Teaching and Learning Research Software Engineering

Heidi Seibold Florian Goth Jan Linxweiler Jan Philipp Thiele
Jeremy Cohen Renato Alves Samatha Wittke Jean-Noël Grad
Fredo Erxleben Magnus Hagdorn Harald von Waldow

July 28, 2023

Contents

Working Title: Identifying foundational competencies of Research Software Engineers across all specializations.

Abstract: Being an outcome of a community workshop held in Paderborn, Germany in February 2023 this paper tries for the first time(FIXME: ? true?) to define which competencies are required to participate in modern digital sciences. Some of these competencies are required in more depth, therefore, giving rise to the trade of the RSE: scientific personnel that specializes in writing research software that facilitates work in all stages of the research cycle. Due to their generality, these competencies are often shared between RSE specializations, and we believe they are also relevant for domains outside of the RSE community.

But knowing a set of competencies is not enough, therefore we discuss explicitly how to make people aware that these skills are required and how these are taught(FIXME: Do we want to add this pedagogical dimension?). In order to also facilitate structural change in the German research institution landscape we will discuss the organizations and structures that support this change and educate new RSEs. The discussion in this paper is meant to be general. Therefore, we will discuss domain specific applications in an appendix.

Keywords: research software engineering, training, learning, competencies

Introduction

- background
  - RSEs have been around for a long time but the name is new.
  - RSEs have a specialist skill set that brings together technical and research knowledge.
  - Skills development traditionally provided largely through peer learning, self learning, introductory training courses not targeted specifically at RSEs, …
  - Requirements for RSE skills growing rapidly across all domains.
- past attempts, other initiatives
- contributions

Computers and software have played a key role in the research lifecycle for many decades. Traditionally, they were specialist tools used only in a small number of fields and the Computer Scientists who maintained and programmed them needed extensive technical training over several years to gain the necessary skills and expertise. Fast forward 50-60 years and software and computation are all around us, underpinning our everyday lives. This shift is also true within research.

With the ability to capture and process ever more data, undertake larger scale, higher resolution simulations and, increasingly, leverage new self-adapting approaches through Machine Learning, computers and software are now vital elements of the research process across almost all domains. However, this shift means that basic research software skills are now required by researchers of all career-levels across a vast array of research fields where these were not previously required. Researchers often lack the skills to write and use software for their research and even to effectively request help from and interact with more experienced staff at their institutions. There still exists a gap in the academic education, as many curricula do not sufficiently prepare their students in that regard. This situation is exemplified by the extracurricular MIT class "The Missing Semester of Your CS Education" [149], which aims to convey computing ecosystem literacy even to students of Computer Science at MIT.

The need to access both research data and software has been formalised with the FAIR principles: software and data need to be easily findable by both people and machines, and they also need to be accessible, interoperable and reusable. More recently the FAIR principles have been extended specifically to research software [10].

The people who focus on writing research software are now known as Research Software Engineers (RSEs) - a term that was coined a little over 10 years ago [69]. RSEs may work within one of the many Research Software Engineering teams that have been set up at universities and research organisations over the last decade, or they may be embedded within a research team. They may have a job title that officially recognises them as an RSE, or they may have a standard research or technical job title such as Research Assistant, Research Fellow or Software Engineer. Regardless of their job title, RSEs share a set of core skills that are required to write software, understand the research environment and ensure that they produce sustainable, maintainable code that supports reproducible research outputs. They are the ones who implement the FAIR principles that make digital research output more valuable. In order to do so they draw upon skills from traditional software engineering, established research culture and a commitment to being part of a team.

Developing and maintaining these skills is time consuming and often challenging. Part of the challenge is that there is not a standard pathway to becoming an RSE and, partly as a result of this, there is something of an ad hoc approach to training within the community. We also see increasing amounts of basic-level training materials that are great to put researchers or technical professionals on a path towards gaining significant RSE expertise, but the trail often ends as developing RSEs want to progress to intermediate and advanced level material. In particular, recent technology developments are ensuring that there is a growing need for specialist expertise, for example in areas such as making efficient use of high-performance computing infrastructure. This is an area where there is a skills shortage and a shortage of training materials.

In this paper, we look at the challenge of understanding the core competencies that underpin Research Software Engineering and the way that these competencies may be combined to help support a more coordinated approach to future RSE skills development. The paper builds on a workshop session held as part of the German Research Software Engineering Conference (deRSE23), held in Paderborn, Germany in February 2023.

[Information on key contributions to add]

'Related work and activities' gives a brief overview over other initiatives in the area of digital skills for researchers. The content is twofold. Firstly, we consider other initiatives trying to identify relevant skills for and pathways to becoming an expert at their specific but related area, e.g. high performance computing, . Secondly, we look at groups and organisations that compose specific training material that can be used for teaching RSE skills.

'Challenges' discusses the challenges in the current academic training and research landscape when it comes to students and researchers learning the necessary skills to write FAIR software in their specific domain.

In 'Results' we describe the main results of the workshop, namely the set of relevant skills as well as the levels of expertise needed in these skills for students and researchers at different career levels. Furthermore, we look at different RSE specializations which may additonally need skills not relevant to RSEs of different specialisations. The skills themselves belong to one of the three main categories needed to perform RSE tasks. Those are the 'R', i.e. an understanding of how research is performed, the 'SE', i.e. writing and maintaining reusable software, as well as team skills.

