University of Alaska
Academic Assessment Repository

Quality Assurance Test Plan

Created By:

Brooxie Robinson

Mignon Kramp

Andrea Mejia

Glenn De Guzman

Table of Contents

INTRODUCTION

Purpose

Quality doesn't happen by accident. It must be planned so there is agreement about how quality is measured, when quality checks occur and how corrective actions are determined and implemented. The Quality Assurance Plan is a tool that can help the project deliver the highest possible quality result within committed resources, schedule and budget.

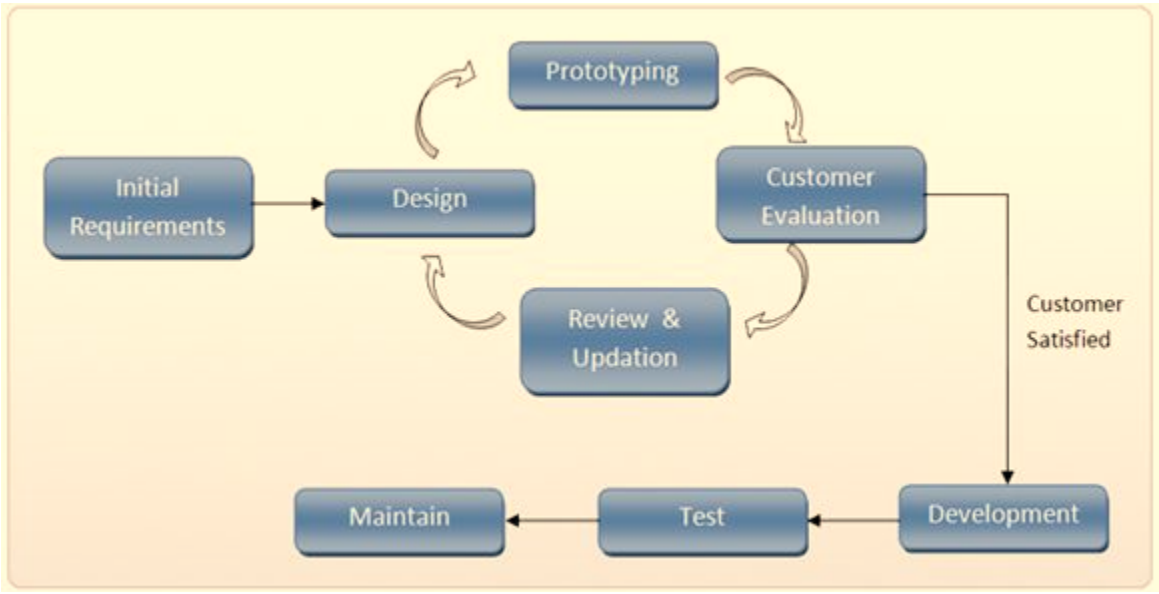
The Quality Assurance Plan describes the strategy and methods the project will deploy to ensure two things:

1. That the project is being managed, developed, and deployed in a sound, reasonable way.
2. That the project's deliverables are of acceptable quality before they are delivered to the project's clients.

Project Overview

This project aims to provide the University of Alaska's Office of Academic Affairs with a Microsoft SharePoint site that will, firstly improve the efficiency of maintaining and regulating a hierarchy of program reports and the program report's supporting artifacts. Secondly it will aid internal communication among faculty, faculty coordinators, and deans. To accomplish this the project team will mix a design to tools lifecycle model with a rapid evolutionary prototype model. These two lifecycle models have been chosen for two main reasons. A. Design to Tools was chosen since Institutional Research has already built the shell of the SharePoint 2013 site and will be maintaining the site after the product has been turned over to the project sponsor. B. The Evolutionary Prototype was chosen since the Project Sponsor had an unclear vision of exactly what she wanted. At each design element the Project Sponsor needs to see what the site is capable of and then send out to the deans of each college for their feedback. The end solution will be a web-based Microsoft SharePoint site hosted on a SharePoint 2013 server.

