

Aspects of usability in RSE

Jan Bernoth (Universität Potsdam)
Leen Lambers (BTU Cottbus-Senftenberg)
Workshop@deRSE am 26.02.2025

Motivation

- RSEs usually familiar with functional requirements, but **quality requirements** often less well understood [1]
- **Usability** is one of the quality requirements that may be critical for the research software to support its *underlying research goal*

[1] Wiese, I.; Polato, I.; Pinto, G.: Naming the Pain in Developing Scientific Software. IEEE Software 4/37, pp. 75–82, 2020.

The screenshot shows the 'Basic information' section of the CFF INIT tool. On the left, a sidebar lists various sections: Basic information (required), Authors (required), Identifiers (optional), Related Resources (optional), Abstract (optional), Keywords (optional), License (optional), Version Specific (optional), and Extra Cff Fields. The main area is titled 'Basic information' and contains instructions: 'Indicate whether your work is a software or a dataset, give it a title, and optionally change the default message.' Below this, there are radio buttons for 'Type of the work' with 'Software' selected and 'Dataset' as an option. A red-bordered box highlights the 'Title of the software (required)' field, which has a red error icon and the message 'Title cannot be empty.' Below this is a text area for a 'Personalized message. Leave blank to use default'. To the right, a 'CITATION.cff preview' box shows a sample CFF file content, including version, title, message, type, and authors. A red warning triangle icon is present below the preview with the text: 'Your CITATION.cff does not have the minimum fields. Make sure the title has been filled and that at least one author was added.' A 'Download' button is at the bottom right of the preview area. At the top right of the interface are links for 'REPORT AN ISSUE' and 'DOCUMENTATION'.

<https://citation-file-format.github.io/cff-initializer-javascript/#/start>

The screenshot shows the 'CodeMeta generator v3.0' web interface. It features a blue header with the title. Below the header, a note states: 'Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.' The form is organized into several columns. The first column, 'The software itself', includes fields for Name (with a red error icon and message 'the software title'), Description (with a red error icon and message 'My Software computes ephemerides and orbit propagation. It has been developed from early '80.'), Creation date, First release date, and License(s) (with a link to 'SPDX license list'). The second column, 'Discoverability and citation', includes fields for Unique identifier (with a red error icon and message 'such as ISBNs, GTIN codes, UUIDs etc. http://schema.org/identifier'), Application category, Keywords, Funding, and Funder. The third column, 'Development community / tools', includes fields for Code repository, Continuous integration, Issue tracker, and Related links. At the bottom, there are sections for 'Run-time environment', 'Current version of the software' (with a 'Version number' field), and 'Editorial review' (with a 'Reference Publication' field).

