

RUBRIC Toolkit: System Options

Choosing a Repository Solution

Implementing a repository solution is not a purely technical issue.

When selecting your software, it is useful to begin with a table or features matrix comparing feature requirements against software functionality. Consider whether the software is a strategic decision for the organisation and whether this impacts on the decision making process.

Reflection on evaluation may be useful in seeing how the University of Newcastle undertook this phase of the project.

This section reviews some of the options for evaluating systems, making a decision and gaining acceptance in an institution. Consider:

- **The institutional context**
 - institutional requirements for software procurement may dictate processes for evaluation. Software solutions may have to go out to tender
 - the institution may have a commitment to a particular operating system
 - there may be an institutional mandate for or against outsourced or hosted solutions
 - policies regarding free and open source software will vary in different institutions
- **Evaluating the software**
 - a functional checklist is useful
 - the RUBRIC Project evaluated software in a series of time-limited cycles of development (outlined in the Case Study)
 - different solutions will require different hardware for the data and access loads expected; this is usually a matter for institutional IT services. This section provides some considerations for this process.
- **Evaluating support services**
 - commercial software and hosted services usually include some form of maintenance or support
 - open source software usually has mailing lists, forums and wikis where users may share the load of support with the software developers
 - it is imperative to evaluate the overall effectiveness and suitability of the solution relative to your institution's resources
 - do not assume that commercial support is better than the support available from open source communities or user groups for commercial software
 - commercial arrangements usually carry a guaranteed service level agreement

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

- **Planned obsolescence**
 - repositories need to preserve data for a long time, perhaps beyond the life of the current software available
 - any repository should be able export data to ensure the protection of that data
 - develop an exit strategy for the solution chosen
 - ensure you have an upgrade and migration path
 - it may be worth starting with a simple repository for the short term and plan for a time when a new solution is available
- **Consider having more than one software solution** for different repository types
 - if a collection has special needs, such as an image collection, it may be preferable to implement different software to manage it
 - use your software comparison table to find the best solution for your needs
- **A strategic approach**
 - the 'best' software in a technical sense may not be the one to use
 - initial costs might be reduced by choosing an open source solution
 - risk might be reduced by choosing a commercially supported system

Evaluating Software

Institutional requirements may dictate the software evaluation process. Thorough evaluation of possible solutions is a valuable exercise which may help to prevent launching an inappropriate solution. Evaluation time needs to allow for the following:

- installation of potential repository solutions
- presentation of potential solutions to appropriate committees / groups
- training in potential solutions
- testing of potential solution
- development of documents associated with evaluation

The evaluation phase may require several weeks, depending on staffing resources available.

An Evaluation Checklist is available to assist with this process.

If the institution is unable to undertake a full evaluation in-house, it is possible to gain some understanding of various solutions by looking at other evaluations:

- [Guide to Institutional Repository Software](#)¹ by the Budapest Open Access Initiative
- [Repository Software](#)² section on the RUBRIC website has information in an Australian context on DSpace, ePrints and Fedora based repositories trialled during the RUBRIC Project

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

- [Notes about possible technical criteria for evaluating institutional repository software](#)³
- [Technical Evaluation of Selected Open Source Repository Systems - a comparison of Eprints, DSpace and Fedora](#)⁴

A specific test script can be useful once the solution has been chosen. These examples are provided by Massey University for testing:

- Administrator
- General User

Hardware Requirements

Most repository software will provide indications on base requirements for hardware on which to run (recommended CPU, RAM, disk space).

Most institutions will have policy or preferences regarding hardware platforms which will need to be taken into account as part of the software selection. This may not be negotiable, but should not greatly affect the type of repository chosen. The Operating System could limit the software solutions evaluated, as not all solutions run on all operating systems (for example Linux versus Windows).

Virtual infrastructure might be a useful option for running the IR software. This decision may not be under the sole control of the repository manager, you will need to consult with your IT division.

[VMWare's ESX Server](#)⁵ was used by RUBRIC for its infrastructure.

A [technical paper](#)⁶ is available on the RUBRIC website to explain how this was deployed.

Data Migration

The Data Management section explains what is involved in identifying and loading data from other sources.

Streamlined data migration may save hours of staff time.

A [migration kit](#)⁷ has been developed by the RUBRIC Project to assist with this process.

Systems which may contain potential data for migration into an Institutional Repository include:

- institutional research systems
- library catalogues
- other repository-type systems

The migration kit has been established around the following transformations:

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

- data is taken from its source in whatever textual format is possible (as close to XML as possible)
- XSL and the Python programming language are used to convert the data into a Dspace-style Simple Archive format
- each object being migrated has its own Dublin Core XML file and a contents file (listing all data streams), along with all data streams for that object
- objects may be ingested into Dspace, or undergo a further transformation using XSL and Python to a Fedora-style format suitable for ingest into a Fedora-based repository system.

