



Foundational Competencies of an RSE: State of the project and what comes next

Florian Goth (University of Würzburg), Toby Hodges (The Carpentries), Jan Philipp Thiele (WIAS)

Overview



The teachingRSE project within de-RSE e.V. focuses on skills that RSEs and domain researchers should know and learn. We have written a first paper on competencies and responsibilities of RSEs^[1] and a material collection on learning and teaching^[2]. We will continue working on getting teaching of these competencies incorporated into institutionalized education.

Competencies

Software/Technical

These skills revolve around designing and implementing software as a user-oriented and future-proof product. Naturally, there is a large overlap with traditional software engineering. However, the emphasis in the skills is on achieving good scientific practice and serving the specific needs of research software.

- Adapting to the software life cycle (SWLC)
- Creating documented code building blocks (DOCBB)
- Building distributable software (DIST)
- Use software repositories (SWREPOS)
- Software behaviour awareness and analysis (MOD)

Research

One of the differentiating aspects of research software is the research context in which it is developed, resulting in specific requirements and challenges. To take these into account and effectively work RSEs, need to have a good understanding of research.

- Conducting and leading research (NEW)
- Understanding the research cycle (RC)
- Software re-use (SRU)
- Software publication and citation (SP)
- Using domain repositories (DOMREP)

Communication

RSEs do not work in isolation. They are often embedded in a research group or are part of a more central RSE team. In both cases they are integrated in research projects where they need to interact with and facilitate communication among colleagues with a very broad spectrum of background-knowledge, specialization and experience.

- Working in a team (TEAM)
- Teaching (TEACH)
- Project management (PM)
- Interaction with users and other stakeholders (USERS)

Tasks and responsibilities To put the competencies into perspective we looked at some current tasks and responsibilities and mapped the required competencies needed for them, e.g.

- Testing (DOCBB,MOD)
- Contributing to larger projects (SWREPOS,SRU,SP,DOCBB,TEAM)
- Mentoring colleagues (SWLC,RC,TEACH)

How much of each competency do you different people need to know?

How much an individual RSE needs to know depends on RSE/academic career level, project/team structure and their specialization. An senior individual RSE embedded in a research domain will (need to) know different things than a junior RSE in a central team or a PI in a domain who wants to effectively employ and RSE.

Specializations

Within the competencies some RSEs will acquire/need a high level of knowledge for particular competencies

- Open Science RSEs need deeper knowledge of RC and of FAIRly distributing software publicly (SRU,SP) and can help researchers navigate the technical questions of practising Open Science
- Project/Community manager RSEs will need to know about PM, USERS and TEAM to effectively manage processes and people in larger research software projects.
- Teaching RSEs have specialized in TEACH, but also need a good understanding of all RSE competencies that they will teach to shape the next generation of RSEs and scientists.

Outside the competencies some RSEs will acquire additional skills to work on more specific areas

- \${DOMAIN}-RSEs will have more intricate knowledge about one particular research domain, e. g. mathematics RSEs, medical RSEs, digital humanities RSEs.
- Web-development RSEs have expertise in at least one of the skills of frontend development, backend development and API design and implementation.
- HPC-RSEs have knowledge needed for building computationally intensive software that can make effective use of parallel computing clusters and modern hardware.
- Research infrastructure RSEs have specialised in SysOps and administration and are able to set up and maintain IT infrastructures for and with researchers.
- Legacy RSEs specialize in maintaining and refactoring code bases that have been built on older programming languages and software stacks

Ongoing Work

Institutionalized education

In our follow-up paper we want to discuss the organizational infrastructure around teaching RSE topics at universities and beyond.

We want to look at:

- What makes a good teacher of RSE topics?
- How do we get those good teachers?
- How is teaching of RSE topics organized?
- How can RSE topics be offered in domain studies?
- Next steps for improving RSE education
- Extra-academic institutions for continuous learning?
- How do we reach people at different career stages?

