Requirements - The Next Generation TU Delft Repository

Alastair Dunning, Alenka Princic, Ardi Nonhebel, Arent Bosman, Armand Guicherit, Jules Schoonman, Just de Leeuwe, Karin Clavel, Marcel Broekarts, Nicoleta Nastase, Paul Suijker, team Repository, Jaap van de Geer, Arie Braat, Andjela Tomic, Nicole Will, Michiel de Jong.

Versie 1.0: 13 June 2022
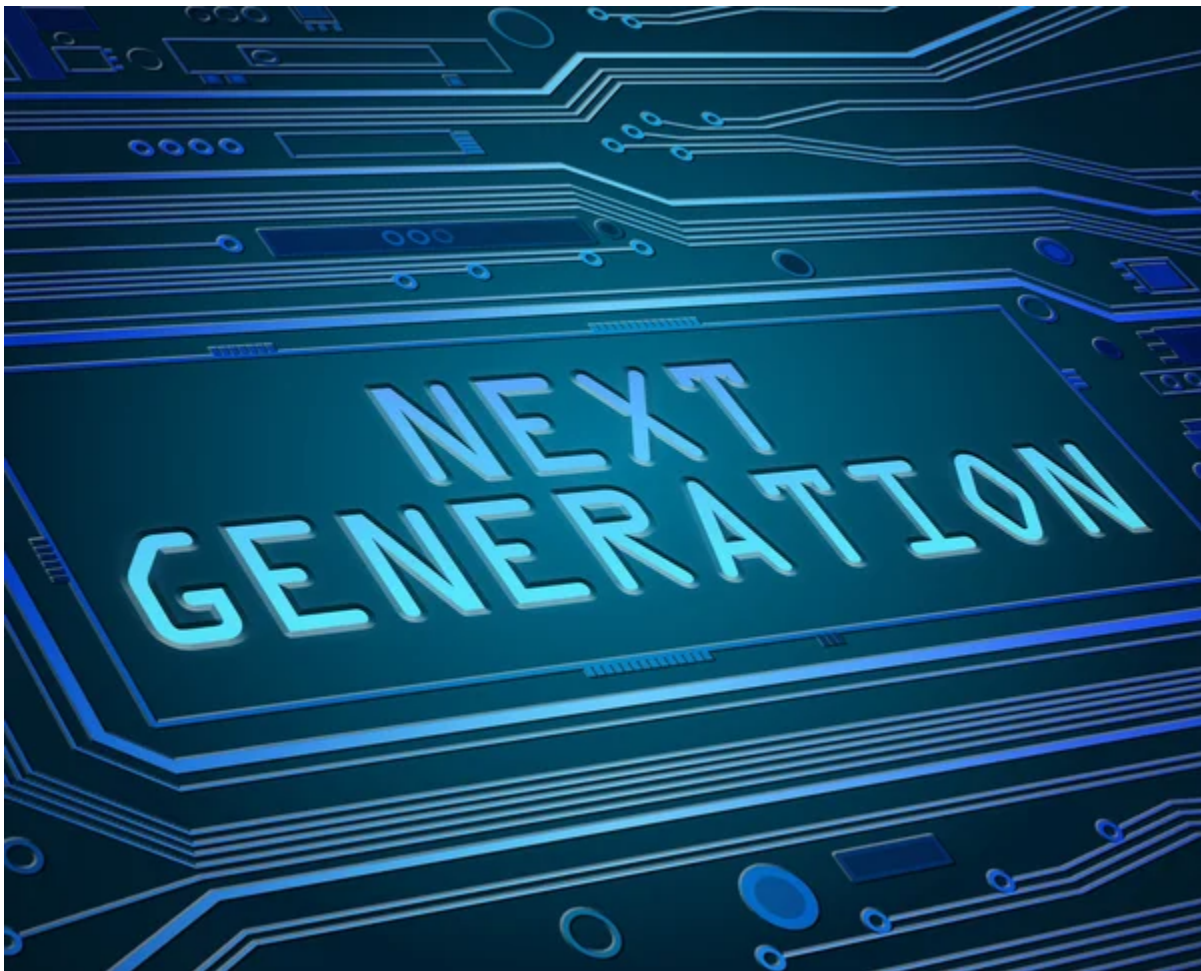Versie 2.0 definitief: 23 September 2022

Table of Contents

Introduction

The TU Delft Repository is an institutional repository with the aim to provide the public free access to the research-related publications and educational- related resources of the TU Delft.

