JISC

JISC Project Plan Template

The Project Management Guidelines have detailed instructions for preparing project plans.
Expand tables as appropriate.
Fill in the information for the header, e.g. project acronym, version, and date.
Prepare a cover sheet using the cover sheet template and attach to the project plan.

Overview of Project

1. Background

Summarise the background to the project (and how it builds on previous work) and the need for it (and why it's important).

The primary motivation for the project is to produce tools that can address the issues of teaching a technically demanding subject with discipline-specific requirements in a changing Higher Education environment. This includes a variety of factors including: increasing class sizes and staff-student ratios; widening participation and distance learning; constantly evolving subject material. The project aims to widen the scope and availability of existing projects at Kingston University, Dublin City University, and The University of Karlsruhe. This will be achieved by integrating the implementation of these services into the JISC Technical Framework. Below, we first describe the existing main partner projects and their benefits, and then describe the need for the proposed project and how it will benefit the wider Higher Education community.

Existing Work at Kingston University

At Kingston University (the lead partner), there have been several projects to improve the quality of teaching and learning of programming. The principle investigators have been involved in three such projects, described below. Some relevant background information: the school teaches programming to an undergraduate cohort of around three-hundred students per year. The first programming language was C++; this is changing to Java in October 2004. The fourth project involves a further school academic with a research interest in the evaluation of learning and teaching environments.

1. **Increased usability of paper-based multiple-choice questions**
   (Dr. Orwell, Year 1 C++ modules, 2002)
   To improve participation and interest in what is for many a difficult and obtuse subject, it was proposed to include, in lectures, short multiple choice tests. To motivate the students, their results from each test would make a small contribution to their final mark. To compensate for the lack of exam conditions (i.e. students sitting closely together in the lecture hall), it was proposed to make six or so versions of the paper, each with a different order of both questions and answers. This project delivered software that automatically generated these pdf versions from a question-bank. It also provided features to appropriately format any source-code embedded in the questions or answers. The software has been used in subsequent years to efficiently produce and update multiple choice papers. Since questions are easy to add to the question-bank, different tests are used each year. Hence, the students are allowed to keep the question papers, the answers are made available immediately after the test, and papers from previous years are available beforehand. The reaction from students is that they find a regime of short frequent tests and immediate feedback a useful, challenging and enjoyable component to the course.

2. **Migration to Blackboard learning units and Perception Environment**
   (Mr. Livingstone, GIS Masters Modules, 2003)

This project involved the conversion and development of masters level materials for mixed-mode teaching (internal and distance-learning) incorporating formative testing and communication/feedback. There were two main components, which:

a) Investigated the use of Blackboard LMS learning tools (such as Learning Units) for providing both sequential and non-sequential access to electronic resources. The tools provided by the LMS, whilst being suitable for the construction of distance learning materials, provided an inflexible interface and a prescriptive structure that places restrictions on the form of the materials produced, and the ability to export and edit these materials.

b) Investigated the use of the internal Blackboard online assessment environment was used for formative testing and the project did preliminary evaluation of the Perception testing environment was for conditional branching. A requirement for the automated generation of online questions was identified.

The use learning units and the perception environment was useful but limited. A need for a more flexible, standardised, and generic framework for producing and structuring materials was identified.

3. **TAPAS - Technology for Automated Programming Assessment and Support**
   (James Orwell and David Livingstone, tools for first and second year Java modules, 2004). This project extends the results of the above two projects and adds some important innovative components. It addresses two components of programming assessment:

   a) Multiple Choice Questions. The Perception testing environment does not meet all the requirements for efficient creation and management of multiple choice tests. In particular, it is cumbersome to generate correctly formatted source code, embedded in questions and answers. This project provides tools for generating correctly formatted questions for use with the Perception System.

   b) Unit testing of student source code. To improve the feedback, accountability and methodology of programming exercises, it is proposed to create tools to allow students to submit their source code (answers to programming exercises) to be 'unit tested' to ensure it performs correctly. The results of the tests are provided back to the student (allowing them to correct their answer and resubmit until correct) and to the instructor (so that results can be used in grading students, where necessary).

