

## Toolkit & Demonstrator Final Report Template (Cut-Down Version)

This template is very loosely based on the JISC Final Report Template available from the [JISC Project Management Guidelines](#) web page.

<b>Project name/acronym:</b>	<b>CATS</b>
<b>Project website/blog address:</b>	<a href="http://assessmentbanks.info">http://assessmentbanks.info</a> <a href="http://sourceforge.net/projects/catsproject">http://sourceforge.net/projects/catsproject</a>
<b>Report, author(s):</b>	Jim Everett
<b>Contact person</b> (if different from above):	Jim Everett Learning Services Alexander Turnbull Building, University of Strathclyde, 155 George Street, Glasgow G1 1RD Jim.everett@strath.ac.uk 0141 548 2186
<b>Date:</b>	September 2006

### Methodology

*Provide information on the methodology your project used (if this was different from your original project plan, please provide the reasoning for this).*

#### The CATS system

The CATS project developed an exemplary system for harvesting assessment items from item bank systems on the basis of an assessment request profile that not only comprised the subject domain query and the target assessment bank, but also included additional "pedagogical" parameters, i.e. the number of items to be retrieved and the algorithm to be used to traverse the target assessment bank. The resulting collection of assessment items would be packaged according to the content/assessment packaging specification indicated in the profile and then written to a files structure, zipped, etc. according to the profile request.

Although a functional system, the CATS system was conceived as a "proof of concept" demonstrating how item banks might not only be queried using web services but the items actually harvested. The design was deliberately extensible and explicitly facilitates the creation of new "flavours" of the services. This was most evident in the connections between the CATS system and item banks. The potential variability in the interfaces offered by assessment banking systems requires a special connector for each system that is able to translate between the common language of the assessment request and the particular language and format required by the assessment bank. Beyond accommodating the development of new item bank connectors, the system permits the development of new packaging services for new and different content/assessment packaging standards or the development of new writing services that could populate assessment management systems.

The architecture also allowed the incorporate of new steps in the harvesting process; among the options discussed was a service for comparing the list of items returned against existing previously created collections to avoid duplicating the same assessment pool or the incorporation of a step where the user was able to preview and screen the collection of items returned before submitting the to the packaging and writing services.

#### The Services

Beyond a coherent functioning system, CATS was conceived as a toolkit of services that could have a wider application. The project took services developed for previous projects and extended their functionality making them more amenable to use in new context. For example, the SPAID zipping service could only handle one assessment item; through CATS the zipping service was extended to process not only multiple assessment items, but also the items' dependent files (e.g. images) in a complex hierarchical files structure.

Similarly, the D+ services were developed to search across data targets and return pointers/references to records that matched the query criteria. Through CATS the service was extensively modified so that, rather than simply returning a reference, it retrieved the actual item from the assessment bank along with dependent files. These retrieved items are returned in an organised stream allowing the file structure to be (re)created.

**Scoping and needs analysis**

The CATS Project was informed by a vision of a time when very large numbers of assessment items are widely accessible to practitioners through multiple item banking systems. As a vision it is close to being realised, however the implications of such a transformation in the availability of interoperable assessment items are not well explored. The CATS Project sought to contribute to this exploration. The development of the CATS system was a way to demonstrate what could be achieved as the item banks became available and as a way concretise the discussion of item bank harvesting, therefore, instead of an abstract discussion of how item banks might be used, there could be a pragmatic scoping of practitioner needs informed by a real system.

The scoping study was pursued at two levels. The first was to consult with experts in the field of e-assessment systems as, given that this exercise was future-focused, these domain experts would already be sensitised to the issues. The second was to elicit the views of the wider community of practitioners. Recognising that these were new issues for many to consider, the approach was to open a dialogue and offer respondents to space to articulate their responses in a guided but open environment. For the second group in particular, it was important to initiate the dialogue with a presentation and demonstration of the CATS system.

