KML Guard

GIS File Hosting Web Application and API

Derivable Title	**Test Specification**
Work Package	**GIS File Hosting Web Application and API**
Version	**v1.0**
Date	**April 7, 2019**
Status	**Under revision**
Stakeholder/Project Visionary	**Christopher Priebe**
Main Contributors	**Chelsea Marfil** **Anton Carrillo** **Christopher Greer** **Marc Marcelino** **Duong Pham**

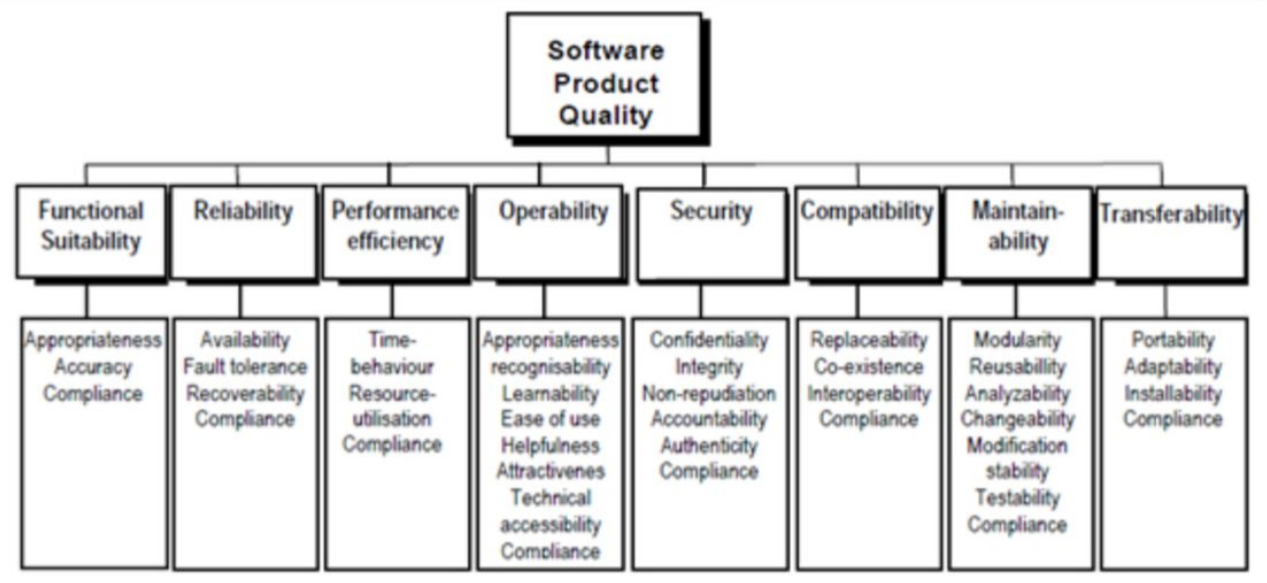
Table of Contents

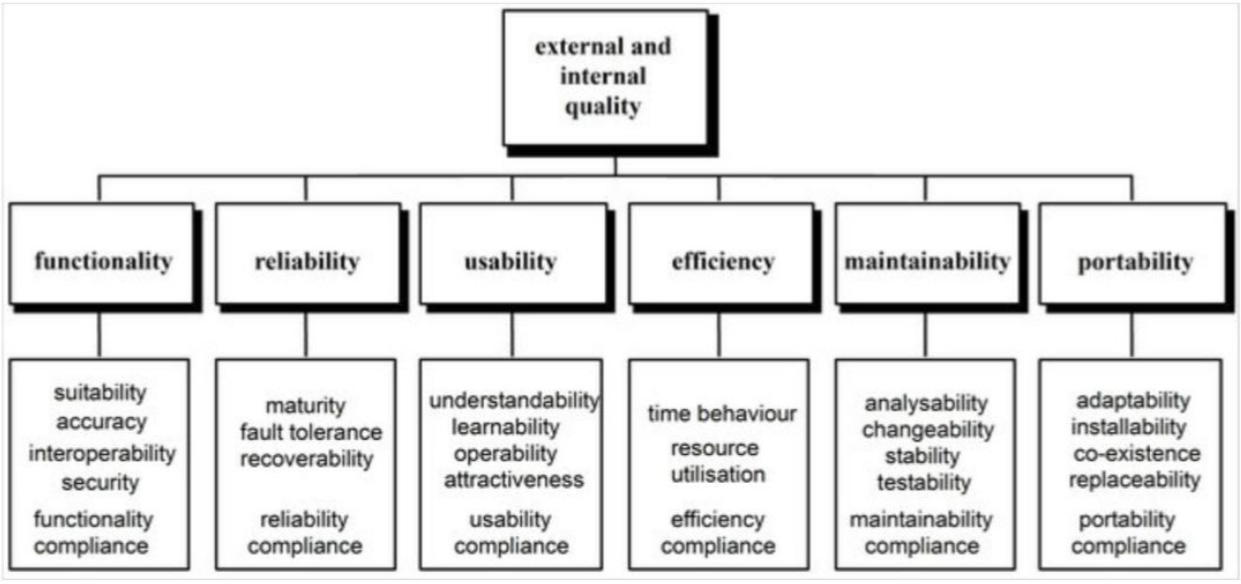
0. Introduction

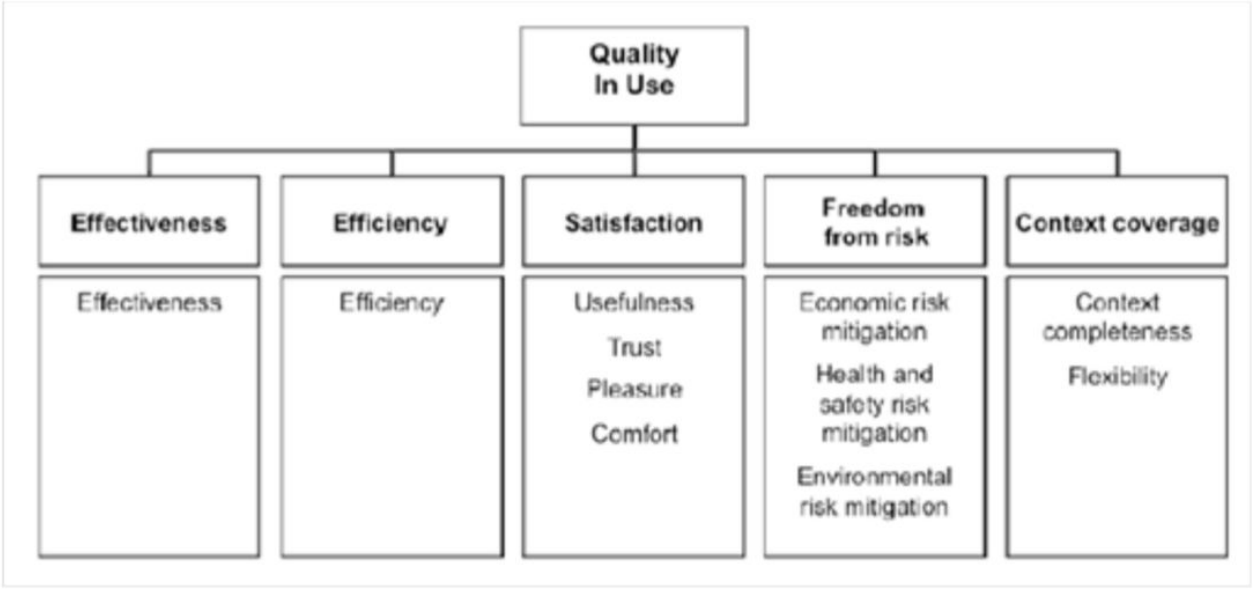
We will test our software and perform both quality control and quality assurance.