The TU Delft Library manages the following repositories:
- research repository (IR)
- education repository (ER)
- cultural heritage repository (CH)
- colonial architecture repository (CA)
- TNO repository

Technical background:

The repositories now use an open source repository solution, Islandora 7. Islandora uses, among others, the following open source components: Drupal, Islandora, Fedora Commons, Apache Solr, Gsearch. The Islandora version 7 that we are currently using will be end of life at the end of 2023. This means that after that time no more security fixes will be issued for this version. The software runs on a Redhat 6 server. Redhat 6 is no longer supported by Redhat and no more security updates are issued for that either.
To prevent the repositories and infrastructure of TU Delft from being attacked by malicious people, it is wise for the TU Delft Library to look for a new software solution for the repositories.

Strategic background:

These repositories run on outdated software that focuses on metadata and not on the content. Content is not visualised, as modern solutions do with various image viewers. pdf etc. The repository has an efficient but not inspiring interface and look.
It is currently not possible for researchers to upload their content to the repository. It is desirable that they can do this themselves in the new repository.
There is the desire to set up a next generation TU Delft Repository according to the principles of COAR and Edustandaard.
The TU Delft Repository is an institutional repository with the aim to provide the public free access to the research-related *publications* of the TU Delft.

The TNO repository is currently the only remaining repository hosted for an external party. A discussion with TNO regarding the impact of our plans for the TNO repository has been started.

Definitions

Roles

- End-users: The public. End-users can query the system to find, view and download content.
- Depositor: Anyone with authorisation to deposit/upload and modify the metadata of their own content ; within the TU Delft context this can be a researcher or other member of staff employed by the TU Delft or a student at TU Delft. Once a depositor is no longer employed or enrolled, they lose this authorisation level.
- Moderator: Any member of (the Library's) staff with the authorisation to moderate can approve/reject the uploads of the depositor. The moderator can also deposit/upload and modify the metadata of all content.
- Admin: The person(s) responsible for the repository; functionality and services (administration, technical operations) and rights management.
- System admin: The person responsible for the software and hardware maintenance, server/API scripts, backup on the server, troubleshooting, bugs and issue resolving.

Used concepts

- Object: (In this context:) the metadata registration of a publication, including the uploaded content.
- Content: The uploaded files; can be documents, images, movies, audio-files.
- Repository: The software solution.
- System: The technical solution and the applications
- Metadata record: well-defined description that aids the findability and accessibility of the object.
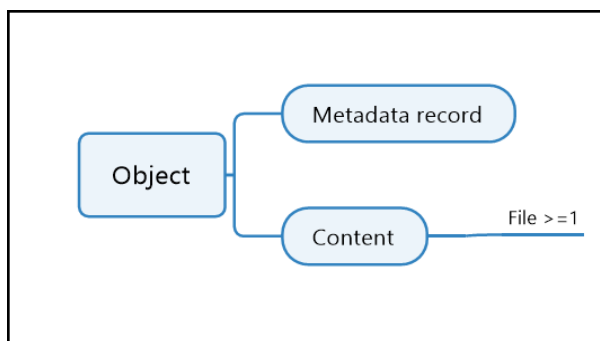
- Repository Group: Highest grouping level in the repository. Created and managed by the (system) admin (e.g. Institutional Repository, Education Repository, Cultural Heritage Repository, Colonial Architecture Repository, TNO Repository).
- Collection: An explicit grouping of objects in a repository group.
- Community: a user group working together on draft objects.
- Embargo: defined period of time that needs to be observed before an object is made available online in the repository.