4. **Evaluation of teaching and learning tools**
   (Graham Alsop, educational technology research and co-ordination, 2001-2004)
   The Learning Technology Research Group in the School of Computing and Information Systems has an established record in establishing the appropriate choice and use of research methods to study the application and use of learning technologies. To date these methods include: Grounded Theory [AT02], Activity Theory, Phenomenography [AT04a] and Action Research [AT04b]

Existing Work at Dublin City University

At Dublin City University (Partner) a set of Java Servlets, Roboprof [Da98], has been developed to provide an e-Learning environment for teaching a Java Programming course. Roboprof has been used for the last four years on introductory Java programming courses. It consists of a web based framework which presents a course to students and pluggable modules which deliver particular questions and feedback on student attempts. The framework gives information about the student to the question module which then creates an appropriate question. The module is also used to mark student responses. Most questions are of the short answer format but Roboprof also supports programming problems where student submissions are marked based on output to selected test cases. Recent evaluative work [Da04] has indicated that RoboProf marked lab exams seem to be better measurement of programming ability than end of year written exams.

Existing Work at the University of Karlsruhe

At the University of Karlsruhe a system, JPlag, has been developed since 1996 for the detection of similarities between sets of students' source code files and the commonalities with instructor-provided code. The system is not designed to identify similarities with source code from external sources e.g. the Internet, rather it compares a student's work with their peers or any other directly available sources of code. Using JPlag a quantitative score determines the degree of similarity with another piece of work and can be used to help assess the likely time and understanding that has gone into the student's submission. Along with other indicators, such as objective tests, this provides a useful way of monitoring a student's learning progress as well as being a deterrent against plagiarism. The system is offered as an online web service for subscribing academics and provides a client-side application with a graphical user interface. The service has been assessed by South Bank University as part of a JISC technical report [La01] which concluded that

> 'The UK computing academic community owes a debt of gratitude to Guido Malpohl and Alex Aitken, and their institutions, for developing and supporting the Service.'
> 'The JPlag service would seem to be the easiest to use [than the comparable service] and produces more comprehensible results.'

The need for the Proposed Project

All of the above work contributes towards a series of practical requirements identified by the individual workers or institutions. This proposal aims to combine the above existing work, and extend the applicability and availability in several important respects. The proposal is to make tools to enable the delivery of unit-test based assessment methods across a range of courses, programming languages and instructional management systems (IMS). These three factors can be thought of as progressively extending the applicability and availability of the existing work:

a) For the instructors, development of high-quality user-agent software allows non-experts to plan, develop and modify course assessement programs.
b) Abstraction of the unit testing engine to enable assessment of any programming activity, e.g. C++, C#, JavaScript, PHP,
c) Adherence to the JISC technical framework enables tools to be used in any compliant context.

The proposal is strengthened by the 'Roboprof' work at Dublin City University and the 'JPlag' work at University of Karlsruhe. The features of these initiatives will be included in IMS-compliant tools, so that the unit tests will be introduced and supplemented by automatically generated questions. In addition the source code submitted by the student will be subjected to the to a tool providing the JPlag service which can be used to indicate (in the case of summative assessment) whether the work is the student's own or (in the case of formative assessment) as an indicator of the student's personal contribution and, in conjunction with the other tools, an assessment of their level of understanding.

2. Aims and Objectives

List the broad aim or purpose of the project, and the specific objectives you intend to achieve.
The aim of the project is to develop, integrate and evaluate eLearning tools to support teaching, learning and assessment of computer programming languages.

The specific objectives are:

- The publication of appropriate standards for the incorporation of unit tests in the educational sector
- The publication of tools for authoring and using unit tests, that can be employed in conjunction with a variety of VLEs (Virtual Learning Environments)

- Interfaces for relevant third-party software, including the Roboprof and JPlag initiatives

The release of the source code for the tools developed during the lifetime of the project, under an appropriate license, and in a form suitable for future collaboration with interested parties.

