Software Quality Assurance Plan

Introduction

Scope and intent of Software Quality Assurance (SQA) activities

The SQA team's objective is to ensure that the product does not deviate far from the original design specifications. If it is discovered that deviation has occurred, the SQA team will notify the development team to prevent future deviations and to correct the previous deviations. Also, the SQA team will perform a walkthrough to analyze the product's quality at any particular stage of development. Error detection and possible enhancements are also expressed to the development team.

SQA organizational role

The SQA organizational role is to review the product(s) at specific times during product implementation. Upon reviewing, the SQA team's duties will be to evaluate the software at its current development stage and recognize any defects in the subsequent stage (design or implementation). The SQA team will directly interact with the software engineering team in group discussions, discussing any errors or possible enhancements that have been identified. In addition, the SQA team will ensure that the software engineering team has not deviated in any way from the initial design specifications.

SQA Tasks

Task Overview

Description of SQA Task 1

The Engine Software Engineer will check with the Requirements Specification on a weekly basis to make sure that what he is coding conforms to the original design. This process will ensure that the product meets the client's expectations and standards and that the engine, up to its current point, is working properly.

Critique: This is a reasonable approach, but we need to be more specific. How will the "check" be conducted? It is important that this activity be conducted in a systematic way so that things don't fall into the cracks.

Work products and documentation for Task 1

As a result of Task 1, any major deviations that occur will be expressed to the other group members and documented on a separate defect log. Documentation will ensure that each group member is aware of the change(s) made to the engine so that each part of the project can be adjusted accordingly.

Comment: Good idea. The log should be available to all, on-line.

Description of SQA Task 2

The User-Interface Software Engineer will check with the Requirements Specification on a weekly basis to make sure that what he is coding conforms to the original design. This process will ensure that the product meets the client's expectations and standards and that the user-interface, up to its current point, is working properly.

Critique: See critique for Task 1.

Work products and documentation for Task 2

As a result of Task 2, any major deviations that occur will be expressed to the other group members and documented on a separate defect log. Documentation will ensure that each group member is aware of the change(s) made to the interface, so that each part of the project can be adjusted accordingly.

Description of SQA Task 3

Each member of the group will routinely perform a hands-on evaluation of the user-interface. Noted evaluation criteria will be: ease of use, principle of least astonishment, unobtrusiveness, and overall attractiveness. This is done to ensure that the user-interface is evaluated honestly, and remains easily understandable and attractive.

Work products and documentation for Task 3

As a result of Task 3, all suggestions or concerns are expressed to the User-Interface Engineer. These are recorded in the defect log. Based on these concerns, the User-Interface Engineer takes note and makes the appropriate adjustments to the user-interface to make sure that the final product is satisfactory.

Description of SQA Task 4

Each member of the group will routinely perform a hands-on evaluation of the DirectX engine. Noted evaluation points will be: any logic errors and/or software glitches that occur, and any desired enhancements. This is done to ensure that the DirectX engine is evaluated honestly, and that it is defect free, sufficiently powerful, and efficient.

Work products and documentation for Task 4

As a result of Task 4, all suggestions or concerns are expressed to the Engine Software Engineer. These are recorded in the defect log. Based on these concerns the Engine Software Engineer takes note and makes the appropriate adjustments to the DirectX engine to make sure that the final product is satisfactory.

Description of SQA Task 5

An SQA leader will be appointed to (1) control the frequent SQA reviews; (2) keep track of all SQA meetings; and (3) manage the flow of information to the correct software engineer. In addition, the SQA leader will review each product defect or enhancement that has been reported, then assign a priority rank to each. The higher the priority rank, the more important it is to fix the defect or enhancement. Priority ranking will be determined by a group discussion, involving the software engineers, and headed by the SQA leader.

Work products and documentation for Task 5

As a result of Task 5, all suggestions or concerns expressed during each evaluation will be recorded. In addition, each recorded item will be assigned a priority ranking and the date the item was reviewed. All requests for defect fixes and enhancement implementations will be recorded.

Standards, Practices and Conventions (SPC)

Every software engineer’s work will be evaluated weekly to ensure that the project is continuing smoothly and on schedule. Upon review, the current prototype will also be checked to determine if the software engineer is deviating from the original specification.

Unscheduled reviews of every software engineer’s work will be conducted to ensure that each subsystem is given ample attention during implementation. By making unscheduled evaluations, it can be determined whether the software engineer is allocating enough time for each subsystem or if the work is being rushed without attention to detail.