Rapid Evolutionary Prototyping Model



Project Sponsor

Name	Telephone	Email
Susan Kalina	907-786-1988	Smkalina@uaa.alaska.edu

Team Members

Name	Telephone	Email
Brooxie Robinson	**907-232-6786**	**Blrobinson@alaska.edu**
Mignon Kramp	**907-232-2817**	**Mckramp@alaska.edu**
Andrea Mejia	**907-240-7117**	**Abmejia@alaska.edu**
Glenn De Guzman	**907-227-5602**	**Glenn0403@gmail.com**

PROJECT QUALITY ASSURANCE

Fast Track Project Required Documents

Project Artifacts	Complete
Project Proposal	✔
Initiation Phase Checklist	✔
Project WBS	✔
Project Charter	✔
Business Requirements Document	✔
Project Plan Review Checklist	
Analysis Phase Checklist	
Design Review Checklist	
Conceptual IT Architecture Review Checklist	
Application Architecture Design	
System Architecture Design	
Code Review Checklist	
Implementation Plan Checklist	
QA Test Plan	
Test Planning Checklist	
Deployment Readiness Assessment Checklist	
User Acceptance Sign Off	
Service Level Agreement and Checklist	
Lessons Learned	
Close out Report	

Document References

This section describes the nature, purpose and audience of every document that is produced.

All documents audience will be general audience unless other is stated. We defined the general audience as follows:

- The client
- Team
- Auditors

Software Quality Assurance Plan (SQAP)

Refer to chapter 1

System Requirements Specification (SRS)

- Define the overall requirements of the system covering in a concise and unambiguous way, covering all aspects of the system including:

    1. The system's functionality.
    2. Content and attributes.
    3. Design requirements.

4. Performance constraints.

- Provide a base of common understanding for both the client and the team.
- Provide a framework that can measure against during acceptance testing.

Software Test Plan (TP)

- Explain test philosophies and techniques.
- Describe the derivation of test cases and approaches used to apply the in the code.
- Identify the techniques used for test result collection and bug monitoring.
- Specify the order in which modules are to be tested.

Software Usability Test Plan (SUTP)

- Leverage the quality of the software usability.
- Identify usability defects.

Project Plan (PP)

- Assist the Project Manager in planning all aspects of the project.
- The project plan should be a reference to the team members and identify the current phase of the development.

User Documentation (UD)

- Insure proper use of the software.

Technical User Documentation (TUD)

- Insure system can be properly maintained

Design Notebook (DN)

- Organize a method for documenting all design proposals and decisions.
- Store all relevant information from meetings (i.e. print-outs, sketches, etc).
- Contain results of reviews.

Management

Role Allocation

Roles are allocated to team members as listed below.

Role	Assigned to:
Project Manager	**Brooxie Robinson**
Lead Developer	**Mignon Kramp**
System Administrator	**Glenn De Guzman**
Instructional Publications Designer	**Andrea Mejia**
Project Sponsor	**Susan Kalina**
Faculty Advisor	**Dennis Drinka**
Technical Review Committee	**Volunteers**
System Testing Group	**Volunteers**

User Acceptance Testing Committee	**Volunteers**

Document Allocation

Each team member is responsible for managing one or more documents, as allocated below.

Document	Document Manager
Software Quality Assurance Plan (SQAP)	**Brooxie Robinson**
System Requirement Specifications (SRS)	**Brooxie Robinson**
Software Usability Test Plan (UTP)	**Glenn De Guzman**
Software Test Plan (TP)	**Mignon Kramp**
User Documentation (UD)	**Andrea Mejia**
Technical User Documentation (TUD)	**Mignon Kramp**
Project Plan (PP)	**Samantha Veilleux (past member)**

Team Members Responsibilities

Project Manager

- Manages and updates the project plan.
- Informs team members of the project plan at team meetings.
- Has an overview over what milestones have been met and which ones are next.
- Informs team member of progress.
- Delegates tasks to team members.
- Ensures everyone is notified of meetings and keeps track of who will be present.
- Creates agenda for meetings and e-mails them to team members, updates agenda in case of eventual inputs by team.
- Maintains contact with Project Sponsor.
- Tries to maintain a fair and balanced workload for all team members.

Lead Developer

- Is responsible for writing the Software Test Plan (TP).
- Manages the process of testing by delegating tasks to team members.
- Documents all tests that are carried out.

System Administrator

- Organizes usability test, this includes gathering test group and preparation.
- Classifies eventual usability errors and together with lead developer suggests improvement to system design.

