

## **EVALUATING REPOSITORY SOFTWARE**

### **How long did I spend on the evaluating phase?**

Approximately four months were spent on activities associated with evaluating repository software (though not on a full-time basis).

### **What were the deliverables from this phase?**

- Criteria for evaluation
- Evaluation and feedback on repository solutions
- Evaluation Report, including the way to proceed with the selected solution, i.e. next steps
- Selection of repository software

### **What did I do?**

- Sourced documentation on the identified products that were being evaluated
- Identified available reports on repository software evaluation
- Determined the criteria for evaluating the product
- Developed an evaluation criteria checklist
- Carried out evaluations against the criteria on a hosted communal instance
- Investigated any possible limitations within the systems
- Documented problems and issues encountered
- Follow-up with technical staff on issues raised
- Investigated configuration possibilities for the software
- Identified basic workflow processes
- Added metadata records to the repository
- Organised data migration into the repository to further testing
- Sort advice from Technical support staff (e.g. asked a lot of questions)
- Reported on the evaluation phase
- Gave presentations and talks on the potential solutions to my own organisation

### **About the process**

At the beginning of the evaluation stage I spent some considerable time sourcing existing repository software evaluations that had been published or made available on the web. I then reviewed these to determine if they may be useful and relevant to inform my evaluation activities and also for the possibility for potential use as a template. Ultimately I developed my own checklist however the existing evaluations that I had located were extremely useful in identifying what the main elements and issues were.

Determine if you are evaluating the user application and the system requirements, or if you have IT assistance to assist with the technical aspects including the system specifications.

## Considerations/Tips for Evaluation

- Existing or future relationships with vendors (or products) that may influence your decision in software selection, e.g. you may already have other systems by a specific vendor which may influence your decision
- Costs: Software/hardware - licence and maintenance fees (ongoing), upgrade costs, development costs, server (pending specs).
- Open Source – versus – Proprietary software. Developments, support, ongoing costs.
- Support for the repository once implemented – the level of support you require versus the support available to you in your institution. This also includes the support provided by the software vendor, or for open source software, the level and nature of support available from user groups, etc.
- Technical – the level to which hardware and software specifications are compatible with your institution's systems and IT environment.
- Business requirements, e.g. the problems that the repository will address, solve or provide solutions to.
- Development of your criteria/checklist/requirements that can be used to assist your evaluation of products, and software selection.
- Identifying standards, e.g. OAI-PMH, etc and interoperability with other systems that need to be complied with.
- Evaluation/test environment. Establishing pilot/test version of software solutions at your own institution is highly recommended, or having vendors provide access to test environments that you can access.
- Exit strategies? Considering and identifying how you will get your data out the system is just as important as considering how to get your data/content into the repository. This is critically important for future migration to new software solutions in the future.
- Seeking technical advice from your IT staff.
- Presentations from vendors can provide a good starting point and provide a contact for follow-up questions.

### **Project Manager's Tip**

To evaluate repository software solutions you need to have a clear understanding of what the business requirements are for your repository prior to evaluation, e.g. what problem is the repository solving, what issue is the repository addressing for your organisation, or what service/function is the repository delivering?

Evaluations can focus on technical elements: specifications, standards, limitations, harvesting, metadata schema, infrastructure requirements and integration with other systems within your organisations or relationships with external systems. User needs can focus on the functionality required of the system, workflow processes including ingest, review and approval, customisation. Management issues can focus on data migration, access and discoverability relative to the drivers associated with your repository.

Most importantly the time for considering and identifying an exit strategy for the repository is at the time of evaluation and before any decision is reached on a solution. You need to know that the data in your repository can be migrated to another repository sometime in the future if need be and what is involved in that process including any limitations, loss of data, workload or financial issues that may arise as a result.

## **What Worked?**

Sourcing and identifying specific criteria to assist with the evaluation of the repository solution assisted to make a much more informed decision about potential software. Talking to other universities or organisations about their experiences with their implemented systems also provides valuable insight into issues that may arise post-implementation that may be unexpected or unplanned for.

To evaluate collaboratively we (multiple universities) were able to access a 'communal' instance of a specific repository solution. This environment was extremely beneficial. Evaluating products collaboratively with other universities worked really well and delivered a far more comprehensive evaluation that could have been achieved alone. Evaluating and testing collaboratively also enabled issues to be identified quickly.

## **What Could have Worked Better?**

It was my experience that there are two major dimensions to the evaluation of potential repository solutions. These being: system functionality and user acceptance and application. Knowledgeable technical staff are needed to partner in the evaluation of the software particularly in respect to system functionality, standards, interoperability, batch ingest, configuration, etc.

When implementing our repository there were questions raised by project administrators and technical staff about the role of the Project Manager/implementer undertaking evaluations of products. It was felt that this was a role for IT technical staff.

I believe it is imperative that Project Manager's and implementers fully understand and are able to reason the decisions (software solution) that they are recommending to their organization. I don't feel that it would be feasible to recommend a solution without thoroughly evaluating it at the organization level. The features, functionality, workflows and management issues are identified through evaluation of the product and it's important to have a complete understanding of the product to make effective decisions relative to ongoing sustainability and support that will be required by the service area.

## **What Did I Learn?**

This is where I fully came to understand and appreciate the potential and power of collaboration.

Evaluating software is not solely about the technical issues but importantly also about the management and workflow issues that are identifiable through the evaluation process.

Five major questions you need to answer before committing to a solution:

1. Are the workflows and processes associated with this solution sustainable for my organisation?
2. Is this system scalable for future growth, services and functionality?
3. Can I exit this system and seamlessly move/migrate my content to another solution in the future?
4. Is this system compatible with my organisation's Information Technology environment as well as internal and external interoperability requirements?
5. Can we meet the support requirements of this system?