In our testing, we will determine whether our software is fit for use by proving incorrectness. Testing will be based on our requirements and design specifications. The following document details how we will test on the following five levels: unit, module, integration, system, and acceptance. Under those five levels we will be testing for some of the quality characteristics from ISO 25010 (screenshots below). For the acceptance level, we will test qualities from the 'Quality in Use' and 'Software Product Quality' diagrams. For the system level, we will test qualities from the 'Software Product Quality' and 'External and Internal Quality' diagrams. For the integration level, we will test qualities from the 'Internal and External Quality' diagram. For the module level, we will test qualities from the 'Internal and External Quality diagram. For the unit level, we will test qualities from the 'Internal and External Quality' diagram.

ISO 25010







Each test case in this document will provide 1. The test level, 2. The quality criterion/attribute from ISO 25010 that will be tested, 3. A description of the test, 4. A reference to the use case being tested 5. Steps the tester will take for testing, and 6. The expected outcome that the system will give after the testing steps have been taken.

Through quality control, we will be reactive by identifying and correcting defects in our software. Through quality assurance, we will be proactive by aiming to prevent defects.

1. Test Level: Acceptance

1.1 Acceptance: Usefulness

Test Level	Acceptance
Quality criterion / attribute	Quality in Use - Usefulness
Description of test	Tester is able to access data and files from the database
Requirements Reference	Use Case 9: Access Files via our API
Steps of the test case	1. Invoke API on software running in application 2. Send HTTP request to browse files
Expected outcome	List of files are shown to the user allow them see the information and interact with them

1.2 Acceptance: Security

Test Level	Acceptance
Quality criterion / attribute	Software Product Quality - Security
Description of test	Test security of logout function. Prevent shared systems from accessing unauthorized resources.
Requirements Reference	Use Case 7: Login
Steps of the test case	1. Tester must be logged into one user or admin account at some point. 2. Log out of that user 3. (Optional) Log into another user 4. Use the browser Back/Forward features to navigate to the previous user's session.
Expected outcome	Tester cannot use the previous user's account.

1.3 Acceptance: Trust

Test Level	Acceptance
Quality criterion / attribute	Quality in Use - Trust
Description of test	Tester is restricted access to specific routes (URL) provided lack of credentials.
Requirements Reference	Use Case 7: Login

Steps of the test case	1. Tester has or has not logged in to a valid user or admin account. 2. User enters a restricted URL in the address bar of a web browser.
Expected outcome	1. Given a URL that is used by a normal user, the traversal succeeds if and only if the tester is logged in with the same credentials required by the specific route entered. 2. Given a URL that is used by an admin, the traversal succeeds if and only if the tester is logged in with admin credentials.

1.4 Acceptance: Compliance

Test Level	**Acceptance**
Quality criterion / attribute	Software Product Quality - Compliance
Description of test	Tester can no longer access specific URLs after his session has expired (time out)
Requirements Reference	Use Case 7: Login
Steps of the test case	1. Tester must be logged into one user or admin account at some point. 2. User' session time has expired 3. Use the browser Back/Forward features to navigate to restricted URLs.
Expected outcome	User will be prompted to re-login

1.5 Acceptance: Functionality

Test Level	**Acceptance**
Quality criterion / attribute	External and Internal Quality - Functionality
Description of test	Ensure that the system only allows the upload of valid KML files.
Requirements Reference	Use Case Scenario 1: Upload File via Our WebUI
Steps of the test case	1. Load the website. 2. Enter valid login credentials on website. 3. Navigate to the upload form on the website. 4. Upload a file of an extension other than KML.
Expected outcome	The user will be notified that the file type is invalid.

2. Test Level: System

2.1 System: Functionality

Test Level	**System**
Quality criterion / attribute	Software Product Quality - Functionality
Description of test	Test the Download File functionality
Requirements Reference	Use Case 2
Steps of the test case	1. Invokes HTTP GET request to server to view files 2. Click on a file 3. Click Download file
Expected outcome	Selected file is downloaded onto the user's computer

2.2 System: Reliability

Test Level	**System**
Quality criterion / attribute	Software Product Quality - Reliability
Description of test	Test the Upload File functionality
Requirements Reference	Use Case 1
Steps of the test case	1. Click on Upload link 2. Fill out upload form 3. Click on upload button 4. Select a KML file from computer 5. Click on Complete button
Expected outcome	Selected file should now be uploaded and show up in the database

2.3 System: Security

Test Level	**System**
Quality criterion / attribute	Software Product Quality - Security
Description of test	Test the login functionality of the web app.
Requirements Reference	Use Case Scenario 7: Login via our WebUI
Steps of the test case	1. Navigate to index page. 2. Enter valid username and password. 3. Click on login button.
Expected outcome	The user should be redirected to the main page.

