



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 different aspects of the project?

Project Management (Kingston) – David Livingstone

ASAP Application Service (Kingston) – James Orwell [Senior Researcher]*

JPlag Application Service (Karlsruhe) – Guido Malpohl

RoboProf Application Service (Dublin) – Charles Daly

Blackboard User Agents (Kingston) – David Livingstone/James Orwell [Senior Researcher]*
User Testing (Kingston) – Graham Alsop
User Testing (de Montfort) – Richard Hall/Jane Curnock
Web-CT Feasibility Study (City University) – Jo Wood
* Line management structure indicated in []

4. Standards and Technologies

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

- Java J2EE for development and web tier architecture (<http://java.sun.com/j2ee/>)
- CETIS/ E-Learning Framework for conceptual architecture (<http://www.cetis.ac.uk>)
- XML for general schema including:
 - WSDL 1.0 – for network application services and documentation (<http://www.w3.org/TR/wsdl>)
 - Apache Axis 1.2 will be investigated for development of Web Portal user agents and application services
 - QTI -IMS 1.2.1 or QTI-IMS v2 is proposed for question and assessment specifications (<http://www.imsglobal.org/question/index.cfm>)
 - This issue will be discussed further with the RoboProf partner at a 2 day meeting November 18th/19th and will be confirmed at this stage.
- UML 1.4 for requirements analysis and design documentation (<http://www.uml.org/>)
- CVS v2 for version control (<https://www.cvshome.org/>)
- SourceForge for code and documentation publication (<http://sourceforge.net/>)
- Blackboard (release 6)
 - Use of Apache Axis 1.2 will be investigated for supporting Blackboard User Agent development
- Web-CT version 2 – with City University partner
- RoboProf – a new/modified version will be required to conform to the specification of the project (to be discussed at partner meeting – see above)
- JPlag – a new/modified version will be required to conform to the specification of the project (to be discussed at partner meeting – see above)

5. Quality Control and Audit Processes

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

Extreme Programming for development [BECK00]:

Analysis and Design by refactoring
Pre-defined Test-cases / Unit Testing
Pair programming (where possible)
Incremental change management (by agreement up the line)

Line of responsibility:

Project and Technical Managers (Metrics and refactoring)
Senior Researcher (Tracking)
Researchers (Code quality and testing)
User Testing Co-ordinator (User testing and feedback)

Weblog for project progress

Devise or source published/standard test cases for all adopted standards and technologies (where applicable)

6. Change Control and Configuration Management Processes

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

CVSv2 (<https://www.cvshome.org/>) and Tortoise CVSv1.7 (<http://www.tortoisecvs.org/>) for versioning and backup

WinCVS (www.wincvs.org/) is being evaluated

7. Quality Tools

List any tools to be used to help ensure quality.

CETIS for ELF compliance

Unit testing for code quality – using in-house code

JUnit v3.8 will be evaluated for in-house + student-testing purposes

XML Compliance and benchmarking (e.g. XSLTmark

<http://www.datapower.com/xmldev/xsltmark.html>) [or other more appropriate tool]

Rational Rose for UML modelling [or other more appropriate tool]

BECK00: K Beck, *“Extreme Programming Explained – Embrace Change”*, Addison-Wesley, ISBN 201-61641-6