

Competencies and responsibilities of an RSE and how to acquire them (in Germany)

Heidi Seibold, Florian Goth, Jan Linxweiler, Jan Philipp Thiele, Jeremy Cohen, Renato Alves, Samantha Wittke, Jean-Noël Grad, Fredo Erxleben, Magnus Hagdorn, Harald von Waldow, Moritz Schwarzmeier, Matthias Braun, Simon Christ, ...

October 10, 2023
FG RSE Fachgruppentreffen

Timeline (I)

- November 2022: Call for Submissions to deRSE23
 - Idea for Workshop on 'Teaching and Learning Research Software Engineering' by Heidi Seibold
 - Joined by Florian Goth, Samantha Wittke, Jeremy Cohen and me
- deRSE23 (Paderborn) Workshop:
 - Brainstorming 1: 5 Questions
 - What are essential topics an RSE should know about?
 - Who needs RSE skills?
 - How can we reach the people who need RSE skills?
 - What are important sub-categories of RSE?
 - Who should teach RSE skills?
 - Brainstorming 2: 3 Groups by 'level':
 - Beginner/Learner: Top 5 things you want to learn now? How do you learn?
 - RSEs: Top 5 things you were missing? Certification programmes?
 - RSE trainers: Top 5 challenges in teaching? Certified trainers? Organizational structures?

Timeline (II)

- Working on 'the' paper
 - Rough draft
 - Reaching out to workshop participants for contributions
 - Weekly meetings
 - After many good contributions - too much content so paper is split
- underRSE23 (Jena)
 - Focus on Competencies paper as foundation for all others
 - Received feedback
 - Started work on missing parts (Appendix: Curriculum and example career path)

Content

- ① Foundational Competencies
- ② How to acquire skills and responsibilities?
- ③ Outlook

RSE?

Definition (most general description of an RSE)

An individual working in the research community who focuses on software development.

- Most are somewhere between two extremes
 - Traditional researcher who also develops software
 - Software Engineer similar to industry
- A wide variety of research domains (STEM, DH, etc.)
 - Vastly different software applications
 - Varied hardware (HPC clusters, embedded systems, Workstations)
 - Different programming languages and skills
 - BUT: a lot of shared problems (CI/CD, reproducibility, etc.)

Foundational Competencies

Goals

- Generic (common denominator)
- Important for all types of RSE
- Time invariant (tech stack agnostic)

Main Categories

- Software Engineering
- Research
- Communication/Team skills

Remark

*Different RSE specializations may need additional competencies:
e. g. extended hardware knowledge (HPC RSE), domain knowledge ($\{\text{DOMAIN}\}$ -RSE)*

Software Engineering Competencies (I)

Creating documented code building blocks

- Reusability & modularity
- Writing 'good' documentation
- Documentation tools

Building distributable libraries

- Handling dependencies
- Using domain/language specific distribution platforms (e. g. Spack, CRAN)
- Build systems (language dependent)

Understanding the software lifecycle

- Evaluating requirements
- Different 'needs' in different stages

Software Engineering Competencies (II)

Use software repositories

- Version control
- Interacting with the public
- Non-programming things (issues, pull requests etc.)

Software behaviour awareness and analysis

- Form a mental model of the software
- Predict software behaviour
- Needed for:
 - Debugging
 - Profiling
 - Test Driven Development
 - UI/UX

Research Competencies (I)

Curiosity

- New technologies
- New domain knowledge (algorithms, methods, etc.)

Understanding the research cycle

- Common goal of furthering knowledge
- Part of a greater cycle
- Different software needs at different stages
- Underlying principles (reproducibility, good research practice, etc.)

Software re-use

- Software Discovery
- Software Evaluation
 - Needed functionality?
 - Integrateable?
 - Sustainable?
 - Expandable?

