

Beulah Works LLC

**Software Quality Assurance Plan
(SQAP)
Version 1.3**

UML Sequence Diagram File Generator

February 6, 2019

Isis Curiel, Bruno Hnatusko III, Brayden Mccoy,

Dhyey Patel, Jesse Primiani, Jacob Taylor,

and Syed Arshiyah Ali Zaidi

Approvals:

<u>Title</u>	<u>Signature</u>	<u>Date</u>
QA Manager/Author	Bruno Hnatusko III	02/15/2019
Project Manager	Isis Curiel	02/15/2019

Revision History

Date	Revision	Description	Author
01/31/2019	0.1	Created skeleton SQAP document using IEEE 730-2002 standards.	Jesse Primiani
02/06/2019	0.12	Created rough draft SQAP document based on skeleton SQAP document, began cover page.	Bruno Hnatusko III
02/07/2019	0.13	Fixed grammar involving sentences ending in words with quotations.	Bruno Hnatusko III
02/08/2019	0.14	Edited unknown names.	Isis Curiel
02/09/2019	0.15	Updated revision history format to a table format and formatted table of contents.	Bruno Hnatusko III
02/11/2019	0.16	Updated Training section.	Jesse Primiani
02/11/2019	0.17	Removed section about a Change Control Board, as we will not have one.	Bruno Hnatusko III
2/12/2019	0.18	Formatting & spelling corrections	Brayden McCoy
2/13/2019	0.19	Made final revisions	Isis Curiel
2/14/2019	0.20	Added page numbers	Jesse Primiani
2/14/2019	1.0		Bruno Hnatusko III
4/26/2019	1.1	Updated section 9 to mention JUnit4.	Bruno Hnatusko III
4/26/2019	1.2	Mentioned test found in the STP	Jesse Primiani
4/26/2019	1.3	Updated Section 7	Jacob Taylor

Table of Contents

1. Purpose	Pg. 4
2. Referenced Documents	Pg. 4
3. Management	Pg. 4
3.1. Organization	
3.2. Tasks	
3.3. Responsibilities	
3.4. QA Estimated Resources	
4. Documentation	Pg. 5
4.1. Purpose	
4.2. Minimum Documentation Requirements	
4.3. Other	
5. Standards	Pg. 6
5.1. Purpose	
5.2. Content	
6. Reviews	Pg. 7
6.1. Purpose	
6.2. Minimum Requirements	
6.3. Other Reviews and Audits	
7. Test	Pg. 9
7.1. Unit Test	
7.2. System Test	
7.3. Performance Test	
7.4. User Acceptance Test	
8. Problem Reporting and Corrective Action	Pg. 10
9. Tools, Techniques, and Methodology	Pg. 10
10. Media Control	Pg. 10
11. Supplier Control	Pg. 11
12. Records Collection, Maintenance, and Retention	Pg. 11
13. Training	Pg. 11
14. Risk Management	Pg. 12
15. Glossary	Pg. 12
16. SQAP Change Procedure and History	Pg. 12

1 Purpose

This document describes the means by which the UML Sequence Diagram File Generator project will produce and maintain a high-quality product. It is also intended to describe mechanisms for improving the quality assurance process itself. “Quality” is defined as the degree to which the application satisfies its requirements. The scope of this document comprises the artifacts of all releases.

2 Referenced Documents

See section 4.2.

3 Management

3.1 Organization

This section assumes the description of the organizational structure of the UML Sequence Diagram File Generator project in 4.2 of the SPMP.

This document will refer to the quality work of developers as “internal”. In addition, for the first three iterations of UML Sequence Diagram File Generator, a quality assurance engineer will be designated who will be responsible to the manager of QA. The QA leader will take the lead for project-wide quality issues. QA tasks will be referred to as “external”.