2.4 System: Accuracy

Test Level	**System**
Quality criterion / attribute	External and Internal Quality - Accuracy
Description of test	Test the delete functionality of the web app.
Requirements Reference	Use Case Scenario 6: Delete File via our WebUI
Steps of the test case	1. Hover over the file(row) that you want delete and click on the delete button.
Expected outcome	The row should be removed.

2.5 System: Usability

Test Level	**System**
Quality criterion / attribute	External and Internal Quality - Usability
Description of test	Test the “Show Files” functionality when a user is logged in
Requirements Reference	N/A
Steps of the test case	1. Navigate to index page. 2. Enter valid username and password. 3. Click on login button.
Expected outcome	System should show all available files

3. Test Level: Integration

3.1 Integration: Functionality

Test Level	Integration
Quality criterion / attribute	Internal and External Quality: Functionality
Description of test	Connectivity between Login and WebUI modules
Requirements Reference	Use Case Scenario 7: Login
Steps of the test case	1. Navigate to index page. 2. Enter valid username and password. 3. Click on login button.
Expected outcome	To be directed to the Login Success

3.2 Integration: Reliability

Test Level	Integration
Quality criterion / attribute	Internal and External Quality: Reliability
Description of test	Connectivity between Upload and WebUI modules
Requirements Reference	Use Case: Upload Files
Steps of the test case	1. Successful log in. 2. Select file to be uploaded 3. Click on upload button.
Expected outcome	Selected file should appear in the list of available files

3.3 Integration: Portability

Test Level	Integration
Quality criterion / attribute	Internal and External Quality: Portability
Description of test	Test compatibility of website in both Android and IOS
Requirements Reference	Use case: Mobile access
Steps of the test case	1. Load website in both Android and IOS devices. 2. Navigate through website on both devices
Expected outcome	Website should work as expected in both devices

3.4 Integration: Accuracy

Test Level	Integration
Quality criterion / attribute	External and Internal Quality - Accuracy
Description of test	Test functionality between “KML Validator” and “View Metadata” modules
Requirements Reference	Use Case Scenario 1: Upload File via Our WebUI
Steps of the test case	1. Load the website. 2. Enter valid login credentials on website. 3. Navigate to the upload form on the website. 4. Upload KML file
Expected outcome	The user will be notified that the file type is valid by showing: “KML Validation: Success”

3.5 Integration: Interoperability

Test Level	Integration
Quality criterion / attribute	External and Internal Quality - Interoperability
Description of test	Each time a user sends a post request to the API, the data is successfully posted to the database.
Requirements Reference	Use Case Scenario 9: Access Files via our API
Steps of the test case	1. Send a post request to the API to create new user, upload a file or download a file. 2. Send a GET request with the returned ID.
Expected outcome	The user should get one result.

4. Test Level: Module

4.1 Module - Main Page: Reliability

Test Level	**Module: Main Page**
Quality criterion / attribute	Internal and External Quality: Reliability
Description of test	The tester should get the same metadata on a file whenever clicking to view it.
Requirements Reference	Use Case 4: View Metadata via our WebUI
Steps of the test case	1. Click on a row in file table to reveal metadata 2. Click same row again to collapse metadata 3. Click on same row once again to reveal metadata
Expected outcome	User should see the same metadata every time

4.2 Module - Database: Stability

Test Level	**Module: Database**
Quality criterion / attribute	Internal and External Quality - Stability
Description of test	MongoDB should output the same JSON file each time the tester queries the database regarding a KML file info
Requirements Reference	N/A
Steps of the test case	1. Successful login 2. Tester queries the database regarding a KML file
Expected outcome	Output (JSON) returned by the database should be the same for each time

4.3 Module - File Delete via API: Testability

Test Level	Module: File Delete via API
Quality criterion / attribute	External and Internal Quality - Testability
Description of test	Test the file deletion functionality using the API.
Requirements Reference	Use Case Scenario 9: Access Files via our API
Steps of the test case	1. Send a DELETE request to /uploads/[upload_id].
Expected outcome	The user should receive a “delete successful” message as a response from the API.

