QA docs 7 Summary

The scope of a review is the detection of problems, but not the correction of them. Thus, a reviewed item will be scrutinised for errors, omissions, inconsistencies, etc. and any problems found will be formally recorded. The changes which must be made will be determined after the review meeting, according to the change control process described in QA document SE.QA.08

Each group member should regard every item that they develop as the product of the group, not the individual.

Comments made about a reviewed item should not be taken as personal criticism of the individual who wrote it.

The criticism is intended to be constructive, since the purpose of the review is to detect problems, thereby reducing difficulties later in the project and resulting ultimately in a high-quality product.

A review will be performed prior to an item being released, and so it should have for review status. An item cannot have release status until it has been reviewed and only then if it has passed any conditions set on it at the review.

The review process should follow these steps:

1. Select the review team and arrange review time and place.

2. Distribute relevant documents.

3. Hold the review.

4. Note and complete actions

**QA Review meetings**

The QA Manager is responsible for arranging the review meetings. He or she must contact all relevant members of the review team and arrange a date and time at which all those persons can attend. A review meeting should not take more than two hours and may take less. The QA Manager is responsible for finding a room in which the review will be conducted, and for giving adequate notice of the date, time and place of the review meeting.

Normally, notice of at least one week should be given.

All relevant documents must be distributed to the review team by the QA Manager in advance of the review

meeting. The following documents should be made available together with any other documents the QA Manager considers appropriate:

* For a user interface specification review, the requirements specification as well as the user interface specification must be made available.
* For a design specification review, the requirements specification and the design specification must be made available.
* For a test specification review, the requirements specification, the design specification, and the test specification must be made available.
* For a software verification review, the design specification and the code must be made available.

It is most important that the appropriate versions of documents are used, and it is the QA Manager's responsibility to ensure that each review team member gets the correct version of each relevant document. Distribution can be done electronically, since all the relevant project documents will be held in the project's repository. The QA Manager could distribute the document by e-mailing each review team member, specifying in the mail message:

• The date, time, and place of the review meeting.

• The name, and version of each item necessary for the review.

The team member could then access the relevant documentation from the repository.

The QA Manager will be responsible for recording problems and actions. All other review team members should contribute to the meeting and criticise constructively the item being reviewed. Appendices to this document contain checklists of questions for each kind of review. After the meeting, the team responsible for the item will carry out the corrective actions and resubmit the document to the QA Manager for approval and release. It is possible that people will be directed to carry out actions that are not associated with problems. For example, the client might be requested to supply background information concerning the requirements. In such situations, the desired action must be described in the meeting minutes, and the person responsible for carrying out the action clearly indicated.

APPENDIX A - CHECKLIST FOR ALL DOCUMENTS

Questions to ask:

1. Do all formal project documents have the following information on the front cover:

• Title (indicating Group\_name and nature of document)?

• Author(s)?

• Configuration Reference?

• Date when the latest version was produced?

• Version number (correct)?

• Document status (Draft, Release)?

• Name, Address of Dept and Copyright notice?

2. Does the header for the document contain title, version and status on each page?

3. Does the footer contain "*Aberystwyth University / Computer Science*" and "Page x of y" correctly?

4. Does the document contain specified sections:

• Contents

• Introduction: Purpose of document, Scope, Objectives

• <Main Body> with appropriate sections

• References

• Document Change History with Version, Issue No., Date, Sections Changed from Previous Versions, Changed by

5. Are the sections numbered correctly from 1?

6. Are fonts appropriate in headers and in body text?

APPENDIX B - USER INTERFACE SPECIFICATION REVIEW

The User Interface Specification hand-in includes two items - a formal UI specification document defining what

the interface must do, and a presentation showing what the UI will look like.

Questions to ask:

1. Does the formal UI specification meet the General Document Standards (see [5] and questions in appendix A)?
2. Does the typical users section give good real-world examples of the range of people that might want to use the system?
3. Do the use cases cover all the items in the product functions section of the requirements specification?
4. Do the use cases give sufficient detail of what will happen in each use case?
5. Does each use case match what is said about it in the requirements?
6. Are any requirements not covered in the use cases?
7. Do the error conditions represent the wrong inputs that might need to be caught by this system?
8. Does the presentation cover each of the use cases, and clearly indicate which use cases are being covered?
9. Does the presentation give sufficient detail that you can understand what is needed in the UI when implementing?

APPENDIX C - DESIGN SPECIFICATION REVIEW

For a design specification review, the objective is to determine whether the design covers all the client's requirements as described in the requirements specification, i.e. it is a software verification activity. Thus, the review should proceed by considering each section in the requirements specification and attempting to verify that the requirements specified in that section are catered for in the design.

Questions to ask:

1. Does the document meet the General Document Standards (see [5] and questions in appendix A)?
2. Does the design adequately address the overall architecture of the system?
3. Does the design show the interaction between the modules sufficiently?
4. Does the design give a correct interface for each module specified?
5. Is there sufficient information on difficult pieces of the design?
6. Will the design meet the requirements? This should be verified from the requirements mapping table, and through examination of the requirements specification.
7. Do the reviewers understand the design?
8. Can the design be simplified?
9. Will the design lead to a maintainable system?
10. Could the design be modified to offer more opportunity for reusing existing software libraries?

APPENDIX D - TEST SPECIFICATION REVIEW

For a test specification review, the objective is to determine whether the testing will comprehensively exercise the system such that if all tests are passed, the system can be said to meet the client's requirements. The test specification must be shown to specify tests that exercise the system comprehensively such that all system functions are performed.

The review should proceed by considering in turn each section of the requirements specification and attempting to confirm that the requirements specified in each section are catered for in the testing.

Questions to ask:

1. Does the document meet the General Document Standards (see [5] and questions in appendix A)?
2. Does the Test Specification have the matrix of Requirement/Test ref/Test content/Pass criteria for each functional requirement? Do the tests cover every requirement?
3. Are the detailed tests reproducible?
4. Do the tests for each requirement cover all possibilities for the relevant requirement? In particular, do they test for all possible types of failure? Are boundary conditions tested?

APPENDIX E - SOFTWARE VERIFICATION REVIEW

For a software verification review, the objective is to determine whether the code implements all aspects of the design as described in the design specification, i.e., it is a software verification activity. Thus, the review should proceed by considering each section in the design specification and attempting to verify that the design information specified in that section is implemented in the code. The code should compile at the time it is being reviewed.

Questions to ask:

1. Does the module definition match the one given in the Design Specification?

2. Does the code achieve what is specified in the Design Specification?

3. Does the module header meet the Coding Standards SE.QA.09 [7]?

4. Are there appropriate comments to document the module, as defined in the coding standards?

5. Are there sufficient other comments to guide the maintainer?

6. Are all variables initialised before their values are used?

7. Have all constants been named?

8. Is each loop certain to terminate?

9. Are upper and lower bounds of arrays sensible?

10. Other questions may be appropriate depending on the technology being used.