3. Overall Approach

Describe the overall approach you will take to achieve the objectives outlined above, including:

- *Strategy and/or methodology and how the work will be structured*
- *Important issues to be addressed, e.g. interoperability*
- *Scope and boundaries of the work, including any issues that will not be covered.*
- *Critical success factors.*

The philosophy behind the proposed tool, ASAP, is that the use of industry-standard unit tests in student programming exercises represents both an important and useful contribution to their education and also a way of assessing their progress towards the course-specific learning outcomes. In particular, the following points are relevant:

1) Methodology.
   Computer Programming is a relatively young discipline, and has been subject to changes in approach, both in its teaching and its practise. Associated with the move towards object-oriented methods, an increasing importance has been attached to automated testing of each 'unit' or software component that comprise the overall project.
   This is established in industry as a software quality assurance tool, and also in the educational environment to ensure the assimilation of learning outcomes. The advantages of this approach are detailed in the literature [Ke99, LF03] but include: separation between interface and implementation; easier transfer of activities and responsibilities between programmers; and a rapid, flexible response to changing project requirements.

2) Continuity between summative and formative assessment.
   The proposed 'unit test' may be used in a formative environment, e.g. allowing students to work in pairs, with a text-book, with unlimited time, or in a summative environment, i.e. in a closed book, individual, fixed time exam. This is an improvement on the current situation, where the exam procedure bears little similarity to the in-course assessments and exercises.

3) Increased emphasis on practical activities.
   The proposed teaching method places a greater emphasis on the practice of programming. For the student, the importance of testing activities are re-enforced through the process of submitting code to be evaluated by an external agent.

4) Accuracy of grade.
   Automated code assessment using unit testing introduces an appropriate objective testing environment for Computer Programming. In the estimation of the investigators, the grade received by the student is likely to be a better indication of their programming ability than that obtained using current procedure, i.e. submitted assessments and paper-based exams.

5) Interoperability and Extensibility.
   It is important that any tool(s) developed are not 'one-offs' that cannot be used by others within and outside the institution. Computer programming trends change rapidly, and any tools need to be flexible and extensible enough to evolve. In addition, any tools need to consider interoperability to both be a benefit to and benefit from the community of practitioners with similar teaching and learning issues.

6) Testing and Evaluation.
   The project has identified a range of courses and modules, internal and external to the lead partner for testing and evaluation of the project outputs. These include:
   - Large group undergraduates, face to face teaching
   - Small group postgraduates, mixed-mode (both face-to-face and distance learning)
   - Java and C++ programming courses

- Other members of the JISC user group with complementary requirements

The project will investigate the interoperability of the partner's tools in conjunction with the ASAP tool in the context of the JISC E-learning framework as well as the different VLE's in partners contributing to the testing and evaluation. However, the prime focus will be on the production of the ASAP tools and its deployment in a practical environment. The feasibility and degree of integration/coupling of the external partner's tools will be of a lower priority and the initial testing and scoping of the project will determine how far these developments will extend.

The critical success factor for the project is to be able to produce a practical programming assessment tool based upon unit testing that can be automatically used by students through a browser environment.

4. Project Outputs

List the tangible deliverables (including reports) your project will create, and the less tangible knowledge and experience you hope to build and share.

The tangible deliverables are enumerated below: these are the reports detailing the requirements of the system; the deliverable software, and the final report.

