



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

Workpackages	Quality Assurance
1. JSMIRK technology feasibility test	5 test applications are proposed and each will be signed off as success or fail
2. Full architectural specification of JSMIRK	The specification will be written by DAK and signed off after being considered as “implementable” by SJB
3. Develop beta version of JSMIRK	Heuristic evaluation of JSmirk interface And bug testing from external software evaluator commissioned.

	Documentation to this effect will be created.
4. Final JSmirk Deliverable	Documentation will be proof read and evaluated.
5. Smirkboard: Content Generation and Upload	A set of tests devised by SJB/DAK will be run and signed off after success
6. Smirkboard: Permissioning and Admin of SMIRKBOARD	A set of tests regarding access to material and functionality of users will be run and signed off after success
7. Smirkboard: Scorm package exporting and final version of server	Sample packages output by the system will be tested in 2 known SCORM players (Reload + 1 other)

3. Quality Responsibilities

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

David Kraithman is responsible for monitoring quality on JSMIRK
 Steve Bennett is responsible for monitoring quality on SMIRKBOARD and all documentation generated during the project

4. Standards and Technologies

XHTML+SMIL (W3C Note)
SMIL (W3C recommendation) v.2.0
IMS Content Packaging v1.1.3 or as incorporated within SCORM 1.2
ACCLIP metadata
Web Content Accessibility Guidelines (W3C)

In terms of coding, the following touchstones will be observed:

- Avoid use of global variables
- Avoid use of small trivial functions except in cases of tangible readability gain
- Comment all functions of greater than 10 lines in length, indicating the data-type and typical values of parameters, provenance of calls and general logic of functioning
- In use of error-trapping, comment on the likely cause of anomalous values arising
- In all serialization routines, ensure recovery from appearance of anomalous data

5. Quality Control and Audit Processes

Because JSMIRK is a multimedia authoring tool, and therefore a large amount of its functionality will be in the user interface, software evaluation by nature has to be through real user testing. It is notoriously difficult to use extreme programming style test harnesses for this type of software, and by prior experience we know that the best way to test such tools is to get people to use it “in anger” and report any faults arising. It is planned to be used in a course in February by approximately 80 students and so this will constitute a fairly extensive test environment.

SMIRKBOARD similarly is being developed as a proof-of-concept application rather than an industrial strength web service. However, testing will be done on the core deliverables via a set of user scenarios in order to evaluate and sign-off work done.

With the exception of the final JSMIRK deliverable, timed to take place at the end of the project itself, therefore not allowing any time for evaluation, all other work packages will have a specific sign-off report which may be audited during the project cycle and thereafter.

Compliance with standards will be achieved as follows:

- SCORM – via the Scorm Conformance Suite (deliverables 2,3,7)

- SMIL (through playback in two SMIL renderers) (deliverables 3,4,5)
- WCAG (through the Bobby tool) (deliverables 3,4,5)

Results of these tests will be published alongside the deliverable, as it appears on SourceForge though we anticipate a small time delay between the publication of the deliverable and the compliance verdict. We would seek guidance and review from Techdis/Cetis on the implementation of the ACCLIP metadata.

6 Change Control and Configuration Management Processes

In any case of development plan amendment, a minuted meeting will take place between all of the major actors in the project (SJB, DAK and SDS). These will be communicated to all relevant JISC stakeholders.

Version control of software will be done through the NetBeans internal version control system (or that of JBuilder – should we progress to that IDE). Weekly zip files of all source code used on the project will be kept for security purposes.

7. Quality Tools

JSMIRK and SMIRKBOARD will use the sourceforge repository's own version control functionality for versioning purposes. Some SMIRKBOARD testing may be undertaken using tools for simulating http requests in order to ascertain strains placed on the server by the hosting and delivery of web presentations.

Two conformance tools will be used

- Bobby 5.0 (for XHTML accessibility compliance with WCAG)
<http://bobby.watchfire.com/bobby/html/en/index.jsp>
- Scorm Conformance Suite 1.3.1 (for Scorm compliance)