'Organizational Infrastructures' focuses on changes that are required to at best support the training of new RSEs.

FIXME (finalization): Exchange chapter titles by numbers?

Some Definitions

A few definitions are in order. First as software we define source code, documentation, tests and all other artefacts that are created by humans during the development process that are necessary to understand its purpose.

We define research to encompass all domains of research. Since we expect a sizeable portion of readers from Germany we quickly want to address a common false friend. The German term 'Wissenschaft' encompasses all domains of publicly funded research, while the English term 'science' is generally limited to natural sciences. Therefore, we will use 'research' to encompass all domains therefore gives the employability of RSEs. Of course 'research' as well as 'Forschung' is not limited to public funding but

also part of industrial and other private companies. We surmise that the same software engineering and team skills will be needed there, but we limit our considerations to the views of public research.

This enables us to define Research Software in this paper to include source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose. This definition is broader than in [10] and is the outcome of a recent discussion in [58].

Using this, Research Software Engineers are now people that create or improve research software and/or the structures that the software interacts with in the computational ecosystem of a research domain. They are highly skilled team members who can also conduct their own research as part of their role.

Intended Target Audience

While this paper is based on discussions held during a workshop at the deRSE23 conference we believe that the competencies formulated here have a far-reaching impact beyond the domain of RSE into adjacent fields of science. The most obvious users come from computer science, are HPC programmers with a background in physical sciences, or manage research data. Graduates from traditional STEM sciences with a focus on software or some library or IT staff will also find this paper interesting. However, these days most research involves some amount of data management, processing and visualisation and the role of RSEs is also becoming increasingly important in medical domains and the digital humanities. Access to central compute resources ranging from small departmental sizes to national facilities is becoming readily available. Additionally, pressure is growing from funding bodies to prioritise projects that generate archived, annotated, re-usable and potentially remotely executable data. These resources and requirements fall within the skill set of RSEs. They become a vital link to cross-pollinate computational skills and infrastructure know-how between domain scientists. Funders and research managers will find the discussion in this paper valuable in order to observe how software development in academia will be institutionalised. Finally, the strong emphasis on team-skills allows RSEs to be very employable in industrial workplaces.

National Context

Having been developed at a workshop in Paderborn in Germany, naturally a part of the discussion in this paper focusses on the German academic landscape. So, although there are Germany specific traits found in this document we are nevertheless dealing with the education of humans to become RSEs - A topic that is of major relevance also in an international context.

Related Work and Activities

The challenges of understanding the current state of skills within the research software community and related areas, as well as identifying required competencies, developing training pathways and providing training materials are areas that are being looked at and addressed by various groups and projects. In this section, we highlight some of these other projects and activities.

Identifying skills and pathways

The development of a standardized list of RSE competencies could help develop metrics to measure an individual's progression in specific RSE skill sets, e.g. using the Software Sustainability Institute's RSE Competencies Toolkit [121].

Special interest groups have in the past outlined the core competencies of various RSE disciplines, such as HPC-RSE [147, 82] (see Appendix: HPC skills and certification), bioinformatics-RSE [150, 103, 175, 159] (see Appendix: Bioinformatics skills and certification) clinical informatics-RSE [37, 36, 55, 16], librarian/RDM-RSE [49, 38, 128], community manager-RSE [176], and generalist RSE [123]. However, the present document is, to our knowledge, the first attempt at defining the skills of a generalist RSE at different levels of seniority.

FIXME:

- [30]
- skill gaps for software verification/testing (fig. 1 in [62]) and bioinformatics [97, 112]
- software evaluation criteria [78, 24]

RSE-related Training Materials

A wide range of software-related training materials and supporting organisations exist within the research software community and beyond.

The Carpentries The Carpentries [143] is a non-profit entity that supports a range of open source training materials and international communities of volunteer instructors and helpers who run courses around these materials. The community also maintains the materials which are based around three core syllabuses – Software Carpentry [133, 168, 169], Data Carpentry [35, 138] and Library Carpentry [92, 8, 29]. The training materials within these areas have been developed, reviewed and enhanced over several years ensuring that they represent best practice in training on these topics. The core Carpentries lessons are targeted primarily at the beginner level. However, the Carpentries Incubator [144] provides an environment for hosting additional community-developed training modules covering a wide range of other topics that have not gone through the peer review process of the core lessons. The material in the Incubator increasingly includes more intermediate-level training modules. After completion of a Carpentry workshop, learners can claim a certificate of attendance [142]. Carpentry instructors must be certified before organizing a workshop. There is a selection process [139] as well as an instructor training course, upon which a certificate is delivered [167].

Coderefinery CodeRefinery [28] is a project currently funded by the Nordic e-Infrastructure and thus active primarily in the Nordics with the goal of teaching essential tools around research software development, that are usually skipped in academic education. CodeRefinery hosts a set of open source training materials including both beginner and intermediate level material and organizes multiple highly interactive large scale workshops per year. Skills learned from the workshops and/or materials allow researchers to produce more reproducible, open and efficient software and thus promote FAIR [161] research practices. One goal of the project is to evolve into a community project that seamlessly integrates with other initiatives. FIXME: elaborate on the integration part if it's relevant, else leave out.

Reprohack The ReproHack Team offers resources to host events where students and researchers can get together to try and reproduce the results of published papers with the methods described there or ideally with the software provided by the authors. FIXME: This is applied FAIR principles elaborate a bit on why this is related for RSE competencies? Or do we want to make the point, that this teaches reproducible science?