Instructional Publications Designer

- Manages the proofreading of all documents.
- Creates user documentation to ensure users satisfaction

Quality Assurance Mechanisms

System Requirements

Functional Requirements

The system will:

- be built in SharePoint 2013
- have a main website with a drop down to the Academic Assessments page
- have a folder for the Office of Academic Assessments named Academic Affairs
- have individual subsites for each college
  - College of Arts and Sciences (CAS)
  - College of Business and Public Policy (CBPP)
  - College of Education (COE)
  - College of Engineering (COEng)
  - College of Health (COH)
  - Community & Technical College (CTC)
- have a subfolder for each of the 200 programs under the colleges the program belongs to
- have a subfolder for the assessment plan under each program
- have a subfolder for the academic year under each program
- have a subfolder for the program report under each academic year subfolder
- have a subfolder for artifacts under each academic year subfolder
- have a subfolder with the complied program reports for each college
- sort data based on programs
- have a workflow process specified for each college if desired by Project Sponsor
- have a dashboard on the home page and one for each college
- conform to user permissions as outlined by the Project Sponsor
- conform to user groups as outlined by the Project Sponsor
- conform to shared documents as outlined by the Project Sponsor
- allow comments from Deans and Campus Directors on program reports
- authenticate users through the User's UAA email and password
- only display data to a user based on his/her permissions group
- record who uploaded the document
- record when the document is uploaded by system timestamp
- record the file type of the document that is being uploaded
- support uploading different types of files
  - the following files types will be:
    - Adobe Portable Document Format (PDF) documents
    - Microsoft Word documents
    - Microsoft PowerPoint documents
    - Microsoft Excel documents
    - Joint Photographic Experts Group (JPEG) files
    - Tagged Image File Format (TIFF) files
    - Windows Bitmap (BMP) files
    - Portable Network Graphics (PNG) files

Nonfunctional Requirements

The system must:

- be secure
- cannot share documents with external users
- have appropriate permission levels, least privileges
- be scalable
- be available 24 hours a day, 7 days a week
- be maintainable
- have a restore point
- be easy to navigate
- be user friendly

Nice to Have Features

As time allows:

- each SharePoint user can have customizable dashboard
- the system will generate reports for the OAA
- automatic email alerts based on calendar events will be set up

The System Will Not

The system will not:

- upload unsupported files
- support unique requirements from staff by each individual college
- have a customized CSS file

Change Management

Purpose

The purpose of Change Management is to allow changes on a project to happen. Change Management is to coordinate and control all changes to the project and to minimize adverse impacts of those changes to the project. If a user, customer, project manager, instructor, or sponsor would like to make a certain change, the Change Management Form must be filled out and then will be reviewed. The change is determined by evaluating the points below:

- The change must satisfy a business objective
- The change must be within the overall scope of the project
- The change must fit the budget, resource, and schedule constraints
- The change must not have a negative effect on other functionality of the system

Change Management Responsibilities

The Change Management Plan including the form and log will be monitored by the Project Manager. The Project Manager will be in charge of completing the Change Management Request Form any time a change is requested. The Project Manager will also be responsible for recording all change request in the Change Management Log.

Change Management Process

For a change to take place, the Change Management Request form must be filled out and reviewed. Signatures will need to be collected on the Change Management Request form.

Sponsor will:
Step 1: request a change in writing (email is okay). Sponsor will include reasoning for the change.

Project Manager will:
Step 2: Write a description of each change.
Step 3: Document the technical and functional impact of the change.
Step 4: Meet with Project Sponsors and Faculty Liaison within one week of receiving the change request to discuss the requested change and the analysis of the change.
Step 5: Determine and justify the priority of all agreed upon change requests.
Step 5: Schedule when the change will take place.
Step 6: Document who will be informed of the change, and when they were informed.
Step 7: Document all change requests.
Step 8: Obtain all necessary signatures.

The key activities for this process are:

- Accept changes
- Prioritize and classify changes
- Coordinate change impact assessment
- Coordinate approval of changes
- Coordinate scheduling of changes
- Coordinate implementation of changes

Project Change Request Form		
Change Request Number:	Requested By:	Date Requested:
Change Request Description		
Change Description:		
Priority: e Top | e High | e Medium | e Low		
Change Impact Analysis		
Impact on Project Requirements: e In Scope | e Out of Scope		
Impact on Project Risk:		
Impact on Project Budget:		
Impact on Project Configuration: (Technical & Functional)		
Impact on Project Schedule:		
Alternatives:		
Recommendation:		
Change Request Resolution		
Change Request Decision:		Decision Date:
e Approved | e Conditional Approval | e Denied		
Decision Made By:		
e Project Manager | e Project Sponsor | e Executive Sponsor | e Other		
Signatures		
Project Manager:		Date:
Project Sponsor:		Date:
Faculty Liaison:		Date:

Defect Tracking

Purpose

Defect tracking is the process of finding defects in the product from beginning to completion by testing, inspection, or recording feedback from customers. The defects are logged and corrections to the product that fix the defects are made.

Defect tracking is an important process in software engineering as complex systems can have hundreds of defects. One of the challenging factors is managing, evaluating and prioritizing these defects. The number of defects can be multiplied over a period of time and to effectively manage them, defect tracking system is used to make the job easier.