WORKPACKAGE 2:

Review and report of scope of project incorporating recommendations for conformity to JISC Framework guidelines and interface specification for institutional context:

- D2.3.1 Report defining ASAP Service interfaces (including annotated UML or equivalent structured diagrams)
- D2.4.1 Report defining Roboprof Service interfaces (including annotated UML or equivalent structured diagrams)
- D2.5.1 Report defining JPlag Service interfaces (including annotated UML or equivalent structured diagrams)

WORKPACKAGE 3 Application Services:

- D3.2.1 Prototype Implementation of ASAP Web Application Service defined in 2.3 (including full release of source code on e.g. SourceForge)

WORKPACKAGE 4: Advanced Application Services

- D 4.3.1 Prototype Implementation as defined in D2.4.1
- D 4.4.1 Prototype Implementation as defined in D2.5.1

WORKPACKAGE 5: User Agents

- D 5.1.1 Prototype Implementation of Blackboard Building Block for ASAP service interface and content management
- D 5.5.1 Prototype Implementation second VLE (Web-CT) for ASAP service interface and content management

WORKPACKAGE 7: Dissemination

- D 7.1.1 Report and documentation for JISC

In addition it is envisaged that that knowledge will be acquired in the following areas:

- The application and transfer of Unit Testing from an industry to educational context
- An increased understanding of the relationship between the developing JISC Framework and educational standards, industry standard VLE's and Web service development approaches
- The range of requirements for eLearning tools across a spectrum of educational curricula and delivery environments with respect to teaching Computer Programming languages

5. Project Outcomes

List the outcomes you envisage, including their impact on the teaching, learning, or research communities, and what change they will stimulate or enable.

The principle outcome of the project is the availability, to the UK higher education community, of tools providing the unit testing assessment initiative being developed at Kingston University. The key philosophy behind the initiative to produce ASAP are enumerated in Section 3: Overall Approach. However, there are additional features that will result from the funding proposed project under the JISC Framework and incorporating the work of the partner:

- The work undertaken at Dublin City University will be incorporated to the tools to enable automatic generation of multiple choice questions
- The work undertaken at University of Karlsruhe will be incorporated to enable automatic detection of substantially similar work
- For the instructor, the tools will provide efficient content management of the assessment material, i.e. the tests and associated documentation. Instructors must be able to add, duplicate, modify, import, export, test and delete the test material.
- The system design will allow extensions to other, possibly new, languages. Java and C++ will be supported in the project, but the application services will be designed to accommodate other languages, e.g. PHP, C#, etc.
- The project is designed to encourage sharing of resources between UK HE/FE institutions – both infrastructural (servers,) and content (tests and course content).
- The core application services will be implemented in a platform-independent, IMS-compliant specification, so that they can be used in conjunction with any IMS-compliant user-interface. Some user agents for these interfaces (such as blackboard and eclipse) will be built, and API's will be made publically available to facilitate third-party development of user agents for other ADSFs.
- The software will be released as an open source license, development can continue by interested parties

At an institutional level, it is envisaged that the deployment of these tools will allow lecturers, involved in teaching computer programming languages to large groups, to concentrate more of their resources on substantive teaching and learning activities. Also it should allow them to be able to more easily assess the progress of an individual learner in a climate of decreased contact time or a distance learning environment. In addition it provides an equitable environment for the students that can contribute toward the university's Quality Assurance goals.

Scenarios

The following use-case scenarios are provided to give an indication of how the proposed system would be used within the teaching and learning community.

Scenario 'A': Student Tee takes his end of semester assessment

Student Tee is currently in his first term at Camford University, studying the popular Business, Biology and Computing (BBC) course. The computing modules were quite challenging as he had never done any programming before. However, To pass the Java Programming module, he must successfully attempt a programming activity in a 2-hour period, without looking

Tee logged in to a computer in his favourite corner of the lab, invoked a web-browser, and authenticated to the Blackboard system used at his university. He quickly navigated to the assessment page – the clock was ticking – and looked at the choice easy, medium and hard tests available for him to choose. The medium one looked OK – and that would mean he could get a 'B' grade, which would keep his parents happy. He would have to implement a class that broke down a name string into four components – title, first, initial and surname. He'd done something similar earlier in the term as coursework, except that was for a line of an address, not a name. Looking at the example inputs and required outputs, he thought he'd figured out how to do it – every element could be split up by looking for the intermediate spaces, and all titles and initials were followed by a dot. He quickly downloaded the initial file and started up the text editor, thankful of all the times he'd done this before.