All software engineers are responsible for submitting any major design changes or implementation variances to the SQA leader. This procedure will account for all impacts on the rest of the software resulting from the change(s). Every major change will be recorded by the SQA leader, including which software engineer requested the change and the date requested.

All software engineers are expected to thoroughly test each completed subsystem of the software following guidelines outlined in the Test Specification. This is to ensure that each subsystem is working properly and efficiently. Any major defect found will be reported to the SQA leader.

Critique: It would be a good idea to cross reference the SCM Plan here. As changes are requested based on SQA activities, the SCM process kicks in.

SQA Resources

An SQA leader will be assigned to control the flow of information from the SQA team to the software engineers. The SQA leader will oversee the software quality control to ensure that the software engineers are conforming to the standards of the Requirements Specification document. Furthermore, the SQA leader will have the duty of assigning a relative rank of priority to every defect reported - functional or cosmetic. The priority will be used to determine which defects or enhancements are deemed the most important for the software engineers to correct first. Every defect or enhancement request is submitted to the SQA leader for review. The SQA leader will be in control of all SQA activities including SQA meetings and reviews.

Critique: Might be useful to indicate how priority will be assigned.

Because software engineers are most familiar with implementation of their particular part of the software, each software engineer will perform software quality analyses. Before and after each subsystem is complete, the software engineer will review the Requirements Specification document to ensure that the subsystem is implemented within the bounds of the original design.

Each team member will actively and frequently test the current prototype of the software for possible defects or necessary enhancements. This ensures that each subsystem of the software is functioning properly and follows the Requirements Specification document. Completely debugging one’s own source code can be difficult; therefore, each team member will be responsible for checking every major iteration of the software prototype to ensure that many or most defects are intercepted in early programming stages.

No special software or hardware will be needed to conduct the software quality assurance walkthroughs. It may, however, be helpful for each group member to have access to a central defect-report database, which will be reviewed and updated frequently. Although this is not necessary, it will be helpful in keeping software defects and enhancements organized and easily accessible to every group member for review.

Reviews and Audits

Generic Review Guidelines

Conducting a Review

The software will have scheduled reviews to detect any defects in the current prototype, and to determine any notable enhancements that should be implemented prior to product shipping. This will be completed for the user-interface and for the DirectX engine. The current prototype will be broken down into smaller subsets and each subset will be reviewed. The development team as a whole (consisting of four people) will attend each review, and critique each other's part of the software to ensure that the maximum number of possible defects are accounted for.

The group will meet on a weekly basis to discuss defects or enhancements. Each meeting will consist of no more than a couple of hours. The current prototype for both the user-interface and the engine will be available at every meeting allowing defects to be explicitly pointed out to each group member, especially the software engineer. If the defect found could not be easily duplicated, the SQA leader will take note of the defect. Any desired enhancements will be discussed to determine which are necessary and feasible. Any enhancement found to be too difficult or unnecessary for product completion will be noted by the SQA leader.

Roles and Responsibilities

The SQA leader will oversee any formal technical reviews. Any defects or enhancements will be discussed and recorded by the SQA leader. Each defect or enhancement will be given a priority rank, which will be recorded. Once the review is complete, the SQA leader will make a summary of each defect or enhancement and distribute them to the appropriate software engineer.

Each software engineer will be responsible for reviewing his own software module during module creation and upon module completion. Once each major software module is complete, it is the software engineer's duty to inform the SQA leader that the module is ready for review.

Review work products

The SQA leader will keep a defect log. The defect log contains all defects and enhancements, as well as a priority rank for each. The following will also be noted in the defect log: (1) whether or not the defect or enhancement has been handled, (2) which software engineer oversaw the correction, and (3) what date the correction was completed.

Formal Technical Reviews

Description of System Specification review

Description of focus of the System Specification review

The System Specification Review will provide a forum to analyze the proposed design of the software. To determine any obvious design flaws, the focus of this review will be to analyze the major software functions outlined in the System Specification. Once a design defect has been recognized, the SQA team will discuss with the software engineers any ideas or suggestions on how to compensate for the design flaw.

Timing of the review

The System Specification Review will be held upon completion of the System Specification. This should occur within the first few weeks of the software's development. This is necessary so that the underlying design of the software is sound and will not create any serious problems for the software engineers in the future.

Work products produced

The SQA leader will create a summary report of the System Specification Review. The summary report will include any changes to the software's major subsystem design. Once subsystem design defects have been identified, the SQA team will discuss possible solutions with the software engineers. Each possible solution will be noted and reviewed to determine if the solution will have an impact on the rest of the design. Once all obvious design defects have been handled, the System Specification will be amended to account for the design changes.