Defect Tracking Parameters

- Defect Id
- Defect Description
- Source of Defect
- Date Detected
- Date Resolved
- Severity of Defect
- Effort to Fix (in hours)
- Resolution Reporting

ADD DEFECT TRACKING LOG

Defect Tracking Report Form

Defect ID:	Date Detected:
Severity (1-5):	Submitted by:
Description:	
Status of Defect:	Date Corrected:
Corrected by:	Hours required to correct defect:
Resolution:	
Cause of Defect:	

Severity	Describes the degree of impact that a defect has on the operation of the application
1 – Critical	Critical loss of function. The defect results in system crashes, the failure of a key subsystem or module, a corruption or loss of data, or a sever memory leak.
2 – Major	Major loss of function. The defect results in a failure of the system, subsystem, or module, but the defect does not result in the corruption or loss of significant data.
3 – Moderate	Moderate loss of function. The defect does not result in a failure of the system, subsystem, or module, but the defect may cause the system to display data incorrectly, incompletely, or inconsistently.
4 – Minor	Minor loss of function, or another problem where a workaround is present. There are no data integrity issues.
5 - Usability	The defect is related to the system usability, is the result of non-conformance to a standard, or is related to the aesthetics of the system. There is no loss of system function.

TESTING

Test Objectives

The quality objectives of testing the University of Alaska Academic Assessment Repository application are to ensure complete validation of the business and software requirements:

- Verify software requirements are complete and accurate
- Perform detailed test planning
- Identify testing standards and procedures that will be used on the project
- Prepare and document test scenarios and test cases
- Regression testing to validate that unchanged functionality has not been affected by changes
- Manage defect tracking process
- Provide test metrics/testing summary reports
- Ensure the application is certified for release into the University of Minnesota production environment
- Schedule Go/No Go meeting
- Require sign-offs from all stakeholders

Testing Goals

The goals in testing this application include validating the quality, usability, reliability and performance of the application. Testing will be performed from a black-box approach, not based on any knowledge of internal design or code. Tests will be designed around requirements and functionality.

Another goal is to make the tests repeatable for use in regression testing during the project lifecycle, and for future application upgrades. A part of the approach in testing will be to initially perform a 'Smoke Test' upon delivery of the application for testing. Smoke Testing is typically an initial testing effort to determine if a new software version is performing well enough to accept it for a major testing effort. For example, if the new software is crashing frequently, or corrupting databases, the software is not in a

stable enough condition to warrant further testing in its current state. This testing will be performed first. After acceptance of the build delivered for system testing, functions will be tested based upon the designated priority (critical, high, medium, low).

Quality

Quality software is reasonably bug-free, meets requirements and/or expectations, and is maintainable. Testing the quality of the application will be a two-step process of independent verification and validation. First, a verification process will be undertaken involving reviews and meetings to evaluate documents, plans, requirements, and specifications to ensure that the end result of the application is testable, and that requirements are covered. The overall goal is to ensure that the requirements are clear, complete, detailed, cohesive, attainable, and testable. In addition, this helps to ensure that requirements are agreed to by all stakeholders.

Second, actual testing will be performed to ensure that the requirements are met. The standard by which the application meets quality expectations will be based upon the requirements test matrix, use cases and test cases to ensure test case coverage of the requirements. This testing process will also help to ensure the utility of the application – i.e., the design's functionality and "does the application do what the users need?"

Reliability

Reliability is both the consistency and repeatability of the application. A large part of testing an application involves validating its reliability in its functions, data, and system availability. To ensure reliability, the test approach will include positive and negative (break-it) functional tests. In addition, to ensure reliability throughout the iterative software development cycle, regression tests will be performed on all iterations of the application.

Testing Environment

Client Requirements

- Mixed browser supported (Internet Explorer, Edge, Chrome, Firefox).
- Client platform: PC and Macintosh

Testing Platform

- Desktop PC – the application supports all A-Grade browsers for Windows and Mac operating systems , as defined by Yahoo!'s Graded Browser Support standards. http://developer.yahoo.com/yui/articles/gbs/ Windows 2000/IE6 may be excluded.

Requirements for Test

The listings below identifies those items (use cases, functional requirements, non-functional requirements) that have been identified as targets for testing. This list represents *what* will be tested.

Data and Database Integrity Testing

None.

Function Testing

Functional testing focuses on the functional requirements of the software and is performed to confirm that the application operates accurately according to the documented specifications and requirements, and to ensure that interfaces to external systems are properly working.