Forty minutes later, Tee thought he'd completed the exercise. It worked on the examples he's tried, so he thought he'd submit it to the system. He held his breath, waiting for a response from the browser….. Almost! 4 out of 5 tests correct, but his solution had failed on a name with two middle initials: he'd overlooked that case. Still, there was time to modify and correct his answer. He took ten minutes adding the necessary loop to that part of the source code, and this time the system confirmed he'd got all 5 out of 5 right – even though they were different tests to the ones it'd used last time. Tee sat back in his chair and felt content. Perhaps he could have had a go at the difficult one after all; but in any case, he could try it next term if he wanted to. He wanted to email his parents to tell them he'd passed the test, but email was disabled during the test, so he would have to log out and do that from elsewhere.

Scenario 'B': Dr. Turing revises his course for the new semester.

It was the end of September and Dr. Turing had just returned from an important conference in the Maldives. He really must turn his attention back to his undergraduate courses: he wanted to make two changes to the coursework in his C++ course. Using a search engine, he'd found solutions to his problems up on a website, which certainly suggested it was time to change. Besides, he wanted to add some extra assignments that used the new library available for parsing xml files.
He logged into the learning management system used by the university, and accessed the control panel for the unit tests.

The programming exercise with the internet-available solution required the student to write a component to convert Centigrade to Fahrenheit, and vice-versa. He decided to require instead a component to convert Pounds to Euros. This required an additional parameter (because the exchange rate was a variable) but all the changes could be seen on a single page of the control panel. For the unit tests, he used several difficult cases – exchange rate of zero, out-of-bounds floating-point number, and negative quantities. He uploaded his own solution, and hit the button that tested it…. Excellent, no errors. Finally, he indicated on the form that the solution would be made available to the students after the fourth week of term, by which time they should have attempted their own versions.

6. Stakeholder Analysis

List key stakeholder groups and individuals that will be interested in your project outcomes, will be affected by them, or whose support/approval is essential, both within your institution and in the community, and assess their importance (low/medium/high).

Stakeholder	Interest / stake	Importance
Computing / Information Systems academics	high	high
Academic Development Department / Registry	high	high
Central Computing support	medium	medium

7. Risk Analysis

List factors that could pose a risk to the project's success, assess their likelihood and severity, and how you will prevent them from happening (or manage them if they if they occur). Cover the types of risks listed and any others that apply.

Risk	Probability (1-5)	Severity (1-5)	Score (P x S)	Action to Prevent/Manage Risk
Staffing (Staff leaving because of short-term research contracts).	1	2		Where possible allocate existing university research staff.
Organisational (Keeping to timescale and meeting milestones)	2	3		Project advisory group to meet with the project team once a month.
Technical (technical problems with the institutional infrastructure which inhibits development)	1	5		Involve a representative from central computing on the project advisory group
External suppliers (Communication and management)	2	4		Online project management module with discussion boards and project logs to manage communications with sub-contractors
Legal complications	1	3		University Solicitor to draw up contracts with the specified sub-contractors.

8. Standards

List any specific standards you will adopt and why they are important.
The system will be initially designed as a standalone tool that can be loosely coupled with any online learning environment. In addition close coupling will be implemented with the Blackboard VLE via its 'Building Blocks' APIs. However through the provision of the source code this close-coupling should be possible for other key platforms.
In terms of specific standards the project will work towards implementing the following standards to promote interoperability:

- IMS Question and Test Interoperability specification to represent the test data.
- the IEEE standard for Learning Object Metadata to describe the 'programming assessments' implemented through the tool.
- XML and WSDL for web services development
- relevant JISC e-learning frameworks identified in project review/scoping study.