Review checklist

- Is the proposed design the best possible solution?
- Is there a better way to break up the software into subsystems? If yes, how?
- Is there any obvious design flaws that have not been accounted for? If yes, what?
- Are there any necessary enhancements for the software?
- Is the proposed System Specification within the time frame?
- Is each subsystem possible to implement in languages of choice?

Description of System Project Plan review

Description of the focus of the Software Project Plan review

The Software Project Plan review will focus on determining if the proposed cost and scheduling estimates are feasible. If it is determined that the proposed estimates within the Software Project Plan are not practical estimates, the estimates will be discussed and the Software Project Plan will be amended to compensate.

Timing of the review

The Software Project Plan review will be held upon completion of the Software Project Plan, which should occur within the first few weeks of the software's development. This is necessary so that the scheduling and cost estimates of the software are within reason and can be followed relatively precisely during the software's development cycle.

The System Project Plan review should also be held as software development commences. This will help keep the project on track, and within the cost and schedule estimates.

Work products produced

The SQA leader will create a summary report of the System Project Plan review. This includes any serious over and under estimations that have been made with regard to development time and cost. If problems have been found, the System Project Plan will be amended with appropriate estimate changes.

Review checklist

- Is there enough time to complete the project overall?
- Is there enough resources to complete the project overall?
- Has enough time been allocated for development of each subsystem of the software?
- Has too much time been allocated for development of a specific subsystem?
- Have tasks been delegated appropriately? If not, why?

Description of the Risk Mitigation, Monitoring & Management (RMMM) review

Description of the focus of the RMMM review

The RMMM review will focus on determining if the proposed risk management for the development of this software is within reason. Main focus will be on if the risk management is capable of handling a proposed problem accordingly. If the RMMM document is not managing each risk accordingly, the RMMM will be amended to correct the oversight.

Timing of the review

The RMMM review will be held upon completion of the RMMM document. This should occur within the first few weeks of the software's development. The RMMM review is necessary, because it provides for a decent guideline on how to handle a problem if a proposed risk occurs early in the software development.

Work products produced

The SQA leader will create a summary report of the RMMM review. This includes any possible risks that have not been covered, and any risks that have been accounted for, but are not managed appropriately. Once a new risk is proposed, a discussion will take place on how to handle the risk if it were to occur. Any risks being managed inappropriately will also be discussed and an amendment to their management will be made to better handle the risk.

Review checklist

- Have all risks been thoroughly covered in the document? If not, what is missing?
- Of the risks covered in the document, are there any that did not seem to be effectively covered? If yes, which one(s)?
- Of the risks covered in the document, are there any that did not seem to be appropriately managed? If yes, which one(s)?

Description of Requirements Specification review

Description of focus of the Requirements Specification review

The Requirements Specification review will work to analyze the proposed design of the software. The focus of this review will be to remove or discuss changes to any obvious design flaws. Once a design defect has been encountered, the SQA team will discuss with the software engineers any ideas or suggestions about how to compensate for the design flaw.

Timing of the review

The Requirements Specification review will be held upon completion of the Requirements Specification. This should occur within the first few weeks of the software's development. This is necessary so that the underlying design of the software is sound and will not create problems for the software engineers in the future.

To ensure that the software is conforming to the restrictions of the design, each software engineer will frequently conduct his own Requirements Specification review. If the software engineer determines that the current design of a module is flawed, it will be brought to the attention of the SQA leader and an appropriate discussion to amend the problem will be held.

Work products produced

The SQA leader will create a summary report of the Requirements Specification Review. This includes any defects or enhancements that have been brought to attention. Once design defects have been identified, the SQA team will discuss possible solutions with the software engineers. Each possible solution will be noted and reviewed again to determine if the solution will have an impact on the rest of the design. Once all obvious design defects have been handled, the Requirements Specification will be amended to account for the design changes.

Review checklist

- Is the proposed design the best possible solution?
- Is there any obvious design flaws that have not been accounted for? If yes, what?
- Are there any necessary enhancements for the software?

- Is the proposed Requirements Specification within the time frame?

Description of Data Design review

Description of focus of the Data Design review

The Data Design Review will work to analyze the proposed design of the major data members for the software. The review will focus on determining if the design of the data objects of the software is within the bounds of languages of choice (C++ and Visual Basic). Further focus will be on reviewing each data object to determine if the object is designed correctly and efficiently. If a data object does not make sense or is not effective in reducing software complexity, a discussion of how to redesign the object to better fit within the constructs of our software will be held.

Timing of the review

The Data Design Review will be held upon completion of the Requirements Specification and System Specification. This should occur within the first few weeks of the software's development. This is necessary to ensure that the underlying design of the data objects within the software is sound. Further, it will help to ensure that data abstraction and communication is at their peak.