3.2 Tasks

External QA tasks shall include, for all iterations:

- Maintaining this document.
- Documenting the quality of the evolving product and associated artifacts.
- Managing external review meetings.
- Ensuring that verification takes place and logging verification.
- Preparing for and attending all inspections.
- Post-unit testing as per the Software Test Documentation.
- Engaging in activities designed to improve the quality assurance process itself.
- Keeping the project leader apprised of QA progress through weekly written reports.
- Carrying out the audits specified in Section 6 of this document.
- Providing the development team with feedback from the activities of QA.
- Assign defect repair to software engineers.
- Identifying methods and tools for collecting and maintaining metrics.

Internal QA tasks shall include, for all iterations:

- Each team member responsible for the quality of his or her work, as defined in this document.
- Maintaining an issue database.
- Collecting the metrics designate by this and other project documents.
- Carrying out the reviews and inspections specified in Section 6 of this document.

3.3 Responsibilities

It is the quality assurance leader's responsibility to see to it that the tasks in Section 3.2 are performed and to ensure that the prescriptions in this document are followed, including scheduling the reviews specified. For the first three iterations, the QA leader will perform all of the quality control (QC) functions.

The project leader will be responsible for ensuring that inspections and unit tests are performed. The schedules are to be placed in the SPMP.

The leaders designated in the SPMP are responsible for the quality of their respective areas (requirements, design, etc.). They shall ensure that the designated metrics are collected and that the quality self-assessments as outlined in Section 5.2 of this document are conducted.

Each member of the UML Sequence Diagram File Generator development team is responsible for quality. This includes testing individual methods and combinations of methods in a class ("unit testing").

3.4 QA Estimated Resources

The estimated resources required for QA on UML Sequence Diagram File Generator are as follows:

- One engineer working half time for the first third of the project.
- One engineer working full time for the second third of the project.
- One engineer working full time for the last third of the project.
- An additional engineer working half time for the last third of the project.

4 Documentation

4.1 Purpose

The purpose of this section is to identify the documentation that will be used to ensure quality.

4.2 Minimum Documentation Requirements

The following documents will be produced:

- Software Quality Assurance Plan (SQAP; this document).
- Software Configuration Management Plan (SCMP).
- Software Project Management Plan (SPMP).
- Software Requirements Specifications (SRS).
- Software Design Document (SDD).
- Software Test Plan and the test documentation it refers to (STP).
- Software Verification and Validation Plan (SVVP).
- User Manual.

In addition to these documents, the Java source code will utilize Javadoc to generate package-, class-, and function-level documentation.

4.3 Other

– Intentionally left blank.

5 Standards, Practices, Conventions, and Metrics

5.1 Purpose

This section describes the standards, practices, conventions, and metrics to be used for the UML Sequence Diagram File Generator project. These are intended not only to ensure quality of the UML Sequence Diagram File Generator product but also to obtain quantitative metric data on the SQA process itself. These data are to be used to help elevate the CMMI level of Beulah Works LLC.

5.2 Content

Standards:

The IEEE documentation standards as of July 1, 2004, with appropriate modifications, are to be used for all documentation. The standards for Javadoc commenting will be followed as found at <http://java.sun.com/j2se/javadoc/writingdoccomments/index.html> and <http://java.sun.com/j2se/javadoc/writingapispecs/index.html>. Documentation standards or templates developed by the company may supersede these at the discretion of management.

Unified Modeling Language standards, as specified in <https://www.omg.org/spec/UML/2.5.1> shall be used in this project.

Refer to the Conventions section below for additional standards.

Practices:

1. Because delaying quality is expensive, engineers are strongly encouraged to apply quality precepts while working, rather than as an afterthought. This is referred to in the company as “internal quality.” It includes all unit testing.
2. All project artifacts are inspected and are made easily available to the team once released by the developer.
3. All project artifacts are placed under configuration management, where the contents can be seen by anyone in the team at any time (see the Software Configuration Management Plan for details).
4. The development process is to be reviewed at least once for improvement, and the written results forwarded to the software engineering laboratory (see Section 6.2.10).

Conventions:

Where feasible, writing conventions should conform to the suggestions in Writing for Computer Science: The Art of Effective Communication by Justin Zobel (Springer Verlag).

The coding conventions as found at <http://java.sun.com/docs/codeconv/html/CodeConventions.doc.html> will be followed.