9. Technical Development

Indicate how the project will follow best practice for technical development, and any specific technologies or development approaches the project will adopt and why.

This section describes the requirements capture and analysis, development, and evaluation stages in the proposed project plan. In accordance with the three-tier architecture presented in the JISC technical framework, the development of the ASAP tools is split into three parts: Application services, User Agents and Common Services.

A first-pass system design suggests a core repository of test data and metadata, containing all necessary information *for* the test (student instructions, test scripts, auxiliary source code etc), and *about* the test (mark allocation, acceptable dates etc). This design is drawn in Figure 1. The numbered boxes represent the ordered set of processes associated with the live-cycle of a unit test.

The requirements capture stage finalises the requirements of the system proposed here. These are manifested as the the set of functional / non-functional requirements for each part of the implementation, and the xml schema (or database design) for the test repository.

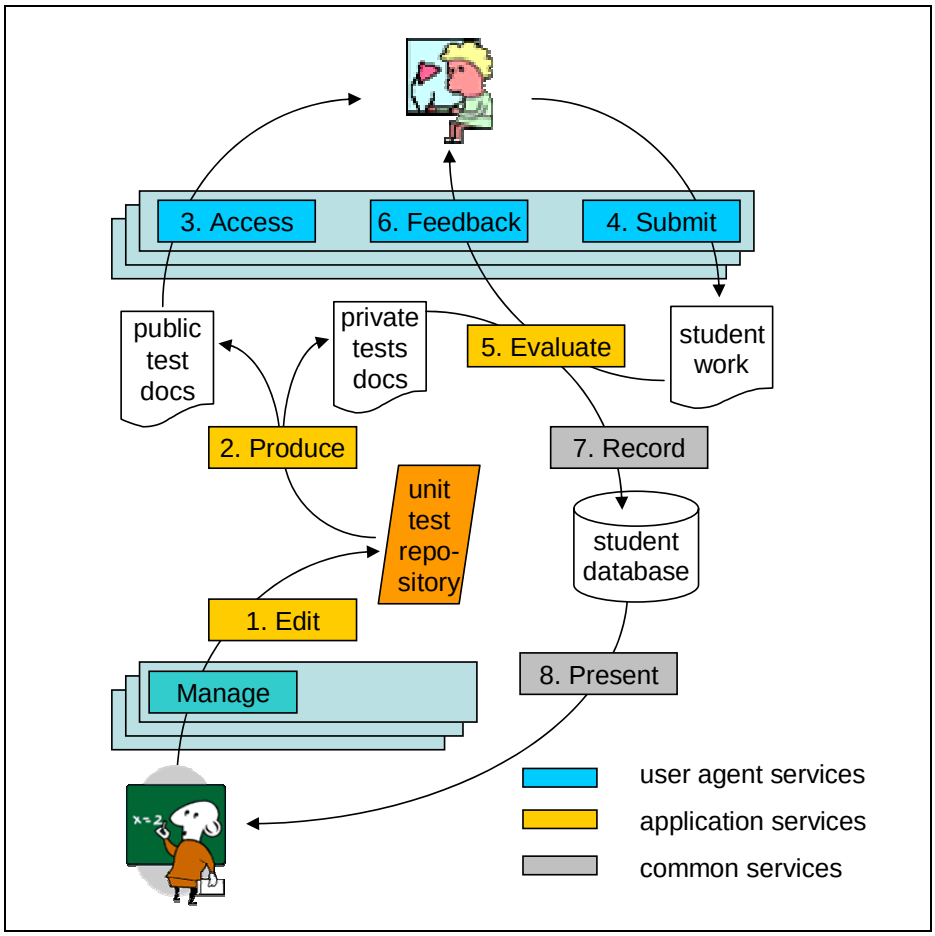

Figure 1: Flow chart showing the life cycle of the unit test. To start, the instructor edits (or creates) the unit test repository (1) , and creates the necessary test documents (2). The public documents are accessed (3) by the students before attempting the assignment. Their solution is submitted (4), evaluated using the private unit test scripts (5). The result communicated to the student (6) and the database of student marks (7), which can be accessed by the instructor (8).