Each software engineer will frequently conduct his own Data Design review to ensure that the software is conforming to the restrictions of the design. If the software engineer determines that a current data design is flawed, it will be brought to the attention of the SQA leader and an appropriate discussion to amend the problem will be held.

Work products produced

The SQA leader will create a summary report of the Data Design Review. This includes data design flaws that have been brought to attention. Once data design defects have been identified, the SQA team will discuss possible solutions with the software engineers. Each possible solution will be noted and again reviewed to determine if the solution in question will have an impact of the rest of the design. Once all obvious data design defects have been handled, the Requirements Specification will be amended to account for the design changes.

Review checklist

- Is the proposed data design the best possible solution?
- Is there any obvious data design flaws that have not been accounted for? If yes, what?
- Does each data object make sense? If not, why?
- Does each data object help to abstract data, or is it simply unnecessary?
- Does each data object further enhance the software's simplicity?
- Does the data object contain any unnecessary information? If yes, what?
- Do the data objects interact properly or are their lines of communication too complex?

Description of Architectural Design review

Description of focus of the Architectural Design review

The Architectural Design review will provide a basis for analysis of the proposed architectural design of the software. The focus of this review will be to assess the current design to ensure that data flow and data control are being handled appropriately. If a design flaw has been discovered, the SQA team will discuss with the software engineers any ideas or suggestions about how to compensate for the architectural design flaw.

Timing of the review

The Architectural Design review will be held upon completion of the Requirements Specification and the System Specification. This should occur within the first few weeks of the software's development. This is necessary to ensure that the underlying architectural design of the software is sound and will not create problems for the software engineers or degrade the software's performance in the future.

Each software engineer will frequently conduct his own Architectural Design review to ensure that the software is conforming to the restrictions of the architectural design. If the software engineer determines that the current architectural design of a module is flawed, it will be brought to the attention of the SQA leader and an appropriate discussion to amend the problem will be held.

Work products produced

The SQA leader will create a summary report of the Architectural Design review. This includes any defects in the architectural design that has been brought to attention. Once architectural design defects have been identified, the SQA team will discuss possible solutions with the software engineers. Each possible solution will be noted and again reviewed to determine if the solution will have an impact on the rest of the design. Once all obvious architectural design defects have been handled, the Requirements Specification and System Specification will be amended to account for any architectural design changes.

Review checklist

- Is the proposed architectural design the best possible solution?
- Is there any obvious architectural design flaws that have not been accounted for (i.e. slow data flow or control)? If yes, what?
- Are there any obvious changes you see would further enhance the software's performance?
- Is the proposed architectural design complete? If not, what seems to be missing?
- Does the proposed architectural design seem possible within languages of choice? If not, why?

Description of Interface (GUI: Graphical User Interface) Design review

Description of the focus of the Interface (GUI) Design review

The Interface Design review will focus on determining if the proposed graphical user-interface is possible, flexible, easy to use, and aesthetically pleasing. If it has been determined that the current proposal for the GUI does not meet any of the standards expected by the client, possible corrections will be discussed.

Timing of the review

The Interface Design review will be held upon completion of the Requirements Specification and the System Specification. This should occur within the first few weeks of the software's development. This is necessary so that the final GUI is as simple to use as possible, flexible (the ability to achieve the same results using various means), and aesthetically pleasing. In order to meet these standards, ample time must be allowed to amend the GUI design.

The Interface Design review should also be held as software development commences. This will help to keep the project on track. The SQA leader will have to be informed when any change or enhancement is made to the GUI.

Work products produced

The SQA leader will create a summary report of the Interface (GUI) review. This includes any changes that have been made or need to be made to the GUI. The summary will also list which software engineer will be responsible for the desired change to the GUI. The System Specification and the Requirements Specification will also have to be amended to reflect the Interface changes.

Review checklist

- Is the GUI pleasing to the eye? If not, why?
- Does the GUI allow you to do certain tasks many different ways?
- Does the GUI conform to what you expect a GUI to function like?
- Does the GUI make sense? If not, why?
- Is the help file helpful?
- Is the GUI forgiving when is comes to user error?

- Is too much or too little information being echoed to the user?

Description of Source Code review

Description of the focus of the Source Code review

The Source Code review will focus on determining if the software's source code is efficient, fast, modular, highly cohesive, and lowly coupled. If it has been determined that the current source code is flawed in one or more of these ways, the source code will be reviewed to determine what can be done to remedy the problem.

