

Findability

F1. Identity Uniqueness

The software has a proper, unique and persistent identifier. The uniqueness of an identifier is a necessary condition to unambiguously refer to that resource, and that resource alone.

F1.1. Uniqueness of Name

The software has a unique name to identify it

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

F1.2. Identifiability of Version

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of metadata

The software is described with rich metadata, including scientific applicability. Metadata makes finding through search engines and deciding if a tool is of interest possible.

F2.1. Structured Metadata

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least a source of structured metadata is considered valid.

F2.2. Standardised Metadata

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

Findability

F1. Identity Uniqueness

The software has a proper, unique and persistent identifier. The uniqueness of an identifier is a necessary condition to unambiguously refer to that resource, and that resource alone.

F1.1. Uniqueness of Name

The software has a unique name to identify it.

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

F1.2. Identifiability of Version

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of metadata

The software is described with rich metadata, including scientific applicability. Metadata makes finding through search engines and deciding if a tool is of interest possible.

F2.1. Structured Metadata

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least a source of structured metadata is considered valid.

F2.2. Standardised Metadata

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

0.8

All

F1. Identity Uniqueness

0.4

The software has a proper, unique and persistent identifier. The uniqueness of an identifier is a necessary condition to unambiguously refer to that resource, and that resource alone.

F1.1. Uniqueness of Name

0.8

All

The software has a unique name to identify it.

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

F1.2. Identifiability of Version

0.2

All

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of metadata

0.2

The software is described with rich metadata, including scientific applicability. Metadata makes finding through search engines and deciding if a tool is of interest possible.

F2.1. Structured Metadata

0.6

All

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least a source of structured metadata is considered valid.

F2.2. Standardised Metadata

0.4

All

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

F3. Discoverability

0.4

How software can be found. There are multitude of mechanisms for scientists looking to find specific software

F3.1. Discoverability in software registries

*

All

The software is included in the main software registries.

Why: Software registries are the main resource scientists use when searching for software.

How: At least one software registry among the instance sources is considered valid.

F3.2. Discoverability in software repositories

*

All

The software can be found in any of the major software repositories e.g. GitHub, GitLab, SourceForge.

Why: Software repositories can be an additional resource used by scientists when looking for software.

How: An associated software repository is considered valid.

F3.3. Discoverability in literature

*

All

The software can be found in specialised literature services e.g. EuropePMC, PubMed, Journals Site, biorxiv.

Why: Specialized literature is a good reference to find software, especially to discover new software.

How: At least one associated publication is considered valid.

All

Web

Non-Web

Type of tools
the indicator
applies to

x
x

Weight
(x in [0, 1])

All

F1. Identity Uniqueness

The software has a proper, unique and persistent identifier. The uniqueness of an identifier is a necessary condition to unambiguously refer to that resource, and that resource alone.

F1.1. Uniqueness of Name

The software has a unique name to identify it.

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

All

F1.2. Identifiability of Version

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of metadata

The software is described with rich metadata, including scientific applicability. Metadata makes finding through search engines and deciding if a tool is of interest possible.

F2.1. Structured Metadata

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least a source of structured metadata is considered valid.

All

F2.2. Standardised Metadata

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

All

F3. Discoverability

How software can be found. There are multitude of mechanisms for scientists looking to find specific software

F3.1. Discoverability in software registries

The software is included in the main software registries.

Why: Software registries are the main resource scientists use when searching for software.

How: At least one software registry among the instance sources is considered valid.

All

F3.2. Discoverability in software repositories

The software can be found in any of the major software repositories e.g. GitHub, GitLab, SourceForge.

Why: Software repositories can be an additional resource used by scientists when looking for software.

How: An associated software repository is considered valid.

All

F3.3. Discoverability in literature

The software can be found in specialised literature services e.g. EuropePMC, PubMed, Journals Site, biorxiv.

Why: Specialized literature is a good reference to find software, especially to discover new software.

How: At least one associated publication is considered valid.

All

Findability

F1. Identity Uniqueness

The software has a proper, unique and persistent identifier. The uniqueness of an identifier is a necessary condition to unambiguously refer to that resource, and that resource alone.

F1.1. Uniqueness of Name

All

The software has a unique name to identify it.

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

F1.2. Identifiability of Version

All

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of Metadata

The software is described with rich metadata, including scientific applicability. Metadata makes finding through search engines and deciding if a tool is of interest possible.

F2.1. Structured Metadata

All

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least a source of structured metadata is considered valid.

F2.2. Standardised Metadata

All

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

F3. Discoverability

How software can be found. There are multitude of mechanisms for scientists looking to find specific software

F3.1. Discoverability in software registries

All

The software is included in the main software registries.

Why: Software registries are the main resource scientists use when searching for software.

How: At least one software registry among the instance sources is considered valid.

F3.2. Discoverability in software repositories

All

The software can be found in any of the major software repositories e.g. GitHub, GitLab, SourceForge.

Why: Software repositories can be an additional resource used by scientists when looking for software.

How: An associated software repository is considered valid.

F3.3. Discoverability in literature

All

The software can be found in specialised literature services e.g. EuropePMC, PubMed, Journals Site, biorxiv.

Why: Specialized literature is a good reference to find software, especially to discover new software.

How: At least one associated publication is considered valid.

All

F1. Identity Uniqueness

F1.1. Uniqueness of Name

The software has a unique name to identify it.

Why: The name is commonly used as the main identifier of a software. Each tool should have a unique name to avoid ambiguities. Different versions of the same software should share a name, but if substantial modifications in the algorithm are done, the identifier should change for the new piece of software.