Research Competencies (II)

Software publication

- License types ('proprietary', 'copyleft', etc.)
- License (in)compatibility
- Technical part (applying licenses, copyright statements, crediting)

Using domain repositories

- Most software has a home domain
- Knowledge of existing repos (data sets, software, etc.)
- Technical interaction (API)

Communication Competencies

Working in a team

- Handling different backgrounds (IT staff, domain researchers, software engineers)
- Code review, explaining implementation choices, etc.

Teaching

- Onboarding, Mentoring and consulting

Project Management

- Basic understanding
- Specific challenges of research software

Interaction with users and other stakeholders

- Ability to communicate with varied user/developer base (from domain experts to 'pure' software engineers)

Concrete Tasks and Responsibilities

Direct outcome of Paderborn workshop (What would you have liked to learn?)

- Testing + Frameworks and CI/CD
(mental model, documented code building blocks)
- Contributing to large projects
(software repositories, re-use, publication, documented code building, communication)
- When or why to keep repos private
(research cycle, working in a team, user interaction)
- Proper Development: IDEs, static analysis, design patterns, documentation, etc. (SE competencies, working in a team, project management)
- Finding a community both for research software projects and oneself (research lifecycle, user interaction, curiosity)

How much do different people need to know?

Different dimensions

- RSE Career level: Junior RSE, Senior RSE, Principal RSE
- Academic Career: Bachelor, Master, PhD, PostDoc, PI/Professor
- Project team structure:
 - RSE team: Individual working alone, group of devs
 - Local domain team: Individual, group of researchers
 - Organisational level

Content

- ① Foundational Competencies
- ② How to acquire skills and responsibilities?
- ③ Outlook

Current situation for competencies and skills

- Basic Programming (Languages mostly) during Bachelor and Master
- Maybe SE courses in computer science majors/minors
- Self taught
- Software Carpentries + Code Refinery
- Senior colleagues

Very hard to evaluate skills as an employer.
But we need a trained workforce with rising requirements.
→ we need dedicated training.

A master in RSE

Short example curriculum of a Master's degree based on a bachelor in a 'home domain'
(STEM + computer science, life sciences, humanities, social sciences)

Will be interdisciplinary and across multiple faculties.

- Basic courses: Intro to Programming, SW Craftmanship, SW Engineering, SW Design, Life Cycles, Interdisciplinary scholarly communication
- An interdisciplinary project
- Specialization electives: Either home domain or SE specific
- A Master's thesis in the home domain under dual-supervision
 - Domain supervisor: gives/discusses research question and grades the research skills
 - RSE supervisor: Discusses and grades software

Remark

Will be flashed out in an upcoming paper about institutionalized RSE education. Ideas wanted!

Current situation for RSE Careers

How to acquire RSE responsibilities

- Get one of the increasing number of RSE positions (still very few)
- By own initiative in research group

Problems:

- Competencies often acquired quite late, i. e. during PhD
- No clear path, more luck and chance
- Hard to advertise as viable option
- Missing acknowledgment
- PhD required for most grant applications

An example (fictional) career path

Kay, a student of 'researchology'

- Bachelor: Learns computational tools through extracurricular course
- Thesis: Fully open and reproducible
- Masters: Chooses generic RSE master finishes with a project in...
- Junior RSE: Position in RSE department at her university (Learns teaching)
- Senior RSE: Takes on RSE responsibilities in a new research group at a different university
- Senior RSE: Responsible for RSE parts of a large inter-organizational project (Verbundprojekt)
- Acquires 'Senior RSE certificate' and can now apply for RSE related grants

Content

- ① Foundational Competencies
- ② How to acquire skills and responsibilities?
- ③ Outlook

Next steps

Competencies paper

- Comparison with other initiatives (conflicting competencies?): no xkcd:927
- Finalizing appendix
- Publishing preprint + Opening community discussion
- Feedback still very much appreciated!

Follow-up papers:

- Institutionalized Organization of RSE Education
- A Survey of Initiatives Providing Educative Material in the RSE Space
- Educating RSEs in Germany - What Needs to be Done

All three very open for contributions!

Remark

The paper(s) are managed in a GitHub repo. Short link: <https://go.uniwue.de/teach-rse>

Thank you for your attention!