Timing of the review

The Source Code review will be officially held nearing product completion. This is to ensure that the software engineers have produced reliable source code that can be easily maintained.

It will also be held persistently by the software engineers as they are writing source code for the software. This is to ensure that the entire software package does not have to be completely rewritten before product shipping. This will cut down on completion time and enhance the final software's overall quality.

Work products produced

The SQA leader will create a summary report of Source Code review. This includes any changes that have been made or need to be made to the Source Code. The summary will also list which software engineer will handle the desired change to the Source Code.

Review checklist

- Does the source code allow for easy maintainability?
- Is the source code easy to read?
- Is the source code commented well?
- Is the source code reliable? If not, why?
- Is the source code efficient? If not, why?
- Is the source code highly modular?

Description of Test Specification review

Description of the focus of the Test Specification review

The Test Specification review will focus on determining if the proper software testing procedures are being accounted for. If it's brought to attention that the Test Specification is not thoroughly covering all aspects of software testing, an amendment to the Test Specification document will be made.

Timing of the review

The Test Specification review will be held upon Test Specification document completion to ensure that the proper testing procedures will take place over the course of the software implementation and completion.

At product completion, the Test Specification review will be held again to ensure that all tests outline in the Test Specification have been followed and passed testing evaluation.

Work products produced

The SQA leader will create a summary report of Test Specification review. This includes any changes that have been made or need to be made to the Test Specification.

Review checklist

- Does the Test Specification seem to test each module thoroughly?
- Are both black box and white box testing being used?
- Are there any tests to any part of the software that you think are not being covered already?
- Are there tests that seem to be overly redundant?
- Is it specified that if changes are made to a software module how to deal with testing after the change takes place?

Description of Change Control review

Description of the focus of the Change Control review

The Change Control review will focus on determining if the proper steps are being followed when making a change to the overall design or implementation of the software. Whether or not the Change Control document is handling change requests appropriately will also be noted. If a problem in the management of Change Control has been identified, a discussion will be held on how to amend the Change Control document to handle the limitation.

Timing of the review

The Change Control review will be held upon Change Control document completion to ensure that the proper change request management is taking place. This will help to eliminate unnecessary changes, and determine what impact a change could have on the entire system. This also helps to ensure that a step by step process takes place from submitting a change request to request approval or dismissal.

Work products produced

The SQA leader will create a summary report of Change Control review. This includes any changes that may be added to the Change Control document. If a requested change to the Change Control document is deemed to be appropriate, the Change Control document will be amended to correct the oversight.

Review checklist

- Does the Change Control document appear to filter out unnecessary changes in the product specification?
- Are the appropriate actions being taken to ensure that any possible side effects are being evaluated?
- Does the Change Control document contain a controlled method of change request?
- Are change requests thoroughly evaluated?
- Are change requests being thoroughly documented?
- Is it specified who has the final decision with regards to the request?

Component Design Reviews

Description of component GUI Database Interface review

Description of focus of the GUI Database Interface review

The GUI Database Interface review will be held upon completion the user-interface's database handler. The focus of the review will be to determine if the database handler is functioning properly and efficiently. In addition, the database handler will be reviewed to determine whether it is powerful enough to allow for the flexibility that will be necessary for this software.

Timing of the review

The GUI Database Interface review will be held upon completion of the user-interface's database handler. This should occur within the first few weeks of the user-interface development, because the user-interface is heavily dependent on a flexible database system.

Work products produced

The SQA leader will create a summary report of the GUI Database Interface review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the User-Interface Engineer. This will inform the engineer what defect or enhancement should be handled first.

The GUI Database Interface review checklist

- Does the database make sense?
- Is updating the database via the user-interface simple?
- Does the database handler successfully create the text files that are used in the DirectX engine?
- Is the database flexible enough to handle future user-interface enhancements?
- Have any unknown errors occurred? When?
- Is the information inside the database easily understandable through the user-interface?

Description of component GUI Level Editor review

Description of focus of the GUI Level Editor review

The GUI Level Editor review will be held upon completion of the user-interface's level editor handler. The focus of the review will be to determine if the level editor handler is functioning properly and efficiently. In addition, the level editor will be reviewed to determine whether it is powerful enough to allow for the flexibility that will be necessary for this software, and if the level editor is simple in its design.

Timing of the review

The GUI Level Editor review will be held upon completion of the user-interface's level editor. This should occur roughly halfway through the software's development cycle. The level editor will be essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the GUI Level Editor review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the User-Interface Engineer. This will inform the engineer what defect or enhancement should be handled first.