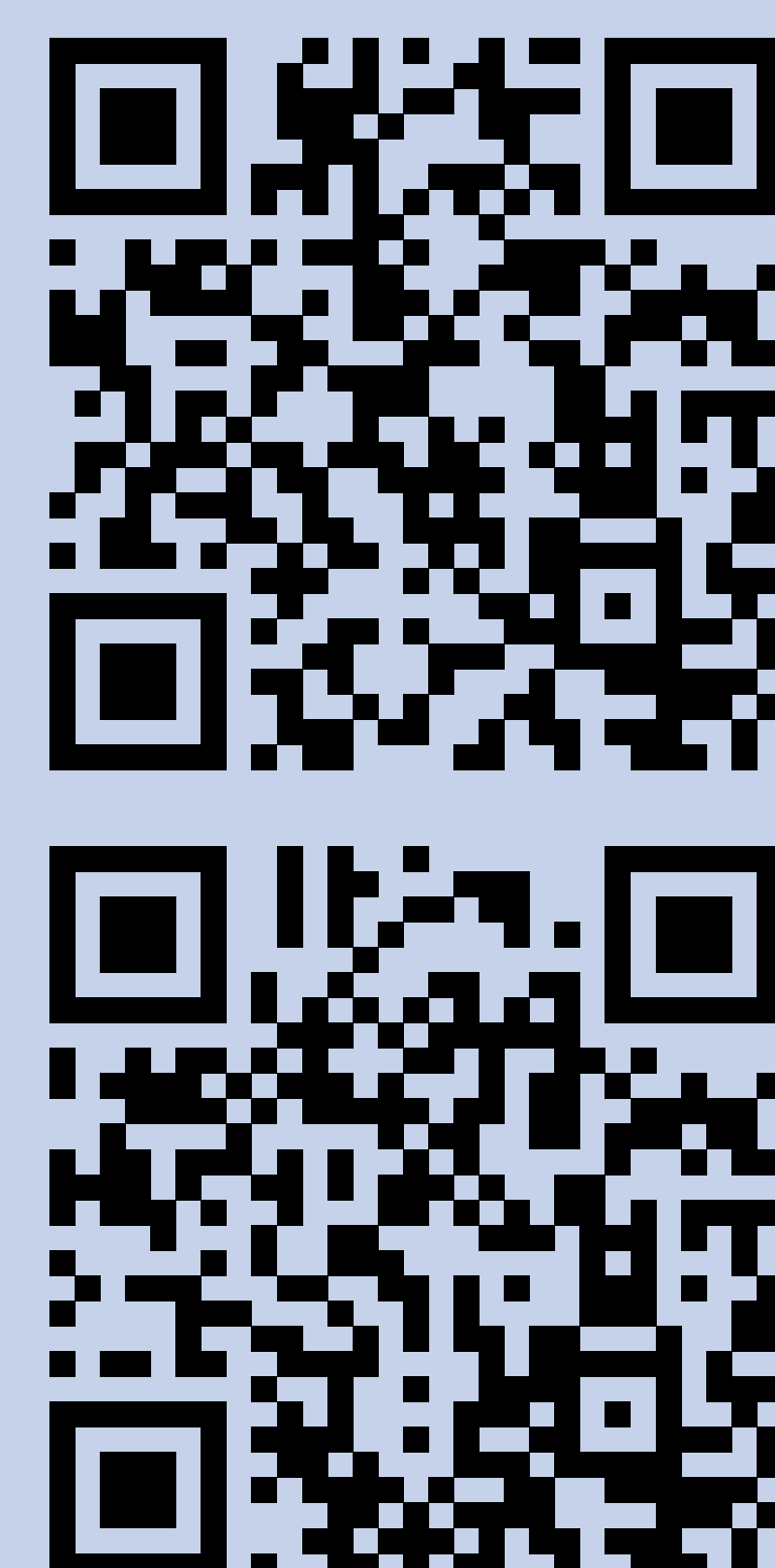
RSE Master curriculum

We started working on a curriculum for an RSE Master^[3]. As a Germany-based project the general framework is specific to our academic system with a Masters degree taking 4 Semesters of 30 ECTS each.

We envision this programme to be interdisciplinary and to foster communication and cooperation between students of different domain backgrounds (including computer science) early on. Recently, we have received funding by the Klaus Tschira Stiftung for a project coordination position to help with planning, advertising and monitoring of RSE master courses at interested universities.

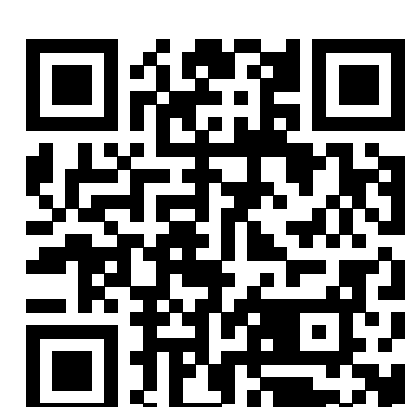


Questions for you!



for everyone

for RSE teachers



^[1] arxiv.org/abs/2311.11457



^[2] de-rse.org/learn-and-teach/



^[3] github.com/the-teachingRSE-project/RSE-Masters