<https://codemeta.github.io/codemeta-generator/>

Aim of the workshop

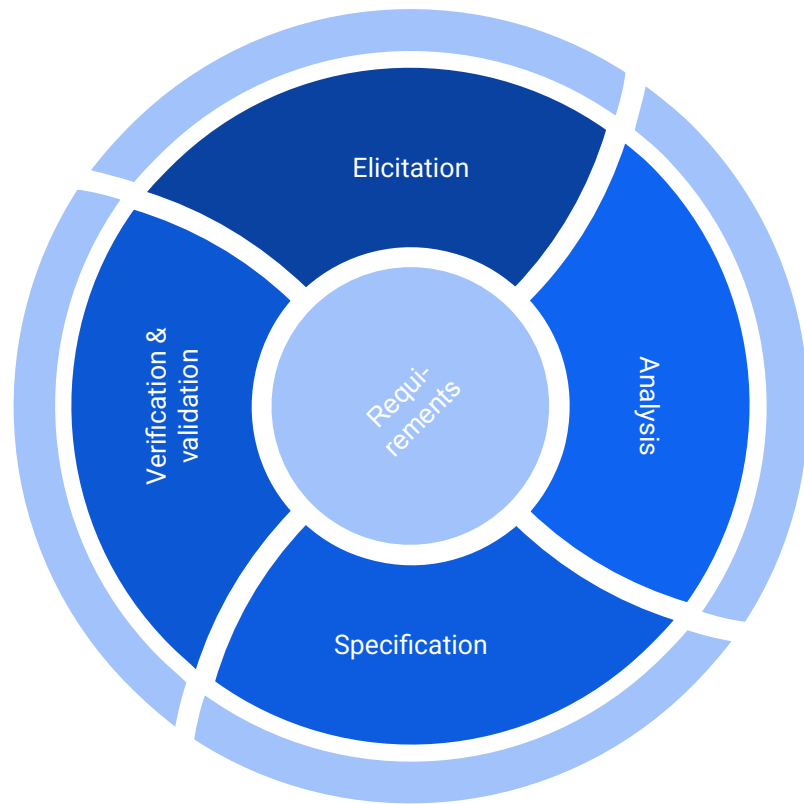
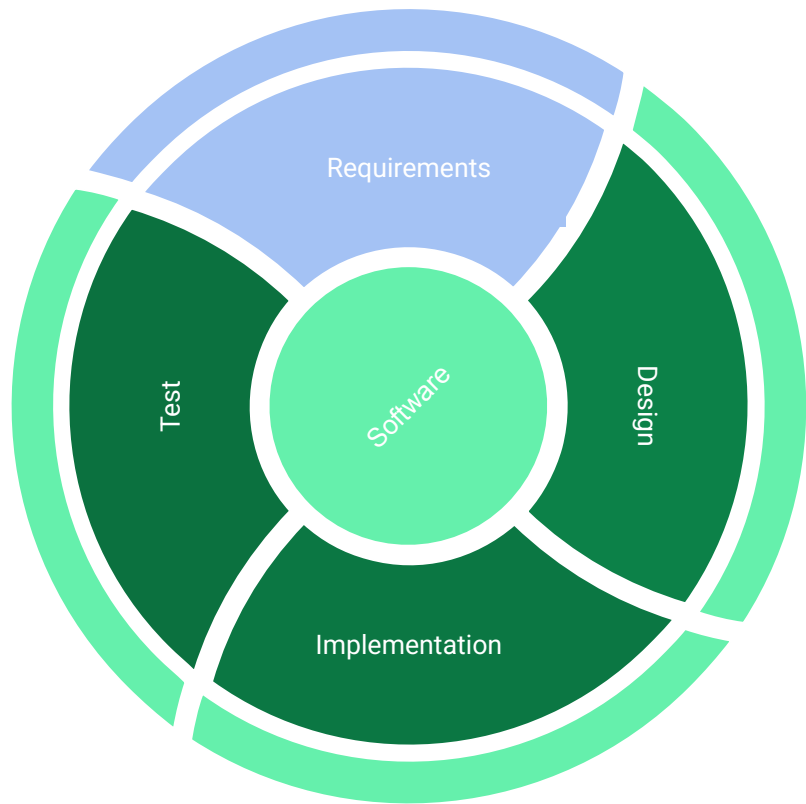
- Study and foster exchange on aspects of usability in RSE
 - **Interaction capability:** capability of a product to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task

https://iso25000.com/images/figures/iso_25010_en.png

SOFTWARE PRODUCT QUALITY								
FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
FUNCTIONAL COMPLETENESS	TIME BEHAVIOUR	CO-EXISTENCE	APPROPRIATENESS	FAULTLESSNESS	CONFIDENTIALITY	MODULARITY	ADAPTABILITY	OPERATIONAL CONSTRAINT
FUNCTIONAL CORRECTNESS	RESOURCE UTILIZATION	INTEROPERABILITY	RECOGNIZABILITY	AVAILABILITY	INTEGRITY	REUSABILITY	SCALABILITY	RISK IDENTIFICATION
FUNCTIONAL APPROPRIATENESS	CAPACITY		LEARNABILITY	FAULT TOLERANCE	NON-REPUDIATION	ANALYSABILITY	INSTALLABILITY	FAIL SAFE
			OPERABILITY	RECOVERABILITY	ACCOUNTABILITY	MODIFIABILITY	REPLACEABILITY	HAZARD WARNING
			USER ERROR PROTECTION		AUTHENTICITY	TESTABILITY		SAFE INTEGRATION
			USER ENGAGEMENT		RESISTANCE			
			INCLUSIVITY					
			USER ASSISTANCE					
			SELF-DESCRIPTIVENESS					

Aspects of usability

- **Appropriateness recognizability:** capability of a product to be recognized by users as appropriate for their needs
- **Learnability:** capability of a product to have specified users learn to use specified product functions within a specified amount of time
- **Operability:** capability of a product to have functions and attributes that make it easy to operate and control
- **User error protection:** capability of a product to prevent operation errors
- **User engagement:** capability of a product to present functions and information in an inviting and motivating manner encouraging continued interaction
- **Inclusivity:** capability of a product to be utilised by people of various backgrounds
- **User assistance:** capability of a product to be used by people with the widest range of characteristics and capabilities to achieve specified goals in a specified context of use
- **Self-descriptiveness:** capability of a product to present appropriate information, where needed by the user, to make its capabilities and use immediately obvious to the user without excessive interactions with a product or other resources



Usability requirements for research software

Research goal

Which *research goal* is supported by the research software project?

Research software

How does the *research software project* support the research goal?
How does *usability* of the research software project influence the research goal?

Users/context/goals

Which *users* are relevant for the research software project?
In which *context* is the research software project used?
Which *goals* do users aim to achieve with the research software project?