The GUI Level Editor review checklist

- Is the level editor easy to use?
- Is the level editor flexible enough to handle a wide array of basic arcade games?
- Does the level editor function properly and efficiency?
- Are there any enhancements that you see that would make the level editor simpler or more powerful?
- Does the level editor make sense?

Description of component GUI Mini-map review

Description of focus of the GUI Mini-map review

The GUI Mini-map review will be held upon completion the user-interface's mini-map. The focus of the review will be to determine if the mini-map is functioning properly, efficiently, and is easy to use. In addition, the mini-map will be reviewed to determine whether it is powerful enough to allow for the flexibility that will be necessary for this software. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The GUI Mini-map review will be held upon completion of the user-interface's mini-map and level editor. This should occur roughly halfway through the software's development cycle. The mini-map will be essential to project flexibility and ease of use and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the GUI mini-map review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the User-Interface Engineer. This will inform the engineer what defect or enhancement should be handled first.

The GUI Mini-map review checklist

- Is the mini-map easy to use?
- Does the min-map seem to function properly?
- Does the mini-map allow for enough flexibility?
- Does the mini-map make sense or not at all?
- Is the mini-map too tedious to use? If yes, why?
- Are there any enhancements you would like to see in the mini-map?

Description of component GUI Sprite Wizard review

Description of focus of the GUI Sprite Wizard review

The GUI Sprite Wizard review will be held upon completion to the user-interface's sprite wizard. The focus of the review will be to determine if the sprite wizard is functioning properly, efficiently, and is easy to use. In addition, the sprite wizard will be reviewed to determine whether it is powerful enough to allow for the flexibility that will be necessary for this software. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The GUI Sprite Wizard review will be held upon completion of the user-interface's sprite wizard. This should occur roughly halfway through the software's development cycle. The sprite wizard will be essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the GUI sprite wizard review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the User-Interface Engineer. This will inform the engineer what defect or enhancement should be handled first.

The GUI Sprite Wizard review checklist

- Is the sprite wizard easy to use?
- Are all the functions of the sprite wizard working properly?
- Is every function necessary for creating a sprite contained within the wizard?
- Does the sprite wizard allow for enough flexibility?
- Does the sprite wizard make sense or not at all?
- Is the sprite wizard too tedious to use? If yes, why?
- Are there any enhancements you would like to see in the sprite wizard?

Description of component GUI Sprite Selector review

Description of focus of the GUI Sprite Selector review

The GUI Sprite Selector review will be held upon completion the user-interface's sprite selector. The focus of the review will be to determine if the sprite selector is functioning properly, efficiently, and is easy to use. In addition, the sprite selector will be reviewed to determine whether it is powerful enough to allow for the flexibility that will be necessary for this software. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The GUI Sprite Selector review will be held upon completion of the user-interface's sprite selector. This should occur roughly halfway through the software's development cycle. The sprite selector will be essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the GUI sprite selector review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the Engine Software Engineer. This will inform the engineer what defect or enhancement should be handled first.

The GUI Sprite Selector review checklist

- Is the sprite selector easy to use?
- Does the sprite selector allow for easy accessibility to sprites?
- Does the sprite selector make sense to you?
- Is the graphical layout of the sprite selector well done?
- Are there any enhancements to the sprite selector that would make it more powerful or easier to use?

Description of component DirectX Engine Database interface review

Description of focus of the DirectX Engine Database interface review

The DirectX Engine Database interface review will be held upon completion the DirectX Engine's database interface. The focus of the review will be to determine if the database interface is functioning properly, efficiently, and is creating the desired output. In addition, the database interface will be reviewed to determine if it allows for a suitable amount of flexibility when it comes to interpreting the data. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The DirectX Engine Database interface review will be held upon completion of the DirectX Engine's database interface. This should occur within the first few weeks of the software's development cycle. The database interface will be essential to project completion and without it will be impossible to correctly implement the DirectX engine; therefore, it should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the DirectX Engine Database Interface review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the Engine Software Engineer. This will inform the engineer what defect or enhancement should be handled first.

The DirectX Engine Database Interface review checklist

- Is the database working properly?
- Is the engine interpreting the data correctly?
- Is the database flexible enough to allow for some errors?
- Is the database interface implemented well in your opinion?
- Are there any enhancements that would make the database easier to understand or simpler to use?

Description of component DirectX Engine DirectX Handler review

Description of focus of the DirectX Engine DirectX Handler review

The DirectX Engine DirectX Handler review will be held upon completion the DirectX engine's DirectX handler. The focus of the review will be to determine if the DirectX handler is functioning properly and is powerful enough to allow for a great amount of flexibility. In addition, the SQA team will review the DirectX handler to determine its ease of use. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The DirectX Engine DirectX Handler review will be held upon completion of the DirectX engine's DirectX handler. This should occur within the first few weeks of the software's development cycle. The DirectX handler is essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the DirectX Engine DirectX Handler review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the Engine Software Engineer. This will inform the engineer what defect or enhancement should be handled first.

