



JISC Project Quality Plan Template

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).

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

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

3. Quality Responsibilities

Programming Manager (Mr. Brian Fuchs)

- Software
 - Version control
 - Testing

Project Director (Dr. Amy C. Smith)

- Budget
- Workplan/staff management
- Risk management
- Liaising with JISC/Programme Manager
- Timing
- Documentation
 - Plans/Progress and other reports
 - Minutes of meetings
 - Project websites (at JISC, CETIS, and Ure Museum)

4. Standards and Technologies

Preexisting projects

- Virtual Lightbox (<http://www.mith2.umd.edu/products/lightbox/>)

- OpenAdapter (<http://www.openadapter.org>)
- Programming Code
- Java (<http://java.sun.com/>)
- Metadata
- Dublin Core (<http://dublincore.org/>)
 - RSLP (<http://www.ukoln.ac.uk/metadata/rspl/schema/>)
- Terminology
- ICOM-CIDOC/Getty (<http://www.willpowerinfo.myby.co.uk/cidoc/stand2.htm>)
- Semantic web compatibility
- RDF/XML (<http://www.w3.org/TR/rdf-syntax-grammar/>)
- Interoperability
- OAI (<http://www.openarchives.org/>)
 - JISC Interoperability Focus (<http://www.ukoln.ac.uk/interop-focus/>)
- Learning
- CETIS (<http://www.ukoln.ac.uk/interop-focus/>)
 - IMS Global Learning (<http://www.imsproject.org/>)
 - JISC Information Environment (<http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/standards/>)
- HTML/XML
- W3C (<http://validator.w3.org/>)
 - XML (<http://www.xml.com/>)
- Accessibility
- W3C (<http://validator.w3.org/>)
 - W3C-WAI (W3C (<http://validator.w3.org/>))
- Documentation for applet
- Docbook XML (<http://www.docbook.org/>)
 - UML (<http://www.uml.org/>)
- Version control
- CVS (<http://www.cvshome.org/>)
 - Sourceforge (<http://sourceforge.net/>)
- IDE
- Eclipse (<http://www.eclipse.org/>)

5. Quality Control and Audit Processes

Unit testing: JUnit

Document testing: JavaDocs

Performance testing: PerfAnal

Version control: CVS

GUI/System testing: we will utilise the services of undergraduates and other volunteers at the Ure Museum/University of Reading

Bugs/faults: Bugzilla, incorporating 'task kit' (after the code has been uploaded to Sourceforge)

6. Change Control and Configuration Management Processes

Code Review Process: We will conduct weekly virtual meetings during which we will assess and evaluate changes to the code and assess the needs for structural changes.

Workflow: All team members regularly log into Jabber (chat) so that we can maintain communication; we each keep records of these chats in our own working environments. We also employ CVS (version control) so that we can log changes to code and/or documentation, including rationale, during development. Once the code has been uploaded to Sourceforge, the 'task kit' feature of Bugzilla will allow us to keep a log of our programming issues and how they are tackled.

7. Quality Tools

RDF validator (<http://www.w3c.org/RDF/Validator/>)

HTML/XML validators

- W3C (<http://validator.w3.org/>)

- W3C-WAI (<http://validator.w3.org/>)