

iBudget

Software Quality Assurance Plan

Part 1 of 2 *developed by Vanya D.*

Approvals: L. Assayah _____ date: _____
C. Leung _____ date: _____
Q. Pham _____ date: _____
V. Velev _____ date: _____
J. Reimels _____ date: _____

This SQAP follows the IEEE standard 730-1989.

1. Purpose

The purpose of this plan is to define the XYZ's Software Quality Assurance Plan (SQAP). It also specifies the standards and procedures used in carrying out all software quality assurance (SQA) activities. In addition it defines the tools, techniques, and methodologies to support SQA activities and SQA reporting.

The goal of the SQA program is to verify that all deliverable software and accompanying documentation meet the technical requirements. The SQA procedures defined in this document will be used to examine both software and documentation to determine compliance and ensure the quality during all phases of the project, namely the design, development, testing and validation phases of the project.

2. Referenced documents

SQAP: software quality assurance plan (this document)
SCMP: software configuration management plan
SPMP: software project management plan
SRS: software requirements specifications

SDD: software design document
STD: software test documentation
User's manual
Maintenance plan

3. Management

3.1 Organization

Project Management/Group Lead is responsible for:

- Implementing the quality program.
- Reviewing and approving the XYZ's SQA Plan.
- Resolving and following-up on any quality issues raised by SQA.
- Developing and maintaining documents such as the Program Management Plan, Test Plans, and this SQA Plan.

Software Design/Development is responsible for:

- Reviewing and commenting on the XYZ's SQA Plan.
- Implementing the quality program in accordance with this SQA Plan.
- Resolving and following-up on any quality issues raised by SQA related to software design and development.
- Identifying, implementing, and evaluating the quality factors to be implemented in the software.

Once in the testing phase, cross-developer testing will take place.

3.2 Tasks

QA tasks shall include

- documentation
- review meetings
- verification (including inspections) of
 - Software Requirements Analysis Process
 - Software Design Process
 - Software Implementation and Unit Testing
 - Evaluate Configuration Management Process
- validation (mostly testing)
- identify standards and guidelines
- activities designed to improve the quality assurance process itself

These tasks are detailed in this document.

3.3 Responsibilities

The quality assurance leader is responsible for the completions of tasks in 3.2, and to ensure that the prescriptions in this document are followed, including scheduling the reviews specified. Vanya Dineva (*or anybody else*) was chosen as the quality assurance leader for the XYZ project.

While each team member is responsible for the quality of his or her work, the ultimate responsibility for the quality of the XYZ software and associated documentation produced by THE iTEAM rests with the Project Manager/Team Lead. The SQA Manager shall implement the SQA procedures defined in this plan and shall monitor project staff activities and review products for compliance to the standards and procedures. The results of SQA monitoring and analysis along with SQA recommendations for corrective action shall be reported to the Project Manager. In addition, all documents and software approved by the Project Manager for release to shall have been reviewed and approved by the quality assurance leader.

4. Documentation

4.1 Purpose

All documentation that supports the XYZ software or its development process shall be created and updated periodically throughout the development cycle. In addition, to ensure the adequacy of these documents, they will undergo a peer review process.

4.2 Minimum documentation requirements

The following documents will be produced.

- SQAP: software quality assurance plan (this document)
- SCMP: software configuration management plan
- SPMP: software project management plan
- SRS: software requirements specifications
- SDD: software design document
- STD: software test documentation
- User's manual
- Maintenance plan

In addition to these documents, the PHP source code will utilize PHPdoc, and will therefore be capable of auto-generating documentation.

4.3 Other

The SVVP (software validation & verification plan) will be generated and maintained by an organization independent of the SQA organization.

5. Standards, practices, conventions and metrics

5.1 Purpose

This section describes the standards, practices, conventions and metrics to be used for the XYZ project. These are intended not only to ensure quality for *Encounter*, but also to obtain quantitative metric data on the SQA process itself. This data is to be used to help elevate the CMM level of Gaming Industries Consolidated (GCI), the developers of *Encounter*, from level three to level four.

5.2 Content

[*Note to the student:* Describe the standards, practices etc. to be used. Organization-wide quality goals can be supplied here, or in a separate appendix. The contents of this section should be specific. For example, statements such as "quality should be as high as possible" should be avoided.]

Standards: the IEEE standards, with appropriate modifications, are to be used for documentation.

Practices:

1. To verify the delivery of a fully conforming, high-quality product, every individual assigned to the project will participate in quality assurance.
2. Since providing quality software that meets the wants and needs of our customers is out topmost priority, engineers should apply quality precepts while they work, rather than as an afterthought.
3. All project artifacts are inspected by at least 2 reviewers and are made available to the team on the team's Google project page.

Writing Conventions: All documentation, including code comments and wiki articles should be written in a concise manner, with no grammatical errors.

Coding Conventions: The following coding convention should be used:

- Class names should start with an uppercase letter and multiple words should be separated with an underscore (e.g. Class_Name). All class methods should be entirely lowercased and named to clearly indicate their function, preferably including a verb. Try to avoid overly long and verbose names.
- Variables should contain only lowercase letters, use underscore separators, and be reasonably named to indicate their purpose and contents. Very short, non-word variables should only be used as iterators in for() loops.
- Use only one statement per line

- Braces should always be included when writing code using if, for, while etc. blocks even if the braces could be omitted.
- Braces should always be placed on a line on their own. They should also align properly so a closing brace is always in the same column as the corresponding opening brace.