Metrics:

For every process and document, metrics shall include the following:

1. Time spent by individuals on preparation and review.
2. Number of defects per unit (e.g. lines of code), classified per Section 8 of this document.
3. Quality self-assessment of the QA process and performance on a scale of 1 through 10, approximately in a bell-shaped distribution; self-assessment scores will not be used for the evaluation of personnel by management; failure to produce them, however, may negatively affect the evaluation of an engineer by management.
The standard for defect classification is given in Section 8 of this document.

Quality Goals:

The quality goals for the UML Sequence Diagram File Generator project are as follows, measured in terms of defects detected within two months of delivery:

- No known “critical” or “serious” defects remain in any delivered artifact.
- Requirements: No more than one “medium”, and no more than three “trivial” defective detailed requirements per 100 lines.
- Design: No more than one “medium” defect per five diagrams. A diagram is any figure that uses about a page of easily legible parts.
- Pseudocode: No more than two “medium” defects per 1000 lines.
- Code: No more than two “medium” defects per KLoC (1000 lines of non-commented code).

6 Reviews and Audits

6.1 Purpose

The purpose of reviews and audits is to continually focus engineers' attention on the quality of the application as it develops. Reviews affect this in a scheduled and thorough manner. Audits do so on the basis of random sampling with short notice.

6.2 Minimum Requirements

6.2.1 Software Requirements Reviews

These are walkthroughs of all proposed requirements in the presence of the entire team and at least one responsible customer representative. They will be led by the requirements leader, who will determine their frequency and scope.

6.2.1A Software Requirements Inspections

After they have been reviewed, all requirements will be inspected. Requirements sections must be completed and signed within a week of beginning the design.

6.2.2 Architecture Design Reviews

This is a review of alternative architectures with the entire team. The review will be led by the design leader in a frequency to be determined. The team will provide feedback, which will be reflected in the final design.

6.2.2A Architecture Design Inspections

After they have been reviewed, architectures will be inspected. Architecture sections must be completed and signed off within a week of beginning detailed design.

6.2.3 Detailed Design Reviews

These are reviews of all proposed detailed designs in the presence of the entire development team. They will be led by the design leader, who will determine their frequency and scope, but at least one design review will be conducted per iteration. If possible, the architecture will be decomposed into detailed designs of its parts, and these will undergo separate detailed design reviews.

6.2.3A Detailed Design Inspections

After they have been reviewed, detailed designs will be inspected. Detailed design sections must be completed and signed off within a week of beginning implementation.

6.2.3 Test Plan Reviews

These are reviews of all proposed test plans in the presence of the entire team. They will be led by the QA leader, who will determine their frequency and scope. The test plan will be decomposed into parts, and these will undergo separate reviews.

6.2.3A Test Plan Inspections

After they have been reviewed, test plans will be inspected. Test plan sections must be completed and signed off within a week of beginning testing.

6.2.4 Verification and Validation Plan Review

V&V is to be conducted by the development team alongside QA. The QA engineer will review the SVV plan prior to its execution.

6.2.5 Functional Audits

The QA leader shall be responsible for auditing the product relative to the SRS. The audit will follow company release procedures guidelines.

6.2.6 Physical Audits

Prior to each delivery, the QA leader is responsible for checking that the physical software and its documentation designated for delivery are complete and the correct version.

6.2.7 In-Process Audits

Project personnel should expect random audits of their work. These will consist of visits to the work site by individuals or teams designated by division management. A day's notice shall usually be given for all visits, but audits without notice shall take place as well. The subject of these audits will be the current work of teams and individuals that has been allocated to the project. All project artifacts will be made freely available to all team members and auditors at all times. It will be organized in a clear, standard fashion, so that audits will be possible without any notice.

6.2.8 Managerial Review

The UML Sequence Diagram File Generator project shall be reviewed by the VP for Engineering during the first week of every month; exceptions are at the discretion of the VP for Engineering. It is the project leader's responsibility to schedule this review and provide the appropriate documentation and software demonstrations.

6.2.9 SCMP Review

The QA leader shall review the status of configuration management on a monthly basis in a manner independent of the procedures specified in the SCMP.