The DirectX Engine DirectX Handler review checklist

- Does the DirectX handler simplify DirectX programming?
- Does the DirectX handler function properly?
- Are there any missing features that should be included in the DirectX handler?
- Does the DirectX handler appear to function smoothly and quickly?
- Is the DirectX handler flexible enough?

Description of component DirectX Engine Object Handler review

Description of focus of the DirectX Engine Object Handler review

The DirectX Engine Object Handler review will be held upon completion the DirectX engine's object handler. The focus of the review will be to determine if the object handler is functioning properly and is powerful enough to allow for a great amount of flexibility in programming. In addition, the SQA team will review the object handler to determine its ease of maintainability. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The DirectX Engine Object Handler review will be held upon completion of the DirectX engine's object handler. This should occur within the first few weeks of the software's development cycle. The object handler is essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the DirectX Engine Object Handler review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the Engine Software Engineer. This will inform the engineer what defect or enhancement should be handled first.

The DirectX Engine Object Handler review checklist

- Does the object handler simplify DirectX programming?
- Does the object handler function properly?
- Are there any missing features that should be included in the object handler?
- Does the object handler appear to function smoothly and quickly?
- Is the object handler flexible enough?
- Does the object handler make sense?
- Is the object handler commented well?

Description of component DirectX Engine Logic Handler review

Description of focus of the DirectX Engine Logic Handler review

The DirectX Engine Logic Handler review will be held upon completion the DirectX engine's logic handler. The focus of the review will be to determine if the logic handler is functioning properly and is powerful enough to allow for a great amount of flexibility in programming. In addition, the SQA team will review the logic handler to determine its ease of maintainability. If a defect has been found or an enhancement suggested, the SQA team will meet with the software engineers to discuss the solution.

Timing of the review

The DirectX Engine Logic Handler review will be held upon completion of the DirectX engine's logic handler. This should occur within the first few weeks of the software's development cycle. The logic handler is essential to project completion and should be completed as promptly as possible.

Work products produced

The SQA leader will create a summary report of the DirectX Engine Logic handler review. This includes any defects or enhancements that have been brought to attention. The summary report will also include a priority rank for the Engine Software Engineer. This will inform the engineer what defect or enhancement should be handled first.

The DirectX Engine Logic handler review checklist

- Does the logic handler simplify DirectX programming?
- Does the logic handler function properly?
- Are there any missing features that should be included in the logic handler?
- Does the logic handler appear to function smoothly and quickly?
- Is the logic handler flexible enough?
- Does the logic handler make sense?
- Is the logic handler commented well?

SQA Audits

Monthly audits of the SQA activities will be held. These audits will allow the SQA team to determine which SQA activities are working to deter product defects and assess how well each SQA activity is being conducted.

The audits will assist in keeping the SQA teams on track, as well as enhance beneficial SQA activities or eliminate useless activities. Each SQA activity will be assessed to determine how well it is affecting product quality. The defect log will be reviewed to determine what methods of SQA have been most useful and which have been least useful. The most useful SQA activities will be reused in further SQA reviews and the least useful will either be discarded or reevaluated to determine how to make the activity more useful.

The defect log will be reviewed to determine if all necessary product enhancement and defect reports have been submitted and what action was taken to handle the defect or enhancement. If the defect or enhancement was handled effectively and quickly, it will be noted and the method used will be reused for further defect removals or product enhancements.

Problem Reporting and Corrective Action/Follow-up

Reporting mechanisms

All defects or enhancements are referred to the SQA leader. Most often, this will be completed in every SQA review held. If a defect occurs between SQA meetings, the defect will be reported to the SQA promptly, so that the SQA leader can analyze the defect and assign a priority level to it. This can occur either by word of mouth (so long as the SQA leader is taking note), email, or a hand written report.

Responsibilities

The SQA leader is in charge of all SQA activities. Any defect or enhancement reported is analyzed and given a priority ranking. The SQA leader then assigns a particular software engineer to correct the problem or add the enhancement. After the correction or addition has taken place, it is the SQA leader's responsibility to review it and ensure that the appropriate action was taken. If the solution or enhancement does not meet the standards of quality set, the SQA leader will redistribute the task to the software engineers until the task is complete.