Usability requirements

Which *aspects of usability* are important for the research software project?
How to specify these *usability requirements*?

Steering questions

1. Who are the users of the RSE project based on the research goal?
2. Which usability aspects play an important role in the RSE project for these users?
3. How do these usability aspects influence the research goal that the RSE project contributes to?

Quality scenario



If a RSE wants to create meta data for a research software project for the first time, then the RSE is able to do this within 10 minutes without looking up additional resources.

Kazman, Rick; Klein, Mark; & Clements, Paul. *ATAM: Method for Architecture Evaluation*. CMU/SEI-2000-TR-004. Software Engineering Institute. 2000. <https://insights.sei.cmu.edu/library/atam-method-for-architecture-evaluation/>

The screenshot shows the 'Basic information' section of the CFF INIT web interface. It includes a sidebar with navigation links for Basic Information, Authors, Identifiers, Related Resources, Abstract, Keywords, License, Version Specific, and Extra CFF Fields. The main content area has a 'Basic information' heading and a form to indicate the type of work (Software or Dataset). A red box highlights the 'Title of the software (required)' field with an error message: 'Title cannot be empty.' Below this is a 'Personalized message' field. A 'CITATION.cff preview' box shows the generated code, and a 'Download' button is at the bottom.

<https://citation-file-format.github.io/cff-initializer-javascript/#/start>

The screenshot shows the 'CodeMeta generator v3.0' web interface. It features a blue header and a form with various fields for generating a CodeMeta file. The form is organized into sections: 'The software itself' (Name, Description, Creation date, First release date, License(s)), 'Discoverability and citation' (Unique identifier, Application category, Keywords, Funding, Funder), 'Development community / tools' (Code repository, Continuous integration, Issue tracker, Related links), and 'Run-time environment' (Current version of the software, Editorial review). A note at the top states: 'Most fields are optional. Mandatory fields will be highlighted when generating Codemeta.'

<https://codemeta.github.io/codemeta-generator/>

Quality scenario



If a RSE wants to fix an error in the generated meta data for a research software project, then the RSE is able to do this within 10 seconds.

The screenshot shows the **CFF INIT** web interface. On the left, there is a sidebar with navigation links: **Basic information required**, **Authors required**, **Identifiers optional**, **Related Resources optional**, **Abstract optional**, **Keywords optional**, **License optional**, **Version Specific optional**, and **Extra CFF Fields optional**. The main content area is titled **Basic information** and contains the following fields:

- Type of the work**: Radio buttons for **Software** (selected) and **Dataset**.
- Title of the software (required)**: A text input field with a red border and an error icon. Below it, a message says "Title cannot be empty."
- Personalized message. Leave blank to use default**: A text input field.

On the right, there is a **CITATION.cff preview** box showing the generated CFF file content:

```
CITATION.cff preview
# This CITATION.cff file was generated with cffinit.
# Visit https://bit.ly/cffinit to generate yours today!

cff-version: 1.2.0
title: ''
message: >-
  If you use this software, please cite it using the
  metadata from this file.
type: software
authors: []
```

Below the preview, there is a warning icon and a message: "Your CITATION.cff does not have the minimum fields. Make sure the title has been filled and that at least one author was added." A **Download** button is located at the bottom right of the preview box.

<https://citation-file-format.github.io/cff-initializer-javascript/#/start>

The screenshot shows the **CodeMeta generator v3.0** web interface. The title bar is blue with the text "CodeMeta generator v3.0". Below the title bar, there is a message: "Most fields are optional. Mandatory fields will be highlighted when generating CodeMeta." The main content area is divided into several sections:

- The software itself**: Fields for **Name** (with a red border and error icon), **Description**, **Creation date** (YYYY-MM-DD), **First release date** (YYYY-MM-DD), and **License(s)** (with a link to [SPDX license list](#)).
- Discoverability and citation**: Fields for **Unique identifier** (with a red border and error icon), **Application category**, **Keywords**, **Funding**, and **Funder**.
- Development community / tools**: Fields for **Code repository**, **Continuous integration**, **Issue tracker**, and **Related links**.
- Run-time environment**: Fields for **Operating system** and **Version number**.
- Editorial review**: Fields for **Current version of the software** and **Reference Publication**.

<https://codemeta.github.io/codemeta-generator/>

Quality scenario



If a RSE wants to complete partially created meta data for a research software project, then the RSE is able to restart with completion within 5 seconds.