6.2.10 Post-Implementation Review

The UML Sequence Diagram File Generator team shall conduct post-implementation reviews to provide data for future projects. These will include reviews of the project phase just completed and reviews of the QA process itself. The QA team or QA leader shall file a process improvement report for every phase, and for the QA process itself, with the manager of the software engineering laboratory.

6.3 Other Reviews and Audits

– Intentionally left blank.

7 Test

The responsibilities for testing were described in Section 3.3. See the STP for test information.

7.1 Unit Test

See section 3 of the STP for test information.

7.2 System Test

See section 4 of the STP for test information.

7.3 Performance Test

See section 5 of the STP for test information.

7.4 User Acceptance Test

See section 6 of the STP for test information.

8 Problem Reporting and Corrective Action

The team will email one another to communicate bugs to each other.

The values for severity are as follows:

- Critical: Causes the application to crash with significant frequency.
- Serious: Causes at least one documented requirement to be unmet.
- Trivial: Could be allowed to stand ad infinitum without impeding the user from exercising a required feature.
- Medium: Is neither serious nor trivial.
- The documentation defect types are as follows:
 - Incorrect
 - Missing Material
 - Unclear
 - Ambiguous
 - Incomplete
 - Redundant (within or between documents)
 - Contradictory
 - Obsolete
- The code and pseudocode defect types are as follows:
 - Syntax
 - Logic
 - Data (i.e. allows a wrong variable value)
 - Insecure (allows unacceptable security breach)
- The workflow of a defect status is:
 - Open: defect found by tester.
 - Assigned: defect assigned to an engineer.
 - Corrected: defect fixed by engineer.
 - Closed: defect closed by tester.

If the defect is reopened, the defect will move from the Corrected state to an Open state.

The QA leader will create and the QA team will maintain a database of problem reports that describe the deficiencies, discrepancies, and anomalies for UML Sequence Diagram File Generator. They will ensure that defects are consistently recorded on this form and that they are routed and repaired in a consistent manner. Problem reports shall be routed in accordance with the SCMP. Full traceability as to their effects and status shall be maintained, including after they are repaired.

9 Tools, Techniques, and Methodologies

The following tools will be used to develop the UML Sequence Diagram File Generator. For documentation we will use Google Suite. For developing the eclipse IDE will be used and the code will be written in Java. JUnit4 will be used for unit testing. For creating the UML diagrams LucidChart will be used. LucidChart will also be used to test the output.

10 Media Control

The SQA team verifies that the software media are built and configured per the SCMP and that authorized changes have been installed and tested. In addition, the SQA team verifies that the software media are duplicated using only the procedures identified in the SCMP. The SQA audit reports for media control are intended as evidence that QA procedures have been followed. All backup media will be stored off-site as described in the SCMP.

11 Supplier Control

Beulah Works LLC verifies all commercial third-party products provided by the suppliers during incoming inspection by reviewing the packing slips that identify the products and their version numbers. The QA manager is responsible for ensuring that all third-party software and hardware meets the expected requirements. The products will be validated by the QA manager through installation and acceptance tests. A QA representative will be responsible for testing all new versions. He will also be responsible for the relationship with the external vendor.

12 Records Collection, Maintenance, and Retention

The SQA records collected and archived shall include the following:

- Task Reports
- Anomaly reports not handled by the regular problem reporting mechanism
- Memos, including recommendations to responsible parties
- Logbooks of SQA activities
- Audit reports
- Signed-off checklists from reviews and audits
- Minutes of inspections
- Metrics for the QA process itself

Besides verifying the archive procedures specified in the SCMP, SQA shall separately archive its own records at least once a week. These records are retained throughout the operation and maintenance phase.

13 Training

Training will be provided by the class entitled “Software Engineering” at Purdue Northwest, alongside other classes offered there.

14 Risk Management

SQA team members are encouraged to identify risks as early as possible and direct them to the project leader. The procedures for risk management are specified in Section 5.4 of the SPMP.

15 Glossary

This section intentionally left blank.

16 SQAP Change Procedure and History

See the Title page.