PRACE The Partnership for Advanced Computing in Europe (PRACE) [116] offers training in the form of massive open online courses (MOOCs), online and on-site training events at European HPC facilities (aggregated on various websites, e.g. EuroCC Training [45]), and white papers. While most training events are tailored for HPC-RSE, several recurring courses about programming languages (C++, Fortran, Python) are suitable for general RSEs, as they teach coding best practices, modern software design [73], project management and version control [59].

Helmholtz As part of its push towards a better RSE environment, the Helmholtz Association launched the Helmholtz Federated IT Services platform (HIFIS) [65] which provides educational material and trainings amongst other services for an audience of over 10,000 scientists in Germany and internationally. All of these materials focus on RSE basics to refresh and expand the software engineering knowledge for recent graduates or to update the existing knowledge in established researchers. They are published under OER licenses and can serve as either self-learning instructions or form the basis of a hands-on training. To allow these educational offers to be easier brought to the scientists, the Helmholtz Information and Data Science Academy (HIDA) [67] sustains a large network within the Helmholtz Association and beyond with a strong focus on graduate schools. Further RSE training offers within the Helmholtz context are provided by the Helmholtz-AI [64] and Helmholtz-Imaging [66] platforms as well as the Helmholtz Metadata Collaboration plaftorm [68].

ENCCS The National Competence Center Sweden (ENCCS) has created a collection of lessons for HPC-oriented RSEs [44] and has adapted instructor training material from The Carpentries and CodeRefinery to create their own instructor manual [43, 42]. The ENCCS lessons are targeted at individuals who already have general RSE skills and are seeking new skills relevant to HPC and software engineering.

Open Source Resources

Due to the ever-evolving nature of skills and infrastructure in the RSE field, training material is often version-controlled, so that trainers can update it between iterations. For example, core lessons from the Carpentries and CodeRefinery are stored on GitHub, and any change is automatically mirrored to their website. Likewise, the reference work on RSE by Fogel [51] was released in its second edition as a living document [52]. The ENCCS instructor training manual [42] is also available as a living document [43]. The Carpentry instructor guide [170] is available as living documents in both English [171] and Spanish [165]. From the MIT Computer Science & Artificial Intelligence Laboratory comes an unofficial course The Missing Semester of Your CS Education which covers a lot of basic computer skills typically taught by RSEs. The skills covered here also provide an important set of core capabilities for anyone looking to become an RSE.

More generally, it is important for training material to follow the FAIR principles [161], so that the community of trainers and learners can make the most out of it [54]. Core lessons from The Carpentries were translated in multiple languages to make workshops more accessible to communities where the English language poses a barrier to learning (for more details, see section How do we reach people in different stages of their careers?). Material with rich metadata are indexed in online databases, such as the INTERSECT [17] collection [76] and the US-RSE collection [122] for all RSE specialisations, the ELIXIR [60] Training e-Support System (TeSS) platform [12, 6] for bioinformatics and life sciences, or the now-defunct Educational Resource Discovery Index (ERuDIte) [154, 3] for data science.

FIXME:

- Better Scientific Software (BSSW) …
- Software Sustainability Institute (SSI) [32] …
- Add NFDI initiatives like edutrain.
- ELIXIR/EXCELERATE/GOBLET train-the-trainer programme [155, 101]
- Library-RSE resources [26] (no longer updated since 2019)
- Happy Belly Bioinformatics [88]
- software energy consumption and "green software engineering" [113, 111, 110], teaching by Sus-Trainable [137, 81]
- reference works for self-study [51], [77], [170]

Challenges

- Point out gaps
- What is missing
- domain application?

Depending on the specific domain there is a gap between the basic software carpentry courses and the required skills to build domain-specific research software. For example, scientists in the field of High Performance Computing (HPC) need to know how to make effective use of concurrency to speed up their simulations and communicate efficiently using message-passing interface (MPI) libraries. The same is true for researchers from other domains who make use of other specialized technologies, methods and/or tools. To bridge those gaps more specialized courses would be needed like the one mentioned in section Identifying skills and pathways for the HPC community.

Moreover, software development is a craft, i.e. it is not only about knowledge but also requires practical experience. Hence we need to create an environment that allows less experienced researchers to practice and gain experience efficiently. Ideally, this learning environment would allow less experienced scientists to be guided by more experienced RSEs. We know such practices e.g. from human medicine, where junior doctors first assist experienced doctors before they work independently. In the field of software development, this approach could be implemented in the form of peer programming, for example. The prerequisite for this, however, is that experienced academics get better career opportunities at German universities so that they do not leave for industry roles.

Workshop Results

Required Generic RSE skills

The role of an RSE lies somewhere on the spectrum between that of a researcher (the "R") and a software engineer (the "SE") and, therefore, requires competencies in both fields. RSEs typically apply their knowledge and experience in larger teams which allows them to cultivate this hybrid nature. Therefore, we categorise the competencies into software engineering skills, research skills, and team skills with particular focus on the software and research cycle and the scientific process. The generic skills are relevant in a broad setting and form the foundation for specific specialisations.