The system will sort data based on programs
The system will conform to user permissions as outlined by the Project Sponsor
The system will conform to user groups as outlined by the Project Sponsor
The system will conform to shared documents as outlined by the Project Sponsor
The system will allow comments from Deans and Campus Directors on program reports
The system will authenticate users through the User's UAA email and password
The system will only display data to a user based on his/her permissions group
The system will record who uploaded the document
The system will record when the document is uploaded by system timestamp
The system will record the file type of the document that is being uploaded

User Interface Testing

Verify ease of navigation through set of screens

Performance Testing

Verify Admin query is ……

Verify the program report count is being updated correctly (addition or subtraction) for each college's dashboard.

Verify calendar event alert is sent out by email to recipients on the day the event occurs on the calendar

Load Testing

Verify system response when loaded with XXX logged on staff and faculty.
Verify upload time of a document when…….

Stress Testing

None.

Volume Testing

None.

Security and Access Control Testing

Verify Logon from a local PC
Verify Logon from a remote PC
Verify Logon security through UAA user name and password mechanisms

Fallover/ Recovery Testing

None.

Configuration Testing

None.

Destructive Testing

Destructive testing focuses on the error detection and error prevention areas of the product. This testing is exercised in an attempt to anticipate conditions where a user may encounter errors. Destructive testing is less structured than other testing phases and is determined by individual testers.

USER ACCEPTANCE TEST PLAN

Definition

The overall purpose of testing is to ensure the University of Alaska Academic Assessment Repository application performs at an acceptable level for the customer. This section outlines the detailed plan for user acceptance testing of this application.

This test plan will be used to record the customer's sign off of the documented scenarios. Detailed test scripts/cases have been developed and will be used to record the results of user testing. This document is a high level guide, and is not intended as a replacement for any specific user acceptance testing procedures that individual areas might have.

Testing Requirements

- Testing will take place in {insert location}. Some testers may choose to perform some testing from their regular workstations where it is possible. Test results must still be coordinated with others.
- UAT will take place beginning on {insert date}.
- Identified testing participants will receive instructions prior to the start of testing.
- Identified testing participants will perform the equivalent of their normal business function in the upgraded environment.
- Test scripts/cases and scenarios will be prepared prior to the start of UAT.
- Test participants will conduct the tests and document results.
- Defects will be entered into Test Director and tracked by the Test Lead.

Testers/Participants

Testing participants should include representatives from all areas involved in the application. There are benefits to including representatives from across all areas to validate the systems functions before the upgrade goes live in production.

The best candidates for UAT are:

- Staff directly impacted by the upcoming system and business process changes.
- Frequent users of the application and functions planned in test scripts/cases.
- Individuals with a sound understanding of business processes in the areas they represent.

- Individuals with the necessary time to commit to this endeavor.
- Willing to experiment (to try various methods to see what works and what doesn't work).
- Patient and have a tolerance for ambiguity.

Tester Name	Department/Area Representing	Area of Testing Focus

Testing Schedule

All upgraded functionality and test data will be migrated to the test environment prior to the start of user acceptance testing.

Activity	Lead Responsibility	Date
Identify and select test users for UAT		
Develop test scenarios and scripts/cases		
Validate participants availability for testing		
Review scenarios/scripts for accuracy, completeness and sequence (confirm test data is correct)		
Ensure UAT desktops configured for testing		
UAT environment validation		
Testing by UAT participants		

Assumptions

- The Business team has reviewed and accepted functionality identified in the business requirements and software requirements documents.
- Project change control process in place to manage requirements.
- Code walkthroughs/reviews will be completed by the development team.
- Unit testing will be completed by the development team prior to release to the test team.
- Testers will test what is documented in the requirements.
- The test team will have a separate test environment to perform testing.
- All changes to requirements will be communicated to the test team.
- Resources identified in this plan are available to test the application and resolve defects and address issues as they are raised by the test team.
- That the delivery of the product to production contains all setup, etc., that is necessary for optimum performance in the production site.
- Project sponsors, business and technical, will provide actionable guidance on defect prioritization and resolution.
- The UAT environment will be available and desktops will be available to perform testing.

Risks

- Scope creep (last minute addition of new requirements) impacts deadlines for development team and test team.

- Aggressive target date increases the risk of defects being migrated to production. If development timelines are not met, this will directly impact the testing timelines.
- Key resources have completing priorities making availability less than scheduled.
- Any downtime of the test system will significantly impact the testing cycle.

Unit Testing – definition, goal, responsibility, expectations, resources required, acceptance criteria

Technical Review -

System Testing – definition, goal, responsibilities, (info on volunteers)