



Bodington III PLE Quality Assurance Plan

1. Scope of This Document

This Document describes quality assurance aspects of the Bodington III PLE Project.

2. Quality Assurance Plan

2.1 Quality Expectations

The project will deliver the eLearning Tool(s) as specified in their proposal and refined in the JISC project plan document in line with following standards/guidelines:

- JISC (draft) Open Source Policy May 2004
- JISC (draft) Software Quality Assurance August 2004
- JISC Project Management Guidelines December 2003
- Release versions of development and final code are to placed with <http://sourceforge.net/>
- CETIS project page be maintained to communicate development progress and mapping of software to the ELF (eLearning Framework). <http://www.cetis.ac.uk/>
- Software should meet the high level functional specification as specified in the project plan.
- Software should be robust, maintainable and extendable (see JISC (draft) Software Quality Assurance August 2004).

2.2 Tolerances

- Cost – project must be completed within agreed grant.
- Time – project must be completed by 30th April 2005.
- Scope – given the short time scale of the project and the unknown nature of the associated risk the scope of the deliverables may be narrowed to ensure completion on time and to budget. Any changes to scope must be motivated by the project's technical manager and documented via a change control procedure.
- Quality – project must adhere to the standards as defined for open standards, open source and software quality

2.3 Acceptance Criteria

To be agreed with JISC during the project.

2.4 Quality Responsibilities

Ultimately all quality control responsibility is the technical project manager's. Individual contractors will have direct responsibility for the quality of their output. Peer review and peer testing will be used to assess quality where appropriate.

2.4.1 Standards and Technologies

- Java
- Java Web Application Specification 2.4
- WebDAV RFC 2518
- XML 1.0
- XHTML 1.0
- UML 1.4

2.4.2 Quality Control and Audit Processes

A quality control and audit process will be developed for each work package as part of the work package. The process definition will identify:

- When or at what stage software testing takes place
- How it will be carried out?
- How the results/faults will be recorded
- The level at which software is being tested
- Testing tools to be used

It is expected that these processes will be as diverse as the workpackages.

2.4.3 Change Control Processes

Change control over project plan, scope, budget (in as much as this is possible within the JISC-approved budget), specifications, requirements, deliverables, standard of quality, and timescale is the responsibility of the project technical manager. He will seek collaborative input to and discussion of the any potential changes from the entire project team in regular meetings. He will inform the Programme manager of any significant changes within a week of those changes being made.

2.4.4 Configuration Management Processes

Configuration management will be provided by local CVS or other suitable mechanism.

2.4.5 Quality Control Tools

These will be chosen as and when need for their deployment arises as part of individual workpackages.

3. General Quality Assurance Issues

Quality assurance will be underpinned by the precept that “good internal design is indicated by software code whose overall structure is clear, understandable, easily modifiable, and maintainable; is robust with sufficient error-handling and status logging capability; and works correctly when implemented.”

3.1 Requirements Analysis and Use Case Deployment

- Requirements analysis to the level of detail required for initial framework development is provided by use cases in the original University of Leeds bid.
- In addition, integrated system test processes will be driven by a further use case analysis. The latter will be performed as part of the generation of a integrated test system (and any servlets that populate it for test purposes).

3.2 Architectural, Design, and Implementation Quality Assurance Practices

- Architectural level specifications will be used as the basis of the development of individual component-level and interface specifications
- UML will be used wherever possible to reduce development time and to provide system documentation. The UML will preferably be produced using CASE tools, but from time to time may sometimes be provided in the form of photographs of clearly-drawn whiteboard diagrams. If used, the photographs will be electronically accessible in an organised form.
- Java class documentation, as per best practice, will be embedded in the code and automatically be made available to documentation and available to . Changes to requirements must follow the change control process defined by the project.
- Coding standards will be developed as appropriate. As examples: In extending Tomcat, Tomcat's standards will be used. In developing project specific components a suitable standard will be adopted that includes details of naming conventions and an exception hierarchy.
- Agile development methods may be used to assure code quality