



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

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

3. Quality Responsibilities

1. **Skills Framework** – Simon Grant, Janet Strivens, Scott Hennessey
2. **Instantiation of Skills Framework** – Scott Hennessey, Adam Marshall, Simon Grant, Janet Strivens

3. **Definition of Medical Skills Framework** – Scott Hennessey, Adam Marshall, Simon Grant, Janet Strivens, Vivien Sieber, Peter Dangerfield, Andy Pellow
4. **Web Service Requirements** – Adam Marshall, Jon Maber
5. **Bodington Enhancements Requirements** – Adam Marshall, Jon Maber
6. **Enhance Bodington to Support Service** – Adam Marshall, Jon Maber
7. **Enhance Teachers Resource Creation Tools within Bodington** - Adam Marshall, Jon Maber
8. **Implementation of Exemplar Applications** – Scott Hennessey, Adam Marshall, Simon Grant, Janet Strivens, Vivien Sieber, Peter Dangerfield, Andy Pellow
9. **Search / Discovery Service** – Adam Marshall, Jon Maber, Vivien Sieber
10. **Skills Profiling Service** – Adam Marshall, Jon Maber
11. **Extend PDP Web Service API and SDK** – Adam Marshall, Jon Maber
12. **LUSID Goes Open Source** – Adam Marshall, Jon Maber, Steve Bickerstaff, Nick Bunyan
13. **Use Cases** – Scott Hennessey, Adam Marshall, Simon Grant, Janet Strivens
14. **Project Management** – N/A

4. Standards and Technologies

At this stage we think the following technologies and open standards will be used:

- Bodington
- Java Web Services Development Pack
- Eclipse 3 IDE – development environment
- TOMCAT 5 servlet container
- Apache AXIS library
- Apache Ant
- IMS RDCEO – to express skills framework
- IMS QTI – to express skills survey (and build up competency map)
- IMS VDEX – for use with skills framework
- z39.50, SRW – to search repositories (JAFER)
- XFORMS – to transform QTI learner answers into an instance of the framework which is used as basis as the search query
- WSDL – for description of web services
- IMS RLI – this is what the JAFER bean will return
- SOAP – for transmission of service data
- IMS LIP – for expression of competencies (or lack thereof)
- UK LOM – for learning objects meta data

In most cases we will use JAXB or AXIS to generate java classes from the schemas and documents – this will ensure conformance. We will also attempt to validate against the schemas wherever possible.

5. Quality Control and Audit Processes

The conditions under which the bid was submitted were that the resultant software “does not have to be perfect” and that “bugs may still exist”¹; it is very difficult to re-factor this very ambitious project to incorporate rigorous quality control without gaining any extra resources. So, it will not be possible to have the ultimate quality control procedures in place, however, we will do all we can to ensure that the software and documentation is of the highest quality possible under the circumstances.

All software will be tested using one or more of the following methodologies

- Black box testing
- Glass box testing
- Integration testing
- Unit testing

All textual material will be read by at least one project member who is not the main author and will be circulated to interested parties (eg, JISC/CETIS special interest groups) via email.

¹ This is what the bidders were told at the London ‘town-meeting’ where information was disseminated about how to write the proposal so this is how we calculated the time-spans of the different workpackages.

The project uses Microsoft Project Professional as a management tool. The original “.mpp” file will be available to all project members (and JISC project managers) and an up to date PDF version of the tasks and Gantt chart will be housed on the project website: <http://bodington.org/bodington/site/spws/>.

6. Change Control and Configuration Management Processes

The project will keep an issue log to record major changes to the project’s components. All software will be housed under CVS on SourceForge.net and the appropriate SourceForge facilities will be used (eg, bug tracking, discussion forums etc).

- The project has specified milestones within the work packages and these will be used to assess whether change management is required.
- We expect to need to make design decisions throughout the project and will evaluate whether the changes to the plan stray too far from the original proposal
- If the direction of the project does need to change significantly then we will consult the programme management team.

7. Quality Tools

We will use XML tools to check ‘standards’ conformance of all XML documents.

Other tools to used

- MS Project professional for project management
- Eclipse IDE / Net Beans for java development
- XML Spy (or Turbo XML) for XML authoring