The development of the Application Servcies and User Agents is split into three separate work-packages. The first workpackage provides the basic application services. An initial design analysis suggests the need for three such services: the first to edit the test repository, second to generate the

public and private test files from the repository, and a third to use these files to perform the test on the solution submitted by the student, and provide feedback and marks.

The Roboprof and JPlag work can extend the second and third application services respectively. These extensions are included in a separate work-package, 'Advanced Application Services'. The user-agents are developed in a separate workpackage and targeted at two VLE's and generic browser access.

10. Intellectual Property Rights

List any intellectual property owned by third parties that will be incorporated into project outputs, when/how you will obtain permission to use them, and any implications for project outputs after the project ends.

'Roboprof' and 'JPlag' are *background* Intellectual Property owned by academics and the institutions (Dublin City and Karlsruhe, respectively). These parties have agreed to make the use of these software available during and beyond the project lifetime. Intellectual Property rights are retained by the respective owners and these components are not included in the public release of tools.

Project Resources

11. Project Partners

List all project partners (including subcontractors), their roles, and the main contact. Indicate the date a consortium agreement was signed (or will be signed), and send a copy to the programme manager.

List of project partners (all subcontractors apart from the main partner.)

Project partner	Role	Contact
Kingston University	Main partner	David Livingstone
Karlsruhe University	Development and Consultancy for the JPlag components	Guido Malpohl
Dublin City University	Development and Consultancy for the Roboprof components	Charlie Daly
De Montfort University	Use and Evaluation of tools and services	Richard Hall
City University	Use and Evaluation of tools and services	Jo Wood

Consortium agreement is not believed to be necessary as all subcontracts can be self-contained.

12. Project Management

Briefly describe the project management framework, including organisation, reporting relationships, decision process, and the role of any local management committee.

List all members of the project team, their roles, and contact details. Indicate the proportion of time the project manager will spend on project management.

Indicate if the project has training needs and how they will be met.

The project will be led by Kingston University. From this institution, a project advisory group will be formed, including the principle investigators, the Head of School of Computing, the Associate Head of the Academic Development Centre (Educational Technology). Three meetings will be held over the course of the project (every three months) with subcontracting parties and consultants. Action points will be agreed through consensus and reviewed monthly. Decisions can also be taken through appropriate communication via email.

At Kingston University the following staff will comprise the advisory group:

Dr. James Orwell		Technical Manager	j.orwell@king.ac.uk
David Livingstone		Project Manager	d.livingstone@king.ac.uk
Prof. Tim Ellis		Head of School	t.ellis@king.ac.uk
Dr. Tim Linsey		Consultant	t.linsey@king.ac.uk
Guido Malpohl		Consultant	

13. Programme Support

Indicate if there are specific areas where you would like support from the programme or programme manager.

Some assistance with ensuring development is compliant with JISC Technical Framework would be welcome.

14. Budget

Use the budget template and attach the project budget as Appendix A. Explain any changes from the budget in the agreed project proposal.

Detailed Project Planning

15. Workpackages

*Use the workpackages template to plan the detailed project work and attach as Appendix B. Clearly indicate project deliverables and reports (in **bold**), when they are due, phasing of workpackages, and explain any dependencies. You may also attach a Gantt chart, diagram, or flowchart to illustrate phasing.*

WP1	Project Management. Procurement of equipment, management of staff, liason between partners, administration of budgets, organisation of meetings, timely delivery of deliverables, etc.	2 man months
WP2	Requirements Capture. Functional requirements definition for each stage of the implementation (WP3-5), schema for unit test repository.	4
WP3	Application Services. Development of web services for core functions editing, producing and using the unit test material.	6
WP4	Advanced Application Services. Integration of RoboProf and J-Plag functions into the Application Services.	6
WP5	User Agent Services. Implementation of user-interface for at least two ADSFs, e.g. BlackBoard and Web-CT.	4
WP6	Content. Creation of unit tests for at a minimum of two modules.	1
WP7	Dissemination and Publication. Submission of academic papers to appropriate conferences, release of software onto a suitable portal (e.g. sourceforge), advertisement of service to Higher Education community.	1
Total		28