Data collection and evaluation

The SQA leader is responsible for all data collection and evaluation. Any product defect or enhancement will be reported to the SQA leader so that he can record the problem and evaluate its priority level. The data is collected by keeping a record of each SQA meeting and keeping track of all problem submissions outside of the SQA meetings.

Statistical SQA

Any defect submitted to the SQA leader will be analyzed. The underlying cause of the product defect will be traced. Once the cause has been identified, the source of the problem is discussed with all group members to determine the best possible course of action. Once a decision has been made, the software will be further analyzed to determine if correcting the original defect has caused any new problems. The group will again determine the best possible solution to correct the defect.

The underlying cause of the defect in question can be traced to one or more of the following:

- Incomplete specification
- Misinterpretation of customer’s needs
- Intentional deviation from Requirements Specification
- Violation of programming standards
- Error in data design
- Incomplete testing
- Inaccurate documentation
- Error in programming language translation of Requirements Specification
- Inconsistent human-computer interface

Once the cause is identified, it will be easier to focus on how to correct the problem and what effects the correction will have on the rest of the software. Measures will be taken to deter the same mistake from happening again. This is especially helpful if a common problem is discovered at the design phase of the software development cycle. If the problem occurs quite frequently, the problem will be reviewed and what measures can be taken to eliminate future instances of the problem will be discussed as a group.

Software Process Improvement (SPI) Activities

Goal and objectives of SPI

The goals and objectives of the SPI are to lower the frequency of software defects and to determine the underlying cause of the defects that occur. Furthermore, once the underlying causes have been identified, measures will be taken to determine the best course of action to eliminate the problem.

SPI tasks and responsibilities

Based on the Statistical SQA information gathered, the SQA leader will keep a tally of what errors or causes of errors occur most frequently. The more often a defect occurs from the same underlying cause, the more problematic an area will be considered. Depending on the nature of the cause and the individuals involved, two actions can take place: (1) the SQA leader will investigate the Statistical SQA information and the defect log to determine if the problem area exists primarily for a single software engineer, or (2) if every software engineer experiences the same problem.

If the problem occurs mostly from one software engineer, the software engineer will be informed of the frequency of the personal problem area. Most often, the software engineer has a better idea of how to handle his own implementation oversights, however, the SQA leader will give suggestions on how to reduce the error.

If the problem occurs from all software engineers, the software engineers' development practices will be analyzed to determine the cause of the problem. The problem and possible solutions will be examined at an SQA meeting to all of the software engineers.

Some time later the SQA leader will re-evaluate the Statistical SQA information to determine if the problem area has been reduced. If not, the problem area is analyzed once again to determine what can be done to reduce the occurrence of the defect.

Software Configuration Management Overview

Any change in the software's design will have to be submitted to the software configuration manager. This is done with a formal or informal submission, either electronically or in person. The change request is reviewed to determine the possible impacts the change would have on the entire software system. Once the impacts have been determined, they are reviewed to judge if the change request is a worthy change, or if the change would cause too many software problems. If the software configuration manager dismissed the change request form, the individual or group will be notified of the dismissal. The individual or group is then free to re-evaluate their change request and resubmit at a later time. Every change request is kept on record whether it is implemented or not.

For more information see the Software Configuration Management Document.

SQA Tools, Techniques, Methods

All SQA activities will follow the same guidelines and methods. Every SQA meeting will include every group member. Every group member is expected to participate in the discussion. Any group member not attending the review will be notified by the SQA leader of what took place at the review. The SQA leader will oversee the discussion and will take notes of any defects or enhancements that need to be analyzed.

The SQA team will analyze the defects or enhancements and determine their complexity, impact on the system, and priority. Once prioritized, the SQA leader will assign each item to the software engineers along with their priority.

After a defect has been eliminated or an enhancement added, the software engineer will inform the SQA leader at the next SQA review. The SQA leader will take note of the correction.

No special tools will be necessary for SQA although access to a central database that all group members can access would be helpful to cut down time and duplication of error.

Appendix

SQA Metrics Used

Function Point: this metric will be used when calculating statistics pertaining to the complete testing process.

Bang Metric: this metric will be used to provide an indication of the number of test cases required.

Cyclomatic Complexity: this metric will be used to target modules as candidates for extensive unit testing. Modules with high cyclomatic complexity are likely to be more error-prone.

Breadth of Testing: this metric provides an indication of testing completeness.

Depth of Testing: this metric provides a measure of the percentage of independent basis paths covered by the testing versus the total number of basis paths in the program.

Fault Profiles: this metric is used to prioritize and categorize uncovered errors.

Frames per second: this metric is used to gauge the performance of the game engine.