The screenshot shows the **CFF INIT** web interface. On the left, a sidebar lists various sections: Basic information (required), Authors (required), Identifiers (optional), Related Resources (optional), Abstract (optional), Keywords (optional), License (optional), Version Specific (optional), and Extra CFF Fields (optional). The main area is titled **Basic information** and contains a form for creating a CFF file. It includes a **Type of the work** section with radio buttons for **Software** (selected) and **Dataset**. Below this is a **Title of the software (required)** field, which is highlighted with a red border and an error icon. A **Personalized message** field is also present. A **Download** button is at the bottom. On the right, a **CITATION.cff preview** shows the generated CFF file content, including version, title, message, type, and authors. A warning icon and message indicate that the CFF file does not have the minimum fields and that the title and authors must be added.

<https://citation-file-format.github.io/cff-initializer-javascript/#/start>

The screenshot shows the **CodeMeta generator v3.0** web interface. It features a blue header with the title. Below the header, a message states: "Most fields are optional. Mandatory fields will be highlighted when generating CodeMeta." The form is divided into several sections: **The software itself** (Name, Description, Creation date, First release date, License(s)), **Discoverability and citation** (Unique identifier, Application category, Keywords, Funding, Funder), **Development community / tools** (Code repository, Continuous integration, Issue tracker, Related links), **Run-time environment** (Operating system, Language), **Current version of the software** (Version number), and **Editorial review** (Reference Publication). The **Name** field is highlighted in red, indicating it is a mandatory field. The **License(s)** field has a link to the **SPDX license list**.

<https://codemeta.github.io/codemeta-generator/>

Workshop organization

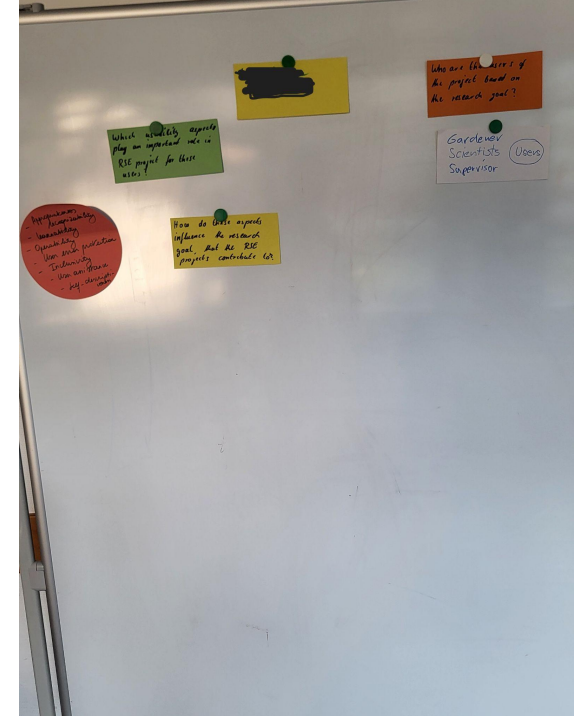
1. Intro & motivation
2. Collect RSE projects from participants
 - Research goal & RSE project descriptions
3. Select RSE projects to be further discussed
4. Split into groups per selected RSE project
5. Discussion of steering questions in groups
6. Report results of group discussion
7. Evaluation

Evaluation

- Which usability aspects are important in the selected RSE projects?
- How can research software engineers be supported with practical software engineering methods to take these usability aspects into account?

Keep in touch...

- If you are interested in collaborating further, ...

[illegible]

Pictures of the results (2)

Research SW project:

Research goal: Simplify the data processing pipeline for CRNS.

process COSMIC-ray neutron sensors (CRNS) for soil moisture monitoring:

1. Parse recorded data
2. Collect external data from NHD AM
3. Quality Assessment
4. Correct Neutrons for additional influence. (CRNS) (weather)
5. Smooth Data. — statistical.
6. Create soil moisture estimates
7. Visualisation
8. Saving.

Sticky notes:

- Which usability aspects play an important role in the RSE project for these users?
- Inclusivity to non-python or CRNS experts
- Ability to apply algorithms without needing to know details
- How do these aspects influence the research goal that the RSE project contributes to?
- Multiple interfaces to address inclusivity
- Obfuscating complex algorithms
- GUI / CLI / Jupyter

Other notes:

- Who are the users of the project based on the research goal?
- Research/Infrastructure interested in soil moisture and have a CRNS.

Research SW project:

Research goal:

- Corrections in display
- no data missing
- do not modify orig. data
- build analysis < 7d
- online work: responsive
- work locally
- maybe another Python code
- proper Python/Doc
- how simple/tutorial
- how local vs. cloud?
- Data collection?

Sticky notes:

- Who are the users of the project based on the research goal?
- How do these aspects influence the research goal that the RSE project contributes to?
- Which usability aspects play an important role in the RSE project for these users?
- Who are the users of the project based on the research goal?
- Programmers (RSE)
- Fieldwork scientists in cooperation