High-Level MoSCoW Requirements

- <u>M:</u> must have
- <u>S:</u> should have
- <u>C:</u> could have
- <u>W:</u> will <u>NOT</u> have

Requirements

A	End Users	MoSCoW	Currently solved in:
A1	The end-user can view content within a collection.	M	repo
A2	The end-user can select content within a collection.	M	repo
A3	The end-user can browse through all the content, communities and collections.	M	repo
A4	The end-user can search the metadata and full text content of the repository.	M	repo
A5	The end-user can search through multiple collections of the repository.	S	repo
A6	The end-user can download - free from embargo - full-text content from the repository.	M	repo/pure
A7	The end user can access the repository through a graphic user interface (GUI) on the internet.	M	repo
A8	The end-user is not required to log in to view/access the content of the repository.	M	repo
A9	The end-user can preview the (non-embargoed) full-text content of the repository (without opening the file).	C	repo
A10	The end-user can leave a comment with an object.	W	…
A11	The end-user can share a DOI or other Persistent Identifier which resolves to an object.	C	portal
A12	The end user can use social media buttons to share an object on a social media channel such as Facebook and Twitter, etc.	C	repo

A13	The end-user can preview videos and images.	S	repo
A14	As an end-user, I need to easily and uniformly identify the licensing and re-use conditions of a scholarly resource, so that I know what I am allowed to do with it.	M	repo
A15	As an end-user, I want to receive recommendations about content that is of potential interest to me and related to my work, so I increase my knowledge in my field.	C	*new*
A16	As an end-user, I want to be able to search the repository using project/funder/grant ID's.	S	*new*
A17	The end-user can export the metadata records of an object or a set of objects in the following formats: - CSV - RIS - BibTeX	M	Pure (RIS & BibTeX), IR (CSV); Repo: CSV, Excel
A18	The end user can export the metadata records of an object or a set of objects in the following formats: - Excel - Citation style formatted RTF (APA[1], IEEE[2])	S	Pure, IR (Excel)
A19	The end-user can see bibliometric data such as citation count on each object.	C	*new*
A20	The end-user can see the number of views of each object.	M	*new*
A21	The end-user can see the number of downloads of each file.	M	*new*
A22	The end-user can request access to an embargoed or closed file. (Request for Copy functionality).	S	*new*
A23	The end-user can use advanced search to build a more complex search query on selected fields using Boolean operators, wildcards and special characters.	M	repo
A24	The end-user can download single files as well as the entire zipped data.	S	*new*

A25	The repository service is technically and functionally usable by people with disabilities, complying to relevant standards and legislation.	M	*new*
A26	The end-user can use and resolve a persistent identifier for each object.	M	repo

ORCiD / DOI, etc.

B	(UPLOADER) Depositor	MoSCoW	Currently solved in:
B1	A depositor can upload his TU Delft content in the repository.	M	pure*
B2	A depositor can add multiple files to an object.	M	pure
B3	A depositor can add additional (new) files to an existing object.	M	*new*
B4	A depositor can decide when to submit his content to the moderator to be made public.	M	pure
B5	A depositor can define an embargo period for his content (files) by providing an embargo end date.	M	pure
B6	A depositor can define an embargo period for his metadata records by providing an embargo end date.	S	*new*
B7	A depositor can define an embargo period for his metadata abstract field.	C	*new*
B8	A depositor can leave the end date of the embargo period open ('indefinite').	W	pure
B9	A depositor can delete his content, provided they have not made it public.	M	pure
B10	A depositor has the option to hide both metadata and file ('dark archive')	W	…
B11	A depositor can upload his content via drag and drop functionality.	S	pure

B12	A depositor can put an embargo on one or more individual files.	C	pure
B13	A depositor can provide metadata at file level.	S	pure
B14	A depositor can edit the metadata at file level.	S	pure
B15	A depositor can save an object as 'draft'. Depositors can return to these draft objects at a later time and make changes to the metadata and data files without creating a new object.	M	pure
B16	A depositor can replace a public file - the existing file is then stored as a previous version.	M	pure
B17	A depositor can create his own collection.	S	*new*
B18	A depositor can add his objects to one or more collections he/she is a member of.	M	*new*
B19	A depositor can request for his object to be included in a collection that is not his own.	S	*new*
B20	The depositor can reserve a DOI (in DataCite) for his object.	S	*new*
B21	The depositor can register a DOI for a object with the push of a button	M	*new*
B22	A depositor can attribute a licence to his content such as Creative Commons. Licences can be given to each individual file.	M	pure/repo
B23	A depositor can request restore access to embargoed files and metadata records.	M	pure/repo
B24	A depositor has no restrictions on the file formats that can be uploaded to the repository.	C	*new*
B25	A depositor can grant access to 'draft' objects to other depositors.	C	*new*
B26	A depositor has the option to create a shareable URL/link by which a draft can be shared with others. This URL cannot be traced back to a person or object title.	C	*new*
B27	A depositor has a profile page where he/she can view and edit their own user information, view their own datasets (accepted, draft, rejected) and related statistics,	M	*new*