How: A name is valid.

All

F1.2. Identifiability of Version

There is a scheme to uniquely and properly identify the software version.

Why: A version scheme is necessary to refer to a specific release of a software and keep track of the incrementally different versions of the software.

How: A version of the form X.X is considered valid.

F2. Existence of metadata

F2.1. Structured Metadata

Metadata is adjusted to specific metadata formats

Why: Specific formats are more machine readable, which increases its findability by search engines.

How: At least one source of structured metadata is considered valid.

All

F2.2. Standardised Metadata

All

Metadata is described using accepted ontologies.

Why: The same piece of information about a software can be stated in many equivalent forms. Each tool being described with different terminology, with non specified meanings, makes metadata very hard to interpret. Automatic processing is also harder. When searching for a software with certain features, the lack of a consensuated terminology makes the process of searching slow and difficult.

How: EDAM, bioschema.

F3. Discoverability

F3.1. Discoverability in software registries

All

The software is included in the main software registries.

Why: Software registries are the main resource scientists use when searching for software.

How: At least one software registry among the instance sources is considered valid.

All

F3.2. Discoverability in software repositories

The software can be found in any of the major software repositories e.g. GitHub, GitLab, SourceForge.

Why: Software repositories can be an additional resource used by scientists when looking for software.

How: An associated software repository is considered valid.

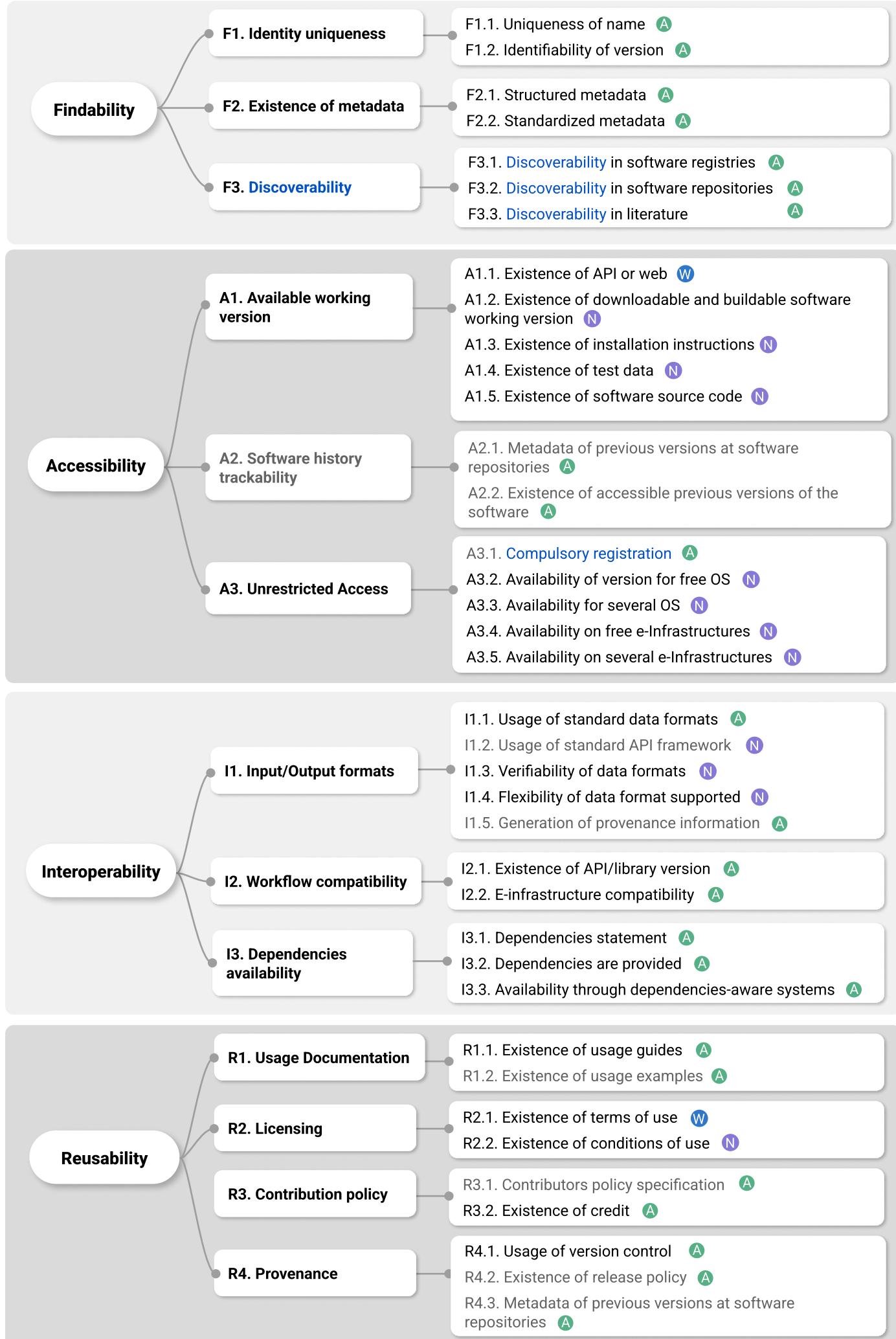
All

F3.3. Discoverability in literature

The software can be found in specialised literature services e.g. EuropePMC, PubMed, Journals Site, biorxiv.

Why: Specialized literature is a good reference to find software, especially to discover new software.

How: At least one associated publication is considered valid.



A The indicator applies to all types of software

N The indicator applies to non web-based software

W The indicator applies to web-based software