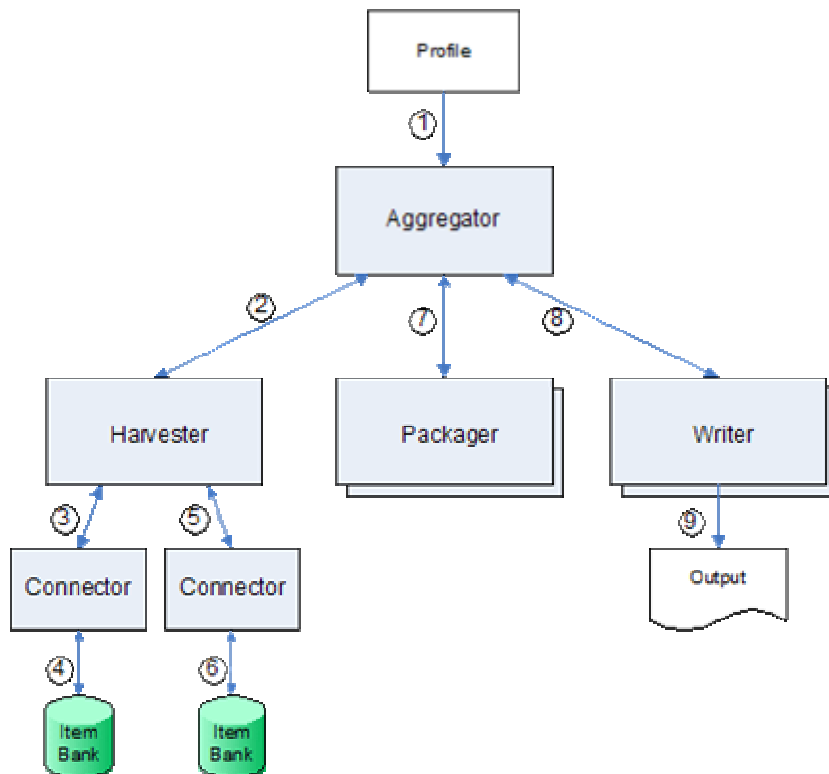
**Implementation**

*Summarising the overall approach taken, technical design, development etc. Tell the story of what you did rather than listing workpackages.*

**Technical Design**

The design and structure of the CATS system remained largely constant throughout the active development phase of the project. The design philosophy was to create a constellation of services where

- each services performed one iteration of a single tasks
- services communicated with a single coordination service (the Aggregator) rather than in a linear sequence



A major attraction of this approach was the flexibility it offered the greatest flexibility in adding not only new instances of services, e.g. alternative connectors or packagers, but also new services. In each case the new options were simply specified in the assessment request profile parameters and tracked by the Aggregator coordination service.

Initially BPEL had been identified as an approach to developing the coordination service; however this was rejected because of the current limitations in the handling of attachments. The need to process varied and complex attachment structures also proved something of a challenge for SOAP 1.1, but effective and widely adopted approaches were found to accommodate this requirement.

The assessment item bank used in the project was the item banking service developed by the JISC SPAID project. This system could already support the harvesting of single items and, as an open source system, could be adapted to support the harvesting of multiple items through SOAP with attachments. The project did explore other item banks, particularly E3AN and Questionmark Perception, however in neither case was it possible to develop a similar harvesting functionality within the constraints of the CATS project.

As a result of the use of the SPAID item bank, the CATS systems used IMS QTI 2.x assessment items; however the system would also work with previous versions of the QTI specification. The packaging service implemented IMS CP v.1.1.4, however alternative content packaging services could be developed and specified.

### **Design, Development and Project Management**

CATS employed a modified and simplified version of the JAD and SCRUM project methodology employed by the SPAID project.<sup>1</sup> The adjustments were made in light of lessons learned from the SPAID project and in the new conditions faced in the CATS project.

At the heart of the methodology was the initial face-to-face Joint Application Development (JAD) meeting. This incarnation remained true to the spirit of the methodology although with minor adjustments. The meeting included all the active participants in the CATS project as well as contributions from members of the expert advisory group who were able to offer particular perspectives and address specialist issues. The meeting was chaired by Project Manager, however all participants took an active part.

Considerable upfront effort was spent through the JAD meeting on the initial design of the services and system architecture. While the technical issues were not inconsiderable, significant effort was spent on ensuring that all the participants and the programmers in particular had a shared understanding of all the terminology and concepts underpinning the project. For example the interaction between the connector, harvester and aggregator services while ultimately quite straightforward was the focus of quite intense and in depth discussion as we tried to marry existing systems (SPAID and D+). This effort and determination to achieve a verifiably shared understanding was vital to the success of the project.