4.4 Module - File Download via API: Functionality

Test Level	Module: File Download via API
Quality criterion / attribute	External and Internal Quality - Functionality
Description of test	Each time a user sends a post request to download a file, a download entry is recorded in the database and the file is returned.
Requirements Reference	Use Case Scenario 9: Access Files via our API
Steps of the test case	1. Send a POST request to /downloads/[upload_id]. 2. Send a GET request to /downloads/[upload_id].
Expected outcome	The user should get a file download as a response from the API.

4.5 Module - KML Validator: Suitability

Test Level	Module: KML Validator
Quality criterion / attribute	External and Internal Quality - Suitability
Description of test	Test for corrupted or invalid KML files
Requirements Reference	N/A
Steps of the test case	1. Upload corrupted KML file to check for validation
Expected outcome	User is notified that although it is a KML file, the file is corrupted

5. Test Level: Unit

5.1 Unit - Login Validator: Security

Test Level	**Unit: Login Validator**
Quality criterion / attribute	External and Internal Quality: Security
Description of test	Make sure user credentials are valid before allowing access. Display error if invalid.
Requirements Reference	Use Case Scenario 7: Login via WebUI from Requirements Specification.
Steps of the test case	1. Type in invalid username (e.g. “invalid”) and password credentials.
	2. Type in invalid password (e.g. “invalid”).
	3. Click Login button.
Expected outcome	Error message prompts user of invalid credentials.

5.2 Unit - File Validator: Functionality - Compliance

Test Level	**Unit: File Validator**
Quality criterion / attribute	External and Internal Quality: Functionality - Compliance
Description of test	Ensure that uploaded files are valid KML files.
Requirements Reference	Use Case Scenario 1: Upload File via Our WebUI from Requirements Specification.
Steps of the test case	1. Click ‘Upload File’ button.
	2. Click ‘Select File’ button.
	3. Select a file with file type that is not .kml (e.g. .txt file).
	2. Click ‘Upload’.
Expected outcome	Error message tells the user that their selected file is an invalid filetype and prompts them to select another file.

5.3 Unit - File Deletion Metadata Update: Functionality - Suitability

Test Level	**Unit: File Deletion Metadata Update**
Quality criterion / attribute	External and Internal Quality: Functionality - Suitability
Description of test	When a file is deleted by a user, make sure that the metadata of the file is updated with the deletion information (delete by / delete date).
Requirements Reference	Use Case Scenario 6: Delete File via our webUI
Steps of the test case	1. Delete a file that is not already deleted via the API. 2. Send a GET request to view the metadata of the deleted file.
Expected outcome	One result that has the delete_by and delete_date fields.

5.4 Unit - KML Parser: Functionality - Accuracy

Test Level	**Unit: KML Parser**
Quality criterion / attribute	External and Internal Quality: Functionality - Accuracy
Description of test	Verify that a file is parsed accurately using the checksum generated by the parser.
Requirements Reference	Use Case Scenario 1: Upload File via Our WebUI
Steps of the test case	1. Upload a KML file to the web server. 2. Download the same file. 3. Verify the md5 hashes of both files.
Expected outcome	The hashes should match.

5.5 Unit - Database Connection: Maintainability - Analyzability

Test Level	**Unit: Database Connection**
Quality criterion / attribute	External and Internal Quality: Maintainability - Analyzability
Description of test	Verify that an error is returned when a request is made to the API and there is an error connecting to the database.
Requirements Reference	Use Case Scenario 9: Access Files via our API
Steps of the test case	1. Send a HTTP request to the API.
Expected outcome	A database connection error is returned in the console.

6. Appendix

6.1 Links to other specifications

Requirements Specification
Design Specification