	resume upload of draft objects, etc.		
B28	A depositor has the option to create a metadata-only record.	C	pure
B29	A depositor can display or link to other objects that have a relationship to the object.	M	Pure
B30	A depositor can log in using SURFconext TU Delft (Single-Sign-On using SAML v.2).	M	Pure
B31	A depositor can upload big files (1GB).	M	Pure/IR
B32	A depositor can view the download statistics of their data.	M	*new*
B33	A depositor can view the *view statistics* of the object.	M	*new*
B34	A depositor can upload the files including the containing folder structure.	S	*new*
B35	A depositor can upload a zip file that will be automatically unzipped. `or just display filetree???`	S	*new*
B36	A depositor can add a new version of an existing file (versioning)	M	IR
B37	A depositor can remove an object provided it is still in draft status.	S	Pure
B38	A depositor can remove an object in the 'for validation' or 'public' workflow status.	W	Pure
B39	A depositor can discontinue (archive) his collection: the content of this collection will remain part of the repository group.	S	*new*
B40	A depositor can add their object to multiple collections.	M	*new*
B41	A depositor can create a community.	C	*new*
B42	A depositor can add their ORCID id to their profile.	M	*new*
B43	A depositor can look up affiliated institutions in the ROR registry for the persons related to an object. They can do this in the repository.	S	*new*

B44	A depositor can provide metadata at object level.	S	pure
B45	A depositor can edit the metadata at object level.	S	pure

C	Moderator	MoSCoW	Currently solved in:
C1	Once an object is submitted for validation, all metadata changes have to be validated by a moderator. The moderator validates objects submitted by depositors and can accept or reject these.	M	pure
C2	A moderator has access to all content and metadata records irrespective the object's workflow status (draft/for validation/public).	M	pure
C3	A moderator is notified of uploaded content and metadata changes.	M	pure
C4	A moderator receives notification of removal requests by depositors for public objects.	C	*new*
C5	A moderator can do all operations that a depositor can do.	M	pure/repo
C6	Moderators can act on behalf of collection owners.	M	*new*
C7	A moderator has the option to hide files from public access. The file will remain accessible to the moderators and admin.	M	pure (not used)
C8	A moderator gets a notification when an end user requests a copy of an embargoed or closed file.	M	repo/pure (not

D	Admin	MoSCoW	Currently solved in:
			activated)
D1	The admin can limit access to one or more metadata fields to moderators and admins only.	**M**	pure
D2	The admin has all the rights that moderators have and more.	**M**	pure
D3	The admin or system admin can create additional metadata fields.	**M**	pure/repo
D4	The admin can define the list of licence options.	**M**	pure/repo
D5	The admin can change the metadata field-labels.	**M**	pure/repo
D6	The admin can make (one or more) metadata fields mandatory.	**M**	*new*
D7	The admin can add contextual help for each metadata form element.	S	Pure (but not for all metadata fields)
D8	The admin can define vocabularies which are used as controlled field values from which the depositors can choose.	**M**	pure
D9	The admin can create different Repository Groups. For each Repository group the admin	**M**	pure/repo

		MoSCoW	Currently solved in:
	can define separate metadata forms.		
D10	The admin can remove content and metadata from public view and make them inaccessible to contributors.	**M**	pure/repo
D11	The admin can discontinue (archive) an orphaned collection (no longer maintained by a depositor).	C	*new*
D12	As an admin, I wish to be able to implement a broader variety of measures to assess impact including repository metrics and incorporate them in my reports that assess the impact of the research I support. (COAR)	C	pure
D13	The admin can generate flexible statistical information regarding the content and usage of the repository at given points in time (including the number of views and downloads)	S	pure
D14	An admin can create his own collection.	S	*new*
D15	An admin can add his object(s) to one or more collections he/she is a member of.	**M**	*new*