These skills and competencies come into play in various forms: first of all the RSEs themselves need to acquire and develop them as their career progresses (**Career level**). However, some knowledge of software and data processing is required at all academic levels and for all positions (**Academic Progression/better title**). The relative importance of the skills and competencies also depend on the size of the RSE team (**Project team size**). Finally, different sets of skills are emphasised in the different RSE specialisations (**RSE specialisations**).

Software Engineering Skills

There are many software engineering curricula out there, that try to define which tasks a software engineer should be able to perform. A recent example highlighting some aspects in more detail than here is [85]. The software skills outlined here are required to make research software adhere to the FAIR principles. [24] defines different levels of research software reuseability and the extent to which the software engineering skills need to be applied to reach them.

Creating documented code building blocks (DOCBB) The RSE should be able to create building blocks from source code that are reusable. This ranges from simple libraries of functions up to complex architectures consisting of multiple software packages. An important part of reusability is that at least oneself, and ideally others, are able to understand what a piece of code aims to do and how to use the provided functionality. This is primarily achieved through a "clean" implementation and enhanced by documentation. Documentation ranges from commenting code blocks to using documentation (building) tools.

Building distributable libraries (LIBS) The RSE should be able to distribute their code with their domain/language specific distribution platforms. This almost always encompasses handling/documenting dependencies with other packages/libraries. It sometimes requires knowledge of using build systems to enable interoperability with other systems.

Understanding the software lifecycle (SWLC) The traditional software lifecycle defines the stages that form the process of building a piece of software. This generally involves a creative process where you try to understand what it is that you need to build, work out how you are going to build it and then implement it. This is then followed by testing that things work as expected and that they continue to do so into the future. We emphasize the the lifecycle is not complete here but also includes periods of maintaining a software and also phasing out a software of its original use.

Use repositories (SWREPOS) The RSE should be able to use public platforms to share the artefacts they have created and invite the public to scrutinise them for public review.

Legal aspects (LEG) The RSE should know licenses and their respective domains for data or software. On an entry level, the competency is mostly about awareness: namely that different (open source) licenses exist, that those might not be compatible with each other, and that use of third party software might restrict licensing of the resulting work.

Software Behaviour Awareness and Analysis (MOD) We define this as a certain quality of analytical thinking that enables an RSE to form a mental model of a piece of software in a specific environment. Using that, an RSE should be able to make predictions about a software's behaviour. This is a required skill for common tasks like debugging, profiling, designing good tests, or predicting user interaction.

The research skills

Curiosity (NEW) RSEs gain their reputation from their effectiveness to interact with their domain peers. Therefore, some curiosity together with a broad overview of the research field is required. Curiosity is also reflected when an RSE is actively trying out new tools. Lifelong learning is then no longer just a phrase, but becomes a motivation to work.

Understanding the research cycle (RC) One of the crucial skills of RSEs is their mental proximity to research and they embrace being part of a larger community which, despite friendly competition, shares the common goal of gaining knowledge for its own sake and not just for personal or commercial gain. Thereby they know, that they are part of a bigger cycle that involves many other parties in and outside of their domain, and also that their software can be utilized in different stages of the research cycle by different persons. Like other researchers, RSEs are open to discussions and arguments beyond their own expertise and appreciate the underlying principles of good research, like publications, review and reproducibility.

Finding/discovering software and attribution (SD) One goal of FAIR software is to avoid unnecessary duplication of work by reusing existing work instead. To (re-) use software, researchers have to be able to find it and then to easily evaluate if the software actually suits their needs. Apart from functionality also licensing, integration with other software, expected sustainability and expandability have to be part of this evaluation. Finally, after obtaining/publishing results by modifying and/or using the software, the original authors need to receive proper attribution.

Using domain repositories/directories (DOMREP) Almost all research software is developed within a specific scientific domain. Some software may be able to cross boundaries, but the majority will have a home domain, with which it needs to be able to interact. The RSE needs to be aware of any domain specific software repositories, data sets and catalogues and the RSE's software needs to be able to interact with the existing domain-specific data repositories.

Outside Party interaction (USERS) Research software is often developed as part of the research process itself, and like research, it will change in unpredictible ways hence it often has to be developed very closely to its users, the researchers. Therefore, roles like developers and users can seldomly be distinguished as most people represent multiple roles ranging from end-users, up to funders. However, regardless if this is the case or not: compared to other SE environments, there is an unusually close interaction between and within different roles in research, as well as between experts from different domains. Often this means it is necessary for an RSE to think "outside their comfort zone", but at the same time to be able to convey their knowledge and experience to experts of other fields in a way they can understand more easily. This includes their own domain knowledge in discussions with RSEs (from other domains), as well as their SE knowledge when talking to domain scientists and also the exchange of new techniques and algorithms to keep their software up-to-date.

Team Skills

Teaching (TEACH) Working in a group means being able to effectively perform e.g. onboarding, or more formal teaching procedures to their colleagues. This includes tasks such as consulting and mentoring since these also often aim at educating people. We deliberately mention, that giving code reviews is also part of this skill, since Code review can be part of teaching people on improving their skills.

Project Management (PM) The RSE should have knowledge about project management. At some institutes, it follows the practices of the local research groups, but it is useful, if an RSE knows its place in a PM scheme, or can bring in new ideas for improvement.

Working in a team (TEAM) There are various facets to working in a team. They range from functioning in a team to leading a team. It includes following measures that increase team cohesion like performing code reviews.