The JAD produced

- an explicit agreed understanding of the terminology and development strategy
- an overall architecture for the whole project based on that agreed understanding
- an initial breakdown of the project into its principal tasks
- an initial identification of core and “nice to have” tasks
- an initial organisation of the tasks into development “sprints”

The principal artefact produced was a series of annotated flip charts, which were photographed at high resolution as a permanent record. A designated note-taker also produced written notes with the assistance of all participants.

The conduct of the JAD deviated from the methodology in the absence of a neutral facilitator. The JAD was to have been facilitated by Mhiari McAlpine (SQA), one of the expert advisory group and sponsor of this project management approach in the SPAID project; however she was unable to fill the role. However the choice of the Project Manager as a replacement did not compromise the methodology as all the participants were new to the endeavour and there were few entrenched positions to defend. Another break with the development approach was the choice of a note-taker who was also

---

<sup>1</sup> See the SPAID deliverable, "Summary report on the use of Joint Application Development (JAD) & SCRUM methodologies" <http://www.learningservices.strath.ac.uk/spaid/>

an active participant in the discussion; again a pragmatic decision as the team was so small which had no negative on the outcome.

After the meeting a formal report was produced that further clarified the decisions of the JAD.<sup>2</sup> The report also included a complete development task list<sup>3</sup> indicating

- tasks
- the person responsible for that task
- an initial time estimate for the task
- the dependencies between tasks

Tasks were organised into five sprints (0 to 4) based on

- the dependencies between tasks
- the project resources available to complete the task (in particular the availability of programming time from the D+ project)
- key milestone checkpoints for the project

The task list also highlighted areas that were identified as needing further research and examination or where it was predicted the task might turn out to be more difficult than hoped.

The resulting sprints corresponded closely to the project management archetype, i.e.

- they were time bound
- had tangible results
- resulted in a “build” will testable functionality

Where they deviated from the archetype was principally in their varying length. While sprints should all have the same duration, CATS sprint varied in length. This was a pragmatic decision taken so that the development of each of the CATS services could be captured in a single sprint. Because some of the services were identified as more complex than others they required more time. While violating one of the “rules” this decision proved to be quite effective and had the added benefit of ensuring that sprint releases possessed a real coherence.

Each sprint began with a formal face-to-face meeting of all active project team members to review the tasks for the sprint and ensure

- that they were still valid in the light of previous work
- that all the requirements and dependencies had been satisfied to begin on the sprint
- that the resources were available to complete the tasks

The meeting also validated the timescale of the sprint and the expected outcomes, typically described as verifiable and testable artefacts. It was at this point that the tasks in the “nice to have” category were formally reviewed for inclusion on the upcoming sprint, thus making sure that these tasks were not deflecting developers from the key tasks while ensuring that they were not lost sight of.

The sprint was brought to a close with a formal face-to-face meeting. The meeting reviewed all the tasks and outputs allocated to the sprint and received reports on

- progress towards the outputs
- deviations from the original sprint plan
- decisions made that would impact on future tasks
- any activity on tasks originally outwith the scope of the sprint

Where necessary the meeting would agree minor additional steps to be carried out before the sprint was closed and the build (where appropriate) released. In most cases the meeting would then turn to consider the next sprint as described above.

In between the sprint meetings weekly progress reports were required from the programming team and others who were responsible for active tasks during the sprint. The reports were specified as a presentation of achievements and highlighting of barriers encountered rather than a simple activity report. These were to be gathered orally or via email in an informal environment. Their primary function was to ensure that barriers could be addressed promptly without waiting till the sprint meeting.

---

<sup>2</sup> Available from CATS website under Downloads

([http://assessmentbanks.info/index.php?option=com\\_docman&task=cat\\_view&gid=23&Itemid=29](http://assessmentbanks.info/index.php?option=com_docman&task=cat_view&gid=23&Itemid=29))

<sup>3</sup> Available from CATS website under Project Management Documents

([http://assessmentbanks.info/index.php?option=com\\_docman&task=doc\\_details&gid=1&Itemid=29](http://assessmentbanks.info/index.php?option=com_docman&task=doc_details&gid=1&Itemid=29))

## Quality Assurance

The quality assurance methodology was based on the development sprints. The alignment of the sprints with the architecture structures rather than to a fixed period of time was a significant benefit as it meant that the sprint outputs were more amenable to quality assurance and testing. Quality assurance was undertaken at three levels; programme code, system and project.

