executed test-cases,
$T^{PLANNED}$ – quantity of planned (to execution) test-cases.

Levels (borders):

Minimal: 80%.
Desired: 95%-100%.

Requirements coverage by tests:

$$R^C = \frac{R^{Covered}}{R^{Total}} \cdot 100\%,$$

R^C – requirements coverage by tests (percentage),
$R^{COVERED}$ – quantity of requirements covered with test-cases,
R^{TOTAL} – overall quantity of requirements.

Minimally acceptable borders:

Beginning project phase: 40%.
Main project phase: 60%.
Final project phase: 80% (90%+ recommended).