RSE Tasks and Responsibilities

These skills, while already numerous are also generic on purpose. They span a multidimensional space in which the day-to-day tasks and responsibilities of a RSE can be found. A snapshot of what this means today can be obtained from learners and novice RSEs that we asked during the Paderborn workshop what they would like to have learnt. Among the top five things mentioned were:

- Testing. This task is a manifestation of the SE competencies of DOCBB and MOD since a model of the software is required in order to write good tests that facilitate understanding and documentation. Today this encompasses the knowledge of testing frameworks and CI/CD practices.
- Contributing to large projects. This is a topic that requires competency in SWREPOS, LEG, in order to understand the ramifications of sharing, and DOCBB, since the contributed code has to be understood by others. Interacting with project members depends on the the TEAM skill. Today this entails the effective use of collaborative platforms like github/gitlab, honouring a projects Code of Conduct, and some knowledge of software licenses like the GPL.
- When or why to keep repositories private. This decision requires knowledge in the RC, to understand when it makes sense to open up or close down a repository. The USERS, TEAM and sometimes LEG skills are required to make this decision. Furthermore, knowledge of the practices and contractual regulations of the RSE's institution are also required.
- Proper Development. This broad topic requires all of the SE skills. Of course these are the competencies that are the most fluid since they have to adapt at a high rate to the technological advancements. Additionally proper SE skills often require knowledge of TEAM, and PM. Today this means effective use of IDEs, static analysis tools, design patterns, documentation (for oneself and others), etc.
- Finding a community. This can be interpreted in two different facets. First we have the aspect of community building for a research project. Since this deals with software that is supposed to be used in research this requires knowledge of RC, USERS, and also NEW, in order to effectively interact with domain scientists. Today, an example is a presence on social media. The other TEAM-related aspect is the embedding of RSE graduates into the community of RSEs. We envision our RSE graduates to be a part in a strong network of other RSEs, tool-related communities and the classical domain communities. This point is further elaborated in How do we reach people in different stages of their careers.

Beyond that, we feel that today Other important tasks of RSEs are

- Mentoring colleagues. This necessitates giving good advice that fits to a projects stage in its lifecycle, thereby requiring knowledge of (SWLC), and its context in its research domain, thereby requiring knowledge of (RC). Research software often starts out as a tool to answer a personal research question and becomes more important when other researchers rely on it. Some research software might even be used to deal with critical questions such as weather forecasting or medical diagnosis. To formalize the process of giving good advice a classification of software is commonly used [156, 127] where research software can move from one class to another during its lifecycle. [127] classify applications based on their scope and criticality and provide software engineering recommendations. The RSE needs to be able to identify the application class they are dealing with and apply the respective RSE practices.
- Enforcing reproducibility. Projects like [119] can greatly help in fostering that competency.

How much do different people need to know?

Now that we have the different competencies, we can explore various dimensions of these competencies, depending on their circumstances. A strong beneficiary of specialized RSEs can also be newly formed RSE centers at research institutions.

Career level

At different career levels, differing skills are required. We have set this up according to the following separation often applied within a single project:

- Junior RSE: These are persons that have just started, but generally speaking they should have the skills to contribute to software projects
- Senior RSE: They have gained experience and can set the examples in the software project.

- Principal RSE: Their actual job description varies a lot. These may be RSE team leaders based in a professional services type role, or they may be professors or research group leaders based in a more academic-focused role. They are often the people responsible for bringing in the money that supports new projects and sustains existing projects. Generally speaking, they do not need to to be actively involved in the day-to-day technical tasks but they should be able to guide projects from both a technical and research perspective.

The required skills are distributed according to this table First Dimension: Career path e.g. Junior RSE -> Senior RSE -> PI scale (1->6) (less -> lot)

	Junior	Senior	Principal RSE(brings in funding)
DOCBB	should be able to write reusable building blocks	same as junior, but the quality should set the standard for the project, while following current best practices	should know the current best practices and point its people to the right resources.
LIBS	should be able to use package distribution platforms	same as junior, but should set the project standard and follow current best practices.	should ensure that their project is in an up-to-date distribution platform
MOD	should have a basic grasp of their piece of the software in order to use basic tools like a debugger	Should understand the characteristics of large parts of the codebase considering a variety of the metrics	Should understand the big idea of the software project in order to define the task that the software solves
SWLC	Awareness about the existence of the software lifecycle.	Should know which decisions lead to technical debt.	Should know in which part of the lifecycle their project is and how to steer development/project resources accordingly.
SWREPOS	Seamless interaction with the swrepo of their project is a must	Should be well-versed in the intricacies of a swrepo, and probably interact with multiple projects' repo's	Should be able to effectively interact with swrepos and especially the interaction with the connecting projects.
LEG	Awareness about legal intricacies about sharing code	Should be able to give advice on legal issues and resolve the most common issues	same as Senior RSE
NEW	Some curiosity required to fit into research teams	same as junior, but a curiosity to enhance the code base is required	Curiosity to know in which direction to steer the project is required

	Junior	Senior	Principal RSE(brings in funding)
RC	Awareness about the RC	should know the position of the project in the RC	Should know what is necessary for the project to fit into its position in the RC
SD	Should be aware about tools for SD	Should be able to find sth. with SD tools	Should be able to effectively find sth. with SD tools and be able to evaluate and perform the integration of a library into the project.
DOMREP	The RSE should be able to interact with the domain repository	same as junior RSE	same as junior, and should know about how it fits into workflows surrounding these domain repositories
USERS	The RSE should be able to communicate with non-SE users of the project	same as junior	same as junior, and take feedback into account of the steering
TEACH	should be able to perform simple peer-to-peer onboarding tasks	should be able to explain logical components to other RSEs	Should be able to effectively communicate about all large-scale parts of the project.
PM	Awareness about the employed project managemement method	Should be able to use the employed PM method	Should be able to design and adapt the employed PM method.
TEAM	Should be able to work in the team in order to effectively fulfill the given tasks. Should be able to learn from code review.	Should be able to break down tasks into more easily digestable sub-tasks	Should be able to lead the team and set the respective direction.

