Peer Review Policy

**Pookas ETventure**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Primary Author(s) | Comments |
| 1.0 | 08 Oct 2009 | Ivan Loh, Sarah Lam, Goh Li-Xian | First Version |
|  |  |  |  |

# 1 PEER REVIEW

### Goals

Goal 1

Ensure the Software Engineering processes are complete

Goal 2

Ensures the delivered software has all defects identified and fixed.

# 2 Commitment to perform

#### Commitment 1 -- The project follows a written organizational policy for performing peer reviews.

This policy typically specifies that:

1. The tasks are performed in accordance with the project’s defined software process.
2. Appropriate tools and software are used to build the delivered software
3. The peer reviews focus on the review of the software being produced and not on the producer.

# 3 Ability to perform

#### Ability 1 -- Adequate resources are provided for performing all the tasks

Resources are provided to:

1. Software are provided to complete all the software engineering processes
2. Tools provided to developers to design and develop the software
3. QA is provided with tools and debugger to test the software
4. Participate in the peer review and any follow-up reviews required based on the defects identified in the peer review.
5. Monitor the rework of the software work product based on the defects identified in the peer review.
6. Collect and report the data resulting from the peer reviews.

#### Ability 2 -- Members of the software engineering technical staff received required training to perform their technical assignments for their roles.

Each member are given required training in areas that they are required for their roles.

Examples of training include:

* The principles and importance of software engineering
* Training to use the existing software so that the team could further enhance it
* The project plan, design specifications and requirements analysis
* C++ familiarization
* Activeworld familiarization
* Blackboard API familiarization

#### Ability 3 -- Reviewers who participate in peer reviews receive required training in the objectives, principles, and methods of peer reviews.

Examples of training include:

* types of peer reviews (e.g., reviews of software requirements, software design, coding, and quality assurance testing);
* the objectives, principles, and methods of peer reviews;
* roles of reviewers;

# 4 Activities performed

#### Activity 1 – The software follows closely to the software engineering process

1. The software functions are produced according to the design specs.
2. Methods and tools appropriate for use on the software project are selected.
3. All the requirements specified in requriments analysis are met

#### Activity 2 -- Software testing is performed according to the project's defined software process.

1. Testing criteria are developed and reviewed as appropriate.
2. Different levels of testing is performed:
   * unit testing,
   * integration testing,
   * system testing, and
   * acceptance testing
3. Following test strategy are selected:
   * + functional (black-box),
     + integration testing

#### Activity 3 -- Integration testing is planned and performed according to the project's defined software process.

1. The plans for integration testing are documented and based on the software development plan.
2. The integration test cases and test procedures are reviewed with the individuals responsible for the software requirements, software design, and system and acceptance testing.
3. Integration testing of the software is performed against the designated version of the software requirements document and the software design document.

#### Activity 4 -- System and acceptance testing of the software are planned and performed to demonstrate that the software satisfies its requirements.

System testing is performed to ensure the software satisfies the software requirements.

1. Acceptance testing is performed to demonstrate to the customer and end users that the software satisfies the allocated requirements.

The following are prepared for the testing:

* + preparing testing documentation,
  + scheduling testing resources,
  + developing test cases.

1. The test cases and test procedures are planned and prepared by the QA that is independent of the software developers.
2. The test cases are documented and are reviewed with the team before testing begins.
3. Problems identified during testing are documented for bug fixes.

# 5 Measurements

#### Measurement 1 -- Measurements are made and used to determine the functionality and quality of the software produced.

Examples of measurements include:

* numbers, types, and severity of defects identified in the software product

#### Measurement 2 -- Measurements are made and used to determine the status of the software engineering activities.

Examples of measurements include:

* status of each allocated requirement throughout the SDLC
* number of changes made to the software engineering process due to changes
* duration and effort to implement and test incorporated changes, including initial estimate duration.

# 6 Verifying implementation

#### Verification 1 -- The activities for software engineering are reviewed with the entire team on a regular basis

The team reviews the software engineering processes/activities regularly up to ensure the software meets all the processes.

#### Verification 2 – QA reviews and/or audits the activities and work produced for software engineering processes and reports the results.

QA reviews and/or audits verify that:

1. The software requirements are reviewed to ensure that they are:
   * complete,
   * correct,
   * consistent,
   * feasible, and
   * testable
2. Readiness and completion criteria for each software engineering processes are satisfied.
3. System and acceptance testing of the software are performed according to documented plans and procedures.
4. Tests satisfy their acceptance criteria, as documented in the software test plan.
5. Tests are satisfactorily completed and recorded.