At the programme code level Quality Assurance was undertaken by the two members of the programming team. This was facilitated by the division of labour and working practices adopted by Keith Davidson (Octavia Consultants) and Robin Taylor (D+, University of Edinburgh), i.e.

- they (largely) worked on distinct aspects of the development
- they worked independently at remote locations
- they had never worked together before

As each sprint drew to a close Keith and Robin would share the code that had been developed. There followed a iterative process of quality assurance, explanation, evaluation and improvement as each examined, tested and sought to understand the code produced and how it would all work together.

This interchange was particularly effective at identifying obscurities in the code and highlighting potential barriers. It was also an effective way to stimulate the development of proactive solutions.

At the system level quality assurance was undertaken by Iain Tulloch (University of Strathclyde) who maintained oversight of the code development. This, coupled with the regular sprint meetings where the code development was presented in some detail to the whole project team and actively discuss, helped to ensure the usability of the code being developed. It provided a way to test the rationale and approach being adopted beyond the actual testing of the code itself. For example it was through these discussions that the decision to drop BPEL as the orchestration language was taken at a project team level at an early stage in the project.

Quality assurance did not only include the programme code and system design, there were many other deliverables, some but not all outlined in the JISC Quality Assurance and Open Source Maturity Model. These project level tasks and deliverables were also included in the sprint tasking and thus reviewed at the regular sprint meeting. These deliverables, such as the project website, blog, etc., were therefore presented and discussed formally by the whole team and where necessary additional tasks were agreed.

## Outputs and Results

*Include information on where the outputs and results for your project can be found.*

The outcomes of the project have been made available from the main project website,

<http://assessmentbanks.info>, and the linked sourceforge site,

<http://sourceforge.net/projects/catsproject>.

## Implications and Recommendations

*Consider the future implications of the work undertaken, and how others can build on it – to include details of any new development work that could be undertaken to help carry it further.*

### Scoping Study

The scoping study indicates that the community, including both practitioners and domain experts, has only just begun to explore the issues that will arise from the development and availability of question item banks. In this new environment we can expect a transformation in approaches and perceptions comparable to the changes in information literacies brought about by the internet information revolution. Beyond exploring the technicalities of item banking and the availability of assessment items there is a need for research into the wider cultural, educational and organisational changes that will be required to derive the greatest value from these banks. However, it has proven difficult to stimulate this debate in the absence of large scale implementations of harvestable item banks.

### Reuse of services

Several of the component services have a functionality that can find general application. In particular the zipping and writing services, with their ability to process structured hierarchical attachments, could be applied in a number of other contexts. The content packager is also applicable to other contexts within the E-Learning Framework. The development of the D+ search into a harvesting tool was identified at the outset as a highly reusable service. The CATS harvesting service itself, with its ability to manage complex hierarchical data streams, may not be the best choice for other contexts where the items being harvested are simpler; however CATS has demonstrated a clear development path from the original D+ service.

### **CATS system**

Consultation identified several areas in which the current proof of concept CATS systems could be developed to make it a production level tool. At the simplest level these involve the creation of more connector, packaging and writing services. More complex is the development of more search and harvesting algorithms, in particular algorithms to ensure the creation of assessment item pools different from those already generated and for the incorporation of usage data in the selection criteria.

The need for human interface clients has been clear from the inception of the project. The current system is explicitly a batch process, although a simple interface has been developed. The scoping study identified a clear demand for a more interactive and iterative system that would allow users to preview and fine-tune their assessment request profiles. Future clients should facilitate the development and refinement of the assessment request profile with a combination of constrained choices, free text and static elements that will allow users to specify complex requests in simple steps. Previewing harvested items is also possible and exploratory discussions have taken place with the R2Q2 team at the University of Southampton with a view to incorporating R2Q2's services that render questions to create such a facility.