Academic Progression / Career Path? (Help me for better title)

Modern digital science requires some digital proficiency at every level. To be a bit more precise, these are how we define the academic levels:

- Bachelor: These are people in their undergrad studies, that mostly consume science/knowledge. They should also learn about the existence of certain digital structures.
- Master: Ultimately, their study should have brought them to a level, where they can participate in science, hence they should be able to use "some" digital structures.
- PhD: Under guidance they perform independent research and hence they should get to know all relevant structures.
- PostDoc: Independent researchers, they are proficient users of all tools.
- PI/Professor: Experts in their field, they should be able to give proper guidance to their students on which digital tools are currently relevant.

It is important to note that the following table does not reflect the current state of academic training and research institutions. Instead, it summarizes the discussions with and between workshop participants at different levels of academic progression on what they would have liked to learn at an earlier stage or know before starting their current position. While individuals already work at implementing some of these changes and teaching these skills it has not yet reached a systemic level.

Additionally, this table tries to cover all domains that rely on software tools in at least a basic level. Certain fields, e.g. sciences relying on simulations, might require higher skill levels in the SE competencies as software development is a large part of their actual research.

	Bachelor	Master	PhD	PostDoc	PI/Professor
DOCBB	They should be aware that RSEs exist and that software has different quality aspects	Same as Bachelor	They should know where they can get help, and maybe able to use libraries	same as PhD	They should know the skills of an RSE and when they might need one in their group
LIBS	They should be aware that RSEs exist and that there are tools available in their domain	They should be aware that there are tools that they can use in their research and maybe are able to use these libraries	same as Master, but able to use libraries	same as PhD	They should be aware of the output of RSEs and motivate their students to use developed tools
MOD	It is sufficient to consider digital tools as black boxes	It is sufficient to be able to *use* software as black boxes	same as Master, but being able to write bug reports	same as PhD	same as PostDoc
SWLC	Awareness of the SWLC	Know that one depends on software in their own research	Being able to evaluate software for their research	same as PhD	Should be able to judge the sustainability of the performed research

	Bachelor	Master	PhD	PostDoc	PI/Professor
SWREPOS	Should know that swrepos exist	same as Bachelor	same as Master, but should be able to find information on them	same as PhD	same as PostDoc, but should be able to follow the interactions among different projects relevant for their research
LEG	Should know that mixing/using software has legal issues and whom to ask	same as Bachelor	same as Bachelor	same as PhD, but should know some simple Open Source guidelines	same as PostDoc, but should know the relevant patterns for their domain and sensitive their students
NEW	Still consumers of lectures	same as Bachelor	Curiosity for their research is required, curiosity for digital tools helpful	same as PhD	same as PostDoc and expert in their field
RC	An awareness that research follows a cycle	Know that research follows a cycle and locate their masters thesis' stages in it.	Same as Master, but applied to the PhD. Additionally awareness about interaction with services	Same as PhD. But proficient in the domain	Same as PostDoc, but ability to lead a topic

	Bachelor	Master	PhD	PostDoc	PI/Professor
SD	They should know that their domain has relevant tools	same as Bachelor	Should know how to find full applications for their research	same as PhD	Should motivate their students to reuse existing tools
DOMREP	Should be aware that their domain has repos	same as Bachelor	Should be able to interact with their domain repos	Proficient users of their domain repos	same as PostDoc
USERS	Should be aware that they are users of a software	same as Bachelor	Should be aware that their user view is different from the developer, in order to write bug reports	same as PhD	Should be able to contribute meaningfully to the steering decisions of the software in their fields
TEACH	Ability to peer-to-peer teaching	Small exercise groups	Ability to supervise a student.	Ability to supervise students and create a course?	Ability to guide students. Give full-size lectures
PM	Awareness about project management optional	Awareness that research teams are structured according to some project management	same as Master, or more depending on structure of research	same as PhD	Should know about the required project management they require for their group
TEAM	Awareness that research is often performed in groups	Ability to work in their group for doing their master's thesis	same as master	same as master	Should be able to lead a research team

Project team Size

Some explanation of the team sizes:

- individual: A single person working on their research software
- Small team(~4 persons) This is a small team, that has decided to work together on something
- Organizations(>10 persons): These are big organizations with clear structures and a bigger degree of specialization.