The RUBRIC migration toolkit recommends that a number of small scripts provide the greatest flexibility for technical staff undertaking data migration tasks.

Some products such as VITAL and Fez provide end-user configurable utilities which are inevitably limited in their flexibility, but can be supplemented by the RUBRIC migration kit. Python gives the RUBRIC toolkit the full power of a general purpose programming language.

Additional documentation and examples on data migration that may be of interest include:

- [APSR Project Reports](#)⁸ website
- [Migrating ePrints data to a Fez repository](#)⁹

Licensing and Costs

There are costs associated with setting up a repository even if Open Source software is selected. Costs may include:

- licensing (commercial solutions)
- maintenance (commercial solutions)
- server hardware
- other software:
 - consideration needs to be given if you will be running your repository with a proprietary database system, web server, etc
 - some software relies on other third party software that may be distributed under different licenses
 - some software licenses include an initial purchase price and maintenance costs
- staff time, including:
 - technical staff to set up the infrastructure
 - technical staff to maintain the infrastructure
 - editorial staff to proof objects
 - staff to be on call to assist users in using the repository
 - administrative staff involved in the running of the repository

- marketing costs – flyers and advertising within your institution
- upgrades and backup services
 - services from another department may be required to undertake the work
 - some upgrades include an installation fee

Create a full cost overview for each potential solution during the evaluation phase to ensure your comparisons are complete and to help inform further decisions.

An institution may deem a solution appropriate or inappropriate based on cost or resource availability.

Case Study: the RUBRIC Evaluation Process

The RUBRIC project is a special case where a dedicated central team was servicing several partners, so the approach taken needs to be adapted for more general use. This section describes the software evaluation approach taken by RUBRIC Central.

Initial Scope

The scope of initial investigation was set by the RUBRIC Project Board covering documents in two broad categories:

- theses & dissertations, including masters and PhD theses and may or may not include undergraduate (Honours) theses, and major reports and/or essays
- research output, including published research in the form of pre-print, re-prints or author's drafts, and other non-peer reviewed materials such as working papers or reports

RUBRIC Central evaluated three software products to determine their suitability. These included:

- [Dspace](#)¹⁰ recommended by the [APSR](#)¹¹ project
- [Fez](#)¹² recommended by the [APSR](#)¹³ project
- [VITAL](#)¹⁴ recommended by the [ARROW](#)¹⁵ project
- [specific packages evaluated](#)¹⁶ is available on the RUBRIC website. This will be updated until the end of 2007 and will then become of historical interest only.

It is noted in the section on lessons learned (see below) that this tightly constrained scope was a limiting factor.

Cyclical Evaluation Process

A cyclical process similar to an [agile development methodology](#)¹⁷ was used for evaluations.

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

An [ePrints repository](#)¹⁸ belonging to the University of Southern Queensland (USQ) was used as a data source for test records.

Using this data, each software product was used to produce a test IR, and initially tested in a two week 'sprint'. Starting from scratch, two technical staff installed the software, set it up and imported sample records.

Feedback on the deployment of all three test IR's was discussed with project partners.

More cycles of three-weeks duration studying each test IR in turn gave the project partners a chance to use each repository solution with real documents from their institution. Customisations were also applied.

This extract from an internal RUBRIC report written before testing began provides an overview of the strategy applied:

There are 5 partners in the nuclear RUBRIC group, and 3 repository software solutions currently recommended by FRODO projects as best practice.

The RUBRIC technical team is proposing to implement three demonstrator systems which will be available for all partners (and their partners) to try. A shared system has some limitations and it may be possible to host up to 3 demonstrator systems for each partner however some testing will need to be done before this can be confirmed.

This would mean up to fifteen systems if all partners wished to explore all options. However, it seems likely that as projects will proceed at different rates in different partner institutions and all partners will be able to observe progress of the others that it will not be necessary to install this number of software instances.

The function of these demonstrator systems is to:

- give technical and library staff experience with repository software and its configuration interfaces
- provide a branded system or systems that can be shown within a partner institution as part of planning and development
- make a start on configuring an institutional repository ready for the deployment phase of a repository, which is likely to happen in late 2006 or early 2007 in most cases

It must be recognized that this is a technical exercise that will not replace detailed project management within an institution.