E	System Admin	MoSCoW	Currently solved in:
E1	The admin or system admin can create additional metadata fields. (= D3)	**M**	pure/repo
E2	The system admin is able to make changes to the data in any of the metadata fields in bulk (CRUD functionality) .	**M**	*new*
E3	The system admin can delete objects.	**M**	pure/repo
E4	The system admin can define their own custom <mark>metadata schemes for OAI-PMH,</mark> such as	**M**	pure/repo

	DIDL-MODS, DC, OpenAire (CERIF).		
E5	The system admin can customise the look and feel, e.g. logo, banner, colour, outlining, fonts, etc.	M	pure/repo
E6	The system admin can define which metadata fields are shown in the graphic user interface. (NOTE: not all metadata needs to be shown to the end-user)	M	pure/repo
E7	The system admin can view all kinds of logs (system up/down, create, change, delete objects, metadata records, files, error/access logs etc.) which can help to provide information about problems.	M	pure/repo
E8	The system admin can define a maximum embargo period	M	pure

G	Technical Requirements	MoSCoW	Currently solved in:
G1	The repository checks all uploads for duplicates based on specified metadata fields.	C	pure
G2	The repository checks for possible duplicate versions of content	C	pure
G3	Raw statistics of the repository can be accessed by an API.	M	*new*
G4	The repository software supports file versioning on storage level.	M	pure/repo
G5	File versions can be 'metadated' so end-users can identify the version.	M	pure/repo
G6	The system logs all changes to a file: what changed, when it changed and who made the change; AKA an 'audit trail'.	S	pure/repo
G7	The repository provides each individual object with its own Persistent Identifier (PID).	M	repo

G8	The CC licence links to the right place on the CC website.	S	repo
G9	The system supports the creation of relations between objects	M	pure/repo
G10	The inclusion of metadata on a file does not result in the creation of a new version of the file.	S	pure/repo
G11	The system supports faceted filtering on metadata fields.	M	repo
G12	The system supports the requirements of the Google Scholar indexing service. (https://scholar.google.com/intl/en/scholar/inclusion.html)	M	repo
G13	The system supports the requirements of the OpenAIRE indexing service.(https://guidelines.openaire.eu/en/latest/)	M	repo
G14	The system logs downloads of files (for statistics).	M	repo
G15	The system logs views of objects (for statistics).	M	repo
G16	The system provides reliable usage statistics (downloads, views).	M	repo
G17	The system is able to show external impact metrics on the content (e.g. Altmetrics).	S	pure
G18	The system can import files and metadata.	M	repo/pure
G19	Files and metadata imported from another resource can be protected from editing.	M	repo/pure
G20	The system can import metadata and files from e.g. Scopus, Web of Science, IEEE Xplore.	C	pure
G21	The system can export all metadata and files to other systems such as a CRIS.	M	pure/repo
G22	The system can identify duplicate objects.	M	pure/repo
G23	An updated file from a source is added as a new version of the file and creates a new version of the object.	M	?

G24	The system fully supports the OAI-PMH protocol.	**M**	repo
G25	The system can export metadata using an OAI-PMH service in multiple customizable formats. (e.g. OAI DC, MODS, CERIF, for Narcis, OpenAIRE, etc.).	**M**	repo
G26	The system can retain existing Persistent Identifiers of all migrated objects.	W	…
G27	The system can create and assign a Persistent Identifier to each new object.	**M**	repo
G28	The system can import (UU)ID's from data source (Pure) in the hidden metadata in the repository.	**M**	repo
G29	Long term digital preservation is supported by the system (link to standard, description of LTP) [Relate to OAIS reference model]	W	…
G30	Publication metadata and administrative metadata are stored together in non-proprietary (human and machine) readable files.	**M**	repo
G31	The system supports export to human and machine readable files (good example or possible candidate: Oxford Common Filesystem Layout).	**M**	*new*
G32	The system provides a well-documented (CRUD) REST API for an admin user that enables the same functionality as the user interface. This includes uploading and downloading data and manipulating metadata.	**M**	pure
G33	The source for all data concerning TU Delft staff and organisational units is derived from the HR system and the student administration system in line with current practices and operations at TU Delft.	W	…
G34	The system allows versioning of an object and its DOI.	S	*new*
G35	The system is RDF/SPARQL based.	S	repo
G36	The software solution has to connect to file or object storage at the TU Delft. Preferably object storage.	**M**	repo/pure