	individual	small team	organization
DOCBB	you might get away with less satisfactory code, as long as the product is OK	think about your colleagues	your organization most likely has guides here
LIBS	you will only be successful if your artefact is usable by others	same here	your organization probably has rules here
MOD	you should precisely know what your entire code is doing where	you should know what your part is doing and have a feeling about the others contributions	You should know what your small part is doing
SWLC	it's you and your software	You should know the Bus factor	The organization takes care of that
SWREPOS	you need academic credit.	same here	your organization probably has rules here
LEG	you carry the responsibility	someone in your group needs to take car of this	your organization will have specialized people for it
NEW	You need a motivation to do this alone	?	Not so much, since other people might do this task
RC	?	you should maybe talk among your peers where your software fits in	You define your research cycle
SD	you need to be able to build on other work to be successful	same here	there might be someone in your organization who does this
DOMREP	You're doing science in a domain	there should be a person in your team who knows how to do it	your organization might have specialists for that, but some basic familiarity
USERS	at one point you hope to have users	same here	maybe you have specialists for outreach
TEACH	N/A	able to peer teach	teaching to groups

	individual	small team	organization
PM	Not much required	able to follow checklist	Working with PM tools, or use them for organization
TEAM	N/A	should be able to give equal feedback to their colleagues	should be able to work within their role

Bonus points may be distributed if managing teams remotely

BIO Excel framework

RSE specializations

What we have defined above are intended to be base skills that an RSE irrespective of domain, place, and time should know about. But not all RSEs are created equal, they specialize in different areas, some of which we want to present below. Many of the specializations may overlap, so the same RSE might for example work on data management and on Open Science.

HPC-RSE

RSEs with a focus on High Performance Computing (HPC) have specialist knowledge about programming models that can be used to efficiently undertake large-scale computations on parallel computing clusters. They may have knowledge of (automatic) code optimization tools and methods and will understand how to write code that is optimized for different types of computing platforms, leveraging various efficiency related features of the target hardware. They are familiar with HPC-specific package managers and can build dependencies from sources. They also understand the process of interacting with job scheduling systems that are often used on HPC clusters to manage the queuing and running of computational tasks. HPC-focused RSEs may be involved with managing HPC infrastructure at the hardware or software level (or both) and understand how to calculate the environmental impact of large-scale computations. Their knowledge of how to run HPC jobs and write successful HPC access proposals can be vitally important to researchers wanting to make use of HPC infrastructure.

They may also be familiar with High-Throughput Computing (HTC) and manage a network of heterogeneous compute resources, typically desktop workstations equipped with multicore processors and possibly GPU accelerators. They can apply their node-level performance engineering skills to maximize utilization of the available resources. Finally, they typically have expert knowledge in at least one compiled language, and can assist domain scientists who have excellent command of scripting languages but only a cursory understanding of compiled languages get up to speed with compiled software.

Research Infrastructure RSE

This RSE is interested in SysOps and sets up infrastructures for and with researchers. This RSE, therefore, requires a deep knowledge of physical computer and network hardware. FIXME: While required, is this an RSE? this sets off the usual infrastructure vs. research discussion….

Web-Development RSE

This RSE is skilled in web applications, front- and/or backend, and/or building and using APIs, for example for research data portals or big research projects. Ideally, this RSE should also have knowledge about (web) accessibility to allow a broad range of researchers or even the public to use the resulting applications. Therefore a deep knowledge of web skills is a required skill for this RSE.

Legal-RSE

With the prevalence of software, we foresee the need for RSEs that specialize in legal questions around software. They are the go-to person if people have a question about licensing, mixing and matching software, and/or patenting.

Data-focused RSE

RSEs working at the flourishing intersection between data science and RSE. They are skilled in cleaning data and/or running data analyses and can help researchers in setting up their analysis pipeline and/or research data management (RDM) solutions. When the field requires research on sensitive data or information, e.g. patient information in medicine, this RSE should have knowledge about secure transfer methods and/or ways to anonymize the data.

OpenScience RSE

Open Science and FAIRness of Data and Software are increasingly important topics in research, as exemplified by the demand of an increasing amount of research funding agencies requiring openness. Open Science RSEs can help researchers navigate the technical questions that come up when practicing Open Science, such as "How do I make my code presentable?", "What do I need to consider when it comes to licensing?", or "How can I use version control / automation for my project?".

Project/Community manager RSEs

When research software projects become larger, they need someone who manages processes and people. Building a community around a research project is an important building block in building sustainable software [130], so these RSEs play an important role, even if they do not necessarily touch much of the code themselves.

Teaching RSEs

RSEs who focus on teaching the next generation of researchers and/or RSEs play a vital role in quality research software.

${DOMAIN}-RSE

While software is the lingua franca of all RSEs there will be RSEs that have specialized in the initricacies of one particular research domain, such as medical RSEs, digital humanities RSEs or physics RSEs.

Optional RSE competencies -> Maintenance RSEs

Oftentimes, a significant amount of effort in (research) software development needs to be spent on maintenance to ensure that software remains useful for researchers now and in the future. The research environment is constantly changing and this can also apply to the software requirements. Accordingly, software often needs to be adapted continuously. If it isn't, the software can reach a point where it simply isn't useful to the researchers anymore. To avoid this, regular work needs to be invested. While ensuring maintenance and sustainability of research software is of huge importance to the communities that build and use it, a particular challenge is that it's often very difficult to obtain ongoing research funding for software maintenance tasks. As a result, when a project that developed or extended a piece of software finishes, it can often be the case that support for the software fades as team members move on to other research, academic or RSE roles, or become busy with other funded work. While this is not a core concern of this paper, we wanted to highlight this important issue that is frequently faced when working with software in the research community. With regard to which additional competency is required, these are people having experience with ancient software stacks that are not anymore part of the general curricula(think of COBOL and FORTRAN).