These systems are:

- not full pilot implementations
- may be limited to tens of documents per partner (not hundreds or thousands) as they will be tightly time-constrained

A methodology for setting up some branded demonstrator systems involves three-week cycles of effort, staggered by institution, along the following lines:

Week 1

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

- set up system 1 for partner A: RUBRIC tech team
- business and communications manager and metadata specialist consult with partner

Week 2

- set up system 2 for partner B: RUBRIC tech team
- configure system 1 with partner staff: RUBRIC metadata specialist

Week3

- review and collect requirements for the next round of trials

Week *n*

- Repeat the staggered cycles above. Business and Communications Manager to work with Project Manager on reviewing the demonstrator.

Evaluating Support Services

Evaluate support services for a given software solution during the software evaluation phase, not post-implementation. This is a good opportunity to investigate:

- the type of support on offer – high level, technical, metadata
- support responsiveness – are questions answered overnight, within a few hours, a few days, weeks
- whether the type of support is satisfactory for your institution

The difference between vendor support and open source support can vary greatly. It is important to check that all involved in looking at the repository are satisfied with whatever is selected and the support offered.

Develop a reporting structure to identify all problems associated with installation, functionality, customisation, metadata, workflows, etc. When problems are encountered, document the problem, responses received and the time and resources taken to resolve them. Include this information in your final evaluation report.

A technical skill list may be useful to determine whether you have the existing skills in house to manage the chosen solution long term. This list was compiled for the RUBRIC partners who selected the VITAL solution.

Planned Obsolescence in a Repository Solution

A [Fedora repository](#)¹⁹ solution was initially thought to be the best long term option for IRs by RUBRIC Project members. This has become a matter of debate and our conclusion is that while this is a reasonable position to take, it is by no means clear that a Fedora-based repository is the best short-term option.

At the time of RUBRIC's initial investigation, Fedora based solutions were immature, whilst the alternative DSpace or e-Prints packages were both well established. RUBRIC data migration work revealed that an organisation could use either solution and be assured that a migration to a Fedora based solution would be possible in the future. Both packages have some limitations, but their stability and proven ability may outweigh these in the short term.

How Many Repositories?

A single repository solution does not need to be used for all repository requirements. While there are benefits to minimising the use of a number of different packages, there can be reasons for using different software for different types of repositories.

Separate repositories may be required for:

- an Open Access research repository
- a thesis repository, which may require embargo features and authorization, for example if theses contain third party copyright material, confidential material or information that could compromise patents
- image collections
- works in progress
- a preservation repository, containing records from all of the above but without a public portal (a closed repository).

Taking a Strategic Approach

Choosing a repository system requires synthesising the considerations discussed here into a decision. Institutional context, the software itself and support options all play a part, as does the requirements and preferences of key stakeholders, who may prefer one solution over another.

It is impossible to provide prescriptive advice on the matter of choosing a system. It may be appropriate to choose a system based solely on the fact that a staff member is familiar with it, or it is offered by a preferred supplier. Alternatively, the institution may require a detailed evaluation process.

References and Further Reading

Refer to the Further Reading section at the end of the Toolkit for bibliographic details of works referenced in this section.

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

“RUBRIC Toolkit: System Options” produced May 2007



20

21

Copyright²² 2007 RUBRIC²³

- 1 <http://www.soros.org/openaccess/software/>
- 2 http://www.rubric.edu.au/repositories/choosing_a_repository.htm
- 3 <http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/ir-software.pdf>
- 4 <https://eduforge.org/docman/view.php/131/1062/Repository%20Evaluation%20Document.pdf>
- 5 <http://www.vmware.com/products/vi/esx/>
- 6 http://www.rubric.edu.au/techreports/VMware_At_RUBRIC.htm
- 7 <http://www.rubric.edu.au/techreports/index.htm>
- 8 <http://www.apsr.edu.au/publications/projects.htm>
- 9 <http://espace.library.uq.edu.au/view.php?pid=UQ:12868>
- 10 <http://www.dspace.org/>
- 11 <http://www.apsr.edu.au/>
- 12 <http://espace.library.uq.edu.au/view.php?pid=UQ:11924>
- 13 <http://www.apsr.edu.au/>
- 14 <http://www.vtls.com/Products/vital.shtml>
- 15 <http://arrow.edu.au/>
- 16 http://rubric.edu.au/repositories/choosing_a_repository.htm
- 17 http://en.wikipedia.org/wiki/Agile_software_development
- 18 <http://eprints.usq.edu.au/>
- 19 [http://en.wikipedia.org/wiki/Fedora_\(software\)](http://en.wikipedia.org/wiki/Fedora_(software))
- 20 <http://creativecommons.org/licenses/by-sa/2.5/au/>
- 21 <http://creativecommons.org/licenses/by-sa/2.5/au/>
- 22 <http://creativecommons.org/licenses/by-sa/2.5/au/>
- 23 <http://www.rubric.edu.au/>