G37	The repository must support a customised URL domain for TU Delft domain (tudelft.nl).	M	repo/port al
G38	The repository must register DOIs for published objects with the TU Delft DOI prefix.	M	repo
G39	Support for MD5 or SHA256 checksums, to guarantee data integrity if data is uploaded/downloaded.	M	*new*
G40	The repository is able to manage log data (all actions through which data files are uploaded, stored, accessed or modified, and username/name of the individual that modified the data).	M	repo/pure
G41	The repository performs virus checks on uploaded individual files and files in zipped files.	S	*new*
G42	The concepts 'Privacy by Design, 'Privacy by Default' and 'Security by Design' are embedded in the repository system.	S	*new*
G43	Adaptive or responsive design is used to ensure the repository service renders well on desktops, tablets and mobile devices	M	pure
G44	Data in transit are encrypted according to suitable encryption standards during the term of the Agreement.	M	?
G45	The data must always be secured via a backup/restore procedure and should be portable as much as possible, so that access can be guaranteed.	M	repo/pure
G46	Existing processing of personal data complies with the guidelines for lawful and careful processing of personal data as determined by the TU Delft.	M	repo/pure
G47	The repository supports the International Image Interoperability Framework (IIIF) API's: - IIIF Image API 3.0 - IIIF Presentation API 3.0 - IIIF Content Search API 1.0	S	*new*

G48	Mime types are retrievable (e.g. via an API) per object	S	?
G49	The system provides a workflow containing the status 'draft', 'for validation' and 'public'.	**M**	pure
G50	The system must pass a DPIA executed by ICT.	**M**	n.a.
G51	The system meets the requirements following from the 'Edustandaard afspraken' regarding metadata standards, interoperability, semantics and persistent identifiers. ● Digital Item Declaration Language (DIDL) ● Metadata Object Description Scheme (MODS) ● Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH) ● Semantics ● Uniform Resource Names (URN)	S	repo
G52	The system indexes all metadata fields and content properties to make these searchable for the end users.	**M**	repo?
G53	Include open graph metadata & preview image for social media: https://ogp.me	C	*new*
G54	Add schema.org, see: https://schema.org. Recommended by NDE (Network Digital Heritage). Can also be added as microdata to improve indexing by aggregators: https://schema.org/docs/gs.html	C	*new*
G55	Used Persistent Identifiers (DOI, ORCID) are linked to the appropriate resolver.	**M**	repo?
G56	The system is linked to the ORCID system, allowing depositors to link their ORCID IDs to the repository.	**M**	pure
G57	The system is linked to the ROR registry allowing depositors to look up institutes. The ROR PID is stored in the system. [See also: **B42**]	S	*new*

I	Third Party Agreement/contract requirements	MoSCoW	Currently solved in:
I1	The ICT party will conduct a DPIA – Data Protection Impact Assessment. A DPIA is carried out once / at the start of the project together with the person responsible in the project team (project leader, client, data protection officer, privacy representative, security representative and tenderer). Based on the results from the DPIA (the privacy information to be processed), mitigating measures may be proposed, which must be included in the implementation of the repository and monitored.	M	
I2	The third party provides technical and logistical support for the migration of data and metadata from the current instance of the repository. The same principles apply to the migration as for the test in the Proof of Concept.	M	
I3	For scheduled maintenance, such as installing releases and security updates, service windows are used. Scheduled maintenance is performed in such a way that users experience the least possible inconvenience.	M	
I4	Adequate back-up and restore facilities are ensured to guarantee availability of the repository service and thus the data stored therein.	M	
I5	The maximum time needed to recover the service in case of an incident (Recovery Time Objective) is 8 hours. The maximum period over which data may be lost (Recovery Point Objective) is 4 hours. If tetrefhe repository service is partly or completely unavailable, end-users will be notified of this via a notification on the website.	M	
I6	The third party reports digitally on the maintenance of the agreed service levels every quarter during the term of the agreement. The report must include the agreements that will be laid down in the SLA, including but not limited to: o Availability of the repository o performance issues	M	

