



JISC Project Quality Plan Template

This document defines the quality expectations the project must achieve and how they will be met.

1. Quality Expectations

The JISC programme manager completes this section defining the standards and level of quality expected to be achieved by the project.

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).

Tolerances

- Cost – project must be completed within agreed grant.
- Time – project must be completed by 31st March 2005.
- Scope – given the short time scale of the project the scope of the deliverable (i.e. eLearning Tool(s)) may be narrowed to ensure completion on time and to budget. Any changes to scope must be agreed with the programme manager and documented via the change control procedure.
- Quality – project must adhere to the standards as defined for open standards, open source and software quality

2. Acceptance Criteria

For each of the main deliverables of the project criteria for its acceptance / completion are defined.

Successful completion of an external evaluation of the projects software outputs and development process.

3. Quality Responsibilities

List of who is responsible for monitoring and ensuring quality for deferent aspects of the project?

Michael Gardner – Project manager – will be responsible for:

- Ensuring the project adheres to the JISC quality assurance policy
- Responsibility for the project and quality plans and overall quality control
- Quality of project documentation/change control

John Scott – Lead software developer – will be responsible for:

- Overall software quality control
- Backup, system testing, configuration and version control
- Supervision of the software team

Anuroop Shahi, Titto Assini – programmers – will be responsible for:

- Unit testing quality

Chris Fowler - Senior academic advisor – will be responsible for:

- Usability quality

4. Standards and Technologies

Referenced list of standards and technologies to be used by this project.

- the W3C OWL specification will be used for the definition of the resource ontologies. See <http://www.w3.org/2004/OWL/>
- Protege 2.1.2 will be used for the creation of OWL-DL ontologies. See <http://protege.stanford.edu/>
- the RACER 1.7 semantic web inference engine will be used. See <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>
- the HP JENA 2.1 open-source semantic web toolkit will be used. See <http://jena.sourceforge.net/>
- the W3C RDQL query language for RDF will be used. See <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>
- all of the models will be implemented using the W3C Resource Description Framework (RDF). See <http://www.w3.org/RDF/>
- the models generated by DELTA will be assessed against the main e-learning standard for learning design, IMS Learning Design.
- the grid-based components will be based on the Open Grid Service Architecture (OGSA) standards and specifically the Globus GT3.2 toolkit and the OGSA Data Access & Integration (OGSA-DAI) release 4 toolkits. See <http://www.ogsadai.org.uk/>
- the web services model will be used throughout based on the SOAP 1.2 protocol (see <http://www.w3.org/2000/xp/Group/>) and WSDL 1.1 (see <http://www.w3.org/TR/wsdl>)
- all code will be developed in Java 1.4.1 and open source tools (eg. JUNIT 3.8.1, Tomcat 4.1.12, Axis 1.2, ANT 1.6.2). See <http://jakarta.apache.org/>
- all of the systems design will be based on the UML 1.4 specification. See <http://www.omg.org/technology/documents/formal/uml.htm>

5. Quality Control and Audit Processes

Description of the process to be used to control project quality and enable auditing.

All software modules will be unit tested using JUNIT.

All code will be commented to JAVADOC guidelines.

Inter-operability and integration testing will be carried out at the end of the project.

Regular project team meetings will be held to review and discuss the technical progress.

An issue-log will be created to log all problems/faults and to review/manage their resolution.

Where resources are based (e.g. case studies) on the IMS LD specification, conformance to IMS LD will be achieved by either using examples directly from the specification or by using the RELOAD Learning Design Editor.

6. Change Control and Configuration Management Processes

Description of the process to be used to manage change and configuration management.

All deliverables (including code) will be stored on the Delta project folder which is backed-up on a daily basis.

The project issues log will be used to record all issues, requests for change and software faults. This will be reviewed at regular project meetings (at least monthly).

All formal project meetings will be minuted and stored on the project server.

CVS will be used to manage the version control of all software produced and regular updates will be placed onto Sourceforge.

7. Quality Tools

List any tools to be used to help ensure quality.

- CVS 1.11.17 for software version control. See <https://www.cvshome.org/>
- JUNIT 3.8.1 for Java unit testing. See <http://jakarta.apache.org/>
- Protégé 2.1.2 for OWL compatibility checking. See <http://protege.stanford.edu/>