Metrics: At least three metrics will be maintained for every process, and every document.

1. time spent by individuals on subtasks.
 2. quality self-assessment on a scale of one through ten, approximately in a bell-shaped distribution. Self-assessment scores will not be used for the evaluation of personnel by management. Failure to produce them may negatively impact the evaluation of an engineer by management, however.
 3. number of defects per unit (e.g., lines of code).
4. Number of Deviations Open per SRD vs Number of Deviations Closed per SRD

GCI quality goals for delivered products are as follows.

[*Note to the student:* the numbers used here should be based on historical data obtained from the group that is developing *Encounter*.]

Detected within 2 months of delivery:

Requirements: no more than one minor defective detailed requirement per 100 requirements.

Design: no more than one minor defect per five diagrams;

Pseudocode: no more than two minor defects per thousand lines

Code: no more than two minor defects per KLOC

The actual metric data from this project are to be reported as appendix X to this document.

SQA is responsible for reporting these measurements to the Project Manager/Team Lead on a bi-weekly basis.

6. Reviews and audits

6.1 Purpose

The purpose of the review process is to formally review the product and identify any defects as early as possible. Reviews carry this out in a scheduled and thorough manner. Audits do so on the basis of random sampling with short notice.

6.2 Minimum requirements

[*Note to the student:* large projects require the full set of reviews and audit listed here.

At a minimum, student teams should conduct reviews and inspections of requirements and design, as well as post-mortem reviews. "Reviews" are discussions of proposed artifacts. "Inspections" are conducted on completed artifacts delivered to the team.] Refer to the SPMP for the schedule of reviews described below.

6.2.1 Software requirements review

This is a walkthrough of the requirements document in the presence of the entire team. The review will be lead by the project leader. It is expected that the requirements will not have been inspected prior to this review. This review is not intended to replace inspections of the requirements. The requirements leader (see SPMP) will be responsible for seeing to it that these inspections are carried out.

6.2.1A Software requirements inspection

All requirements will be inspected in accordance with CGI Inc's inspection process manual, document CGI 345678.

6.2.2 Preliminary design review

This is a review of alternative architectures with the entire team. The review will be led by the project leader or his/her designee. It is expected that the team will provide feedback, which will be reflected in the final design. The alternative architectures will not have been inspected prior to this review. The design leader (see SPMP) will be responsible for seeing to it that this review is carried out.

6.2.3 Critical design review

This is an inspection of the proposed architecture, in the presence of the entire team. The design leader will be responsible for seeing to it that these inspections are carried out. The architecture will have been inspected prior to this review. If possible, the architecture will be decomposed into detailed designs of its parts, and these will undergo separate critical design reviews.

6.2.4 SVVP review

Since V&V is to be conducted by an independent team, there will be no review of the SVVP Plan by QA.

6.2.5 Functional audit

Prior to delivery, the project leader shall be responsible for checking that the product being delivered satisfies the requirements in the SRS. Where exceptions are necessary, the project leader shall obtain the prior consent of his or her management to allow the delivery. He or she shall communicate these exceptions to the customer by appropriate means, including a "readme" file, and a cover letter citing this file.

6.2.6 Physical audit

Prior to each delivery, the QA leader shall be responsible for checking that the physical software and its documentation designated for delivery are indeed delivered.

6.2.7 In-process audits

Project personnel should expect random audits of their work. This will consist of visits to the work site by teams designated by division management. A day's notice shall be given for all visits. The subject of these audits will be the current work of teams and individuals that has been allocated to the project.

As the organization migrates to CMM level five, all work will be made freely available to all team members and auditors at all times. They will be organized in a clear, standard fashion, so that audits will be possible without any notice.

6.2.8 Managerial review

The *Encounter* project shall be reviewed by the VP for Engineering during the first week of every month. It is the project leader's responsibility to schedule this review.

6.2.9 SCMP review

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

6.2.10 Post mortem review

As with all GCI projects, the *Encounter* team shall conduct post-mortems of all phases, in order to provide a log for future projects. These will include reviews of the project phase just completed, as well as 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 Inspections

All artifacts of the XYZ project will be inspected by all team members upon their completion and any issues found should be fixed within 10 days and resubmitted for follow-up .

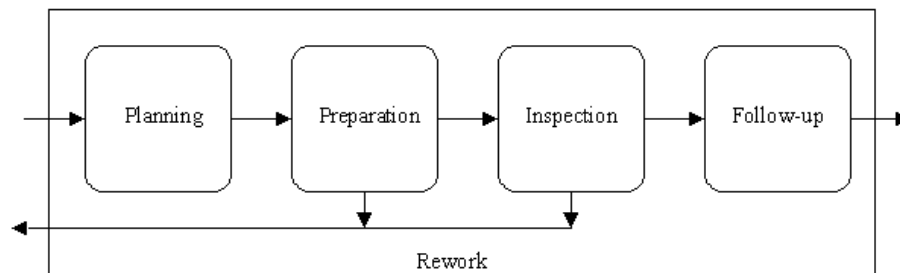


figure 1: The inspection process