	o incidents and solutions offered		
I7	Technical and functional documentation about the repository system is provided in the form of product information, manuals, etc. The documentation is in English. The documentation is accessible via a website.	M	
I8	A release policy is implemented, including a minor and major release deployment and release notes when an update is released.	M	
I9	Training and transfer of knowledge must be provided for administrators and moderators of the repository, during implementation and when the new repository service is up and running, and when there are significant changes to the functionality.	M	
I10	A roadmap for the next two years should be presented with an overview of planned development of functionality.	M	
I11	The third party provides a testing environment and sandbox.	M	
I12	The third party is able to detect and manage security breaches in the repository system.	M	
I13	Information security is applied to keep the information confidential, available, and assuring its integrity.	M	
I14	It's crucial to the contracting authority that when data is transferred from the current system to the new repository service, no data and metadata will be lost or are migrated incorrectly. To ensure a successful migration and a smooth transition to the new repository service, the contracting authority wants the third party to provide a description of all stages, steps and activities associated with a successful full implementation and migration process.	M	

COAR ngr requirements (not for testing)

https://ngr.coar-repositories.org/

J1	As a research administrator, I wish to use a broader variety of measures to assess impact including repository metrics and incorporate them in my reports that assess the impact of the research I support.
J2	As a repository manager, I want to be able to access the metadata in my repository in real time through an API in order to build views or services on any platform using the data.
J3	As a user, I want my repository to recognize me and others so that I can be connected with other users who I know, leave comments and be informed of content that is of interest to me.
J4	As a user, I want to discover repository materials of interest via aggregators or other search services such as Google, Google Scholar, CORE, and so on.
J5	As a researcher or repository manager, I want to deposit my research object with as little effort as possible into the repository platform.
J6	As a scholar or institution, I want my research outputs to be available over the long term and remain as a permanent part of the scholarly record.
J7	As a web reference manager, annotation tool, or crawler, when I encounter a landing page or any other web resource that is part of a scholarly object, I need to easily identify the associated persistent HTTP URI for the resource, so that I can retrieve it.
J8	As a machine or human user, I need to easily and uniformly identify the licensing and re-use conditions of a scholarly resource, so that I know what I am allowed to do with it.
J9	As a human or machine user, I want to easily and uniformly identify the metadata in a repository record, so that I can ascertain the relevance of a resource.

J10	As a human or machine user, I want to be able to mine the collective full text content of repositories to discover new relationships and make new discoveries.
J11	As a repository manager, I want my repository to be automatically notified about new or modified relevant objects and metadata, so that I can have a more complete and accurate collection. I also want other, remote systems to be notified of changes made to my collection to ensure that records are standardised across various locations.
J12	As a user, I want to be able to comment on or review the work of my colleagues and have those reviews (and reviewers) publicly available to other readers, so that the quality of a resource can be assessed by others.
J13	As a user, I want to receive recommendations about content that is of potential interest to me and related to my work, so I increase my knowledge in my field.
J14	As a researcher, I want to know how many times a repository object has been downloaded and used by others according to standard metrics.
J15	As a repository user, I want to have access to a global, cross-repository social feed so that I am informed about activities in which I have registered an active interest.
J16	As a research manager (funder of an institution), I want to be able to track the research outputs related to a specific funded project.

References

1. https://apastyle.apa.org/
2. http://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf
3. https://ngr.coar-repositories.org/
4. https://qracorp.com/functional-vs-non-functional-requirements/
5. https://metadata2020.org/resources/metadata-principles/

Recommended reading

6. Zenodo principles: https://about.zenodo.org/principles/
7. CoreTrustSeal trustworthy data repository requirements: https://zenodo.org/record/3638211#.YY4tKb3MKfU
8. OpenDOAR: https://v2.sherpa.ac.uk/opendoar/policytool/
9. COAR Controlled Vocabularies: https://vocabularies.coar-repositories.org/
10. EU Grants AGA: https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/common/guidance/aga_en.pdf
11. NWO Open Access policy: https://www.nwo.nl/en/open-access-publishing
12. Architectuurprincipes https://intranet.tudelft.nl/-/architectuurpincipes-ict-nl?p_l_back_url=%2Fsearch%3Fq%3Darchitectuurprincipes
13. ISO 25010 https://nl.wikipedia.org/wiki/ISO_25010
14. Nonfunctional requirements https://www.scaledagileframework.com/nonfunctional-requirements/ en
    https://en.wikipedia.org/wiki/Non-functional_requirement