Table 1: List of Workpackages, and man months allocated to each.

	total mm	0	1	2	3	4	5	6	7
WP1 (Management)	2								

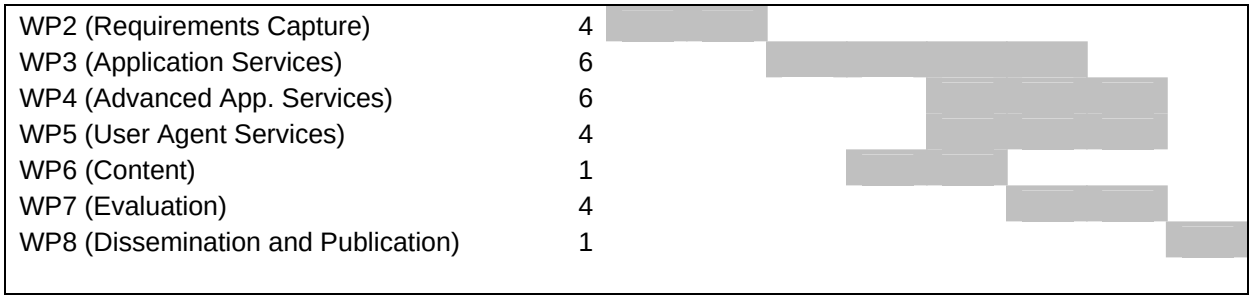

Figure 2: Activity chart over the duration of the project.

16. Evaluation Plan

Indicate how you will evaluate the quality of the project outputs and the success of the project. List the factors you plan to evaluate, questions the evaluation will answer, methods you will use, and how success will be measured. Expand as appropriate on how you will conduct the evaluation.

To be determined in consulation with stakeholders nearer to project commencement date.

Timing	Factor to Evaluate	Questions to Address	Method(s)	Measure of Success

17. Quality Assurance Plan

Explain the quality assurance procedures you will put in place to ensure that project outputs comply with JISC technical standards and best practice, and what will constitute evidence of compliance.

To be determined in consulation with stakeholders nearer to project commencement date.

Timing	Compliance With	QA Method(s)	Evidence of Compliance
	Fitness for purpose		
	Best practice for processes		
	Adherence to specifications		
	Adherence to standards		
	Accessibility legislation		

18. Dissemination Plan

Explain how the project will share outcomes and learning with stakeholders and the community. List important dissemination activities planned throughout the project, indicating purpose, target audience, timing, and key message.

Timing	Dissemination Activity	Audience	Purpose
End of	Conference publications	Educational	Share good

project		Academics, journalists	practice
End of Project	Demonstration of use	Educational Academics and journalists	Encourage take-up of tools
	Electronic Publication of tools	Edecational academics, technicians	Allow third-party use of tools

19. Exit/Sustainability Plan

Explain what will happen to project outputs at the end of the project (including knowledge and learning). Focus on the work needed to ensure they are taken up by the community and any work needed for project closedown, e.g. preservation, maintenance, documentation.

Project Outputs	Action for Take-up & Embedding
Tools for support of automated assessment	Continued availability from project website
Source code for tools	Maintenance and improvement to tools via shared source code repository, e.g. SourceForge.

Project Outputs	Action for Take-up & Embedding	Action for Exit

List any project outputs that may have potential to live on after the project ends, why, how they might be taken forward, and any issues involved in making them sustainable in the long term.

Project Outputs	Why Sustainable	Scenarios for Taking Forward	Issues to Address

Appendixes

Appendix A. Project Budget

Appendix B. Workpackages