FIXME: I think it would be nice if we could move each of these optional competencies to a different specialization.

How do we reach people in different stages of their careers?

Many current RSEs have found their way to being an RSE during their doctoral studies. This transition usually happens slowly. You start programming a tool, and someone else likes it, it becomes known that you have programming skills and suddenly you are the RSE of the group that everyone would like to have in their projects. If you enjoy this role, you need to be aware that there is a RSE career path as well as that specialized training materials exist. One place to generate awareness of the career option and training is universities' doctoral onboarding processes or right thereafter. RSE training could be offered as elective courses at universities organized by some central organization. RSE could be presented as a career path in suitable events. Since many RSE-minded people also at some point find their way to an HPC cluster, mailing lists of said clusters could be utilized to advertise RSE courses. One important aspect to think about is also the wording in the advertisement. Potential future RSEs might not know the term yet or know that the course advertised includes topics that are of interest to them. If the university or organization has a GitHub/Lab organization/project, having a banner there might reach the right people. Most important is that people working in IT helpdesks know about the courses offered so that they can advise students/researchers on visiting the course/reviewing the materials if related

questions are asked. For an RSE it is important to be a part of a network with other RSEs, irrespective of the career level. A perfect first step for forming this network is topic-related conferences, workshops or meetups. Beyond that, there is the broader RSE community organized at the local and regional level with chapters or local/regional communities, at the national level with societies and the international RSE society. Each of them offers possibilities for connecting within or beyond an individual institution and is a great way to find like-minded people to grow a wider network and thereby facilitate the sharing of information on interesting events or help each other out. This available layered network can greatly benefit the RSE in finding help with issues outside of their own comfort zone and provides a welcoming, social safety net providing a home for the RSE. Since we feel providing aspiring RSEs this net is of utmost importance we envision compulsory events introducing that to young RSEs. Qualification badges are another venue, that RSEs to find people with the same technical interest.

Short primers on RSE skills, infrastructure and good coding practices can be found in field-specific scientific articles and conference proceedings, such as [120, 11, 117, 91, 174, 135, 33, 31, 50, 57], some of which are specifically tailored to group leaders, institutions and scientific journal editors rather than RSEs [25, 24, 80, 136]. Scientific journals have the advantage of reaching a large spectrum of research scientists at all stages of their career.

Localizing RSE teaching material and RSE information in languages other than English can help reach a much wider audience by lowering the barrier to entry in the field. In 2014 the community behind The Carpentries engaged in an international effort to translate their training material into Spanish [173], Korean [172] and Portuguese [132]. Core lessons have been translated to Korean in 2015 [89], and the Spanish core lessons are now officially part of the Software Carpentry material [140]. Similarly, in the period 2015-2017, the Stack Overflow website launched localized versions in Portuguese [114], Russian [115] and Spanish [96] to reach a wider community. There are also RSE short primers [5] and RSE guidelines [27, 4, 61] in non-English languages to address the need of specific communities.

Teaching RSE in relevant undergraduate courses of domain scientists can be the first point of entry in the field. However, considering many RSEs come from domain studies, only fundamental concepts (of RDM and RSE) can be explained to and experienced by all of them. Those interested in domain-specific RSE skills or even programming will gain the special knowledge in classes and projects they choose. For example, statistics curricula can be used to showcase RSE infrastructure, e.g. the R programming language and its ecosystem of statistics libraries and integrated development environments [118, 13, 20]. There are also bioinformatics courses designed for high school students that cover topics such as pen-and-paper algorithm design, genomic database querying and data mining, and open data [53, 7], as well as graduate courses designed for Master's degrees and Ph.D. programs [75].

Teaching incubators can be leveraged to develop and test new academic curricula that introduce basic RSE topics, such as the "Algorithmic Battle" [124] (version control, documentation, good coding practices), "digit@L" [153] (coding, data analysis, machine learning) and "DigiFlex" [74] (digital tools) experimental modules funded by the German Foundation for Innovation in Higher Education [134] to reduce skill gaps among first-year university students. Likewise, The Carpentries teaching material can be made more modular and re-usable in domain-specific contexts to better suit the needs of specialized RSEs. Examples include HPC Carpentry [107] and Data Carpentry for Biologists [160].

FIXME:

- find more examples of teaching material in non-English languages
- discuss the role of translations in overcoming linguistic and cultural barriers?

Further ideas:

- making RSE best practice guides fun to read with memes or satire [9, 84, 5, 50]
- reducing the skill gap by organizing more inclusive workshops [100, 2, 131, 104], to address gender disparities or take into account economic status, cultural background, or special educational needs; peer-reviewing of code of conducts by the CHAOSS Diversity and Inclusion Badging organization [22, 23]
- related discussion in CSE [157]

Organizational Infrastructures

So we have defined our set of competencies that we feel every RSE should possess. Table 2 above nevertheless already hints at the fact that some RSE skills are required during the domain studies, while Table 1 tells us that we also need an ongoing qualification programme for people that want to become specialized RSEs. In order to set up a proper educational scheme we need to discuss two more items:

- Who are our teachers?
- How is this teaching organised?

The teachers

What issues are trainers facing today? There are already some people out there who are teaching RSE related topics sometimes in university structures, but often outside of formal structures. The community discussion shed some light on the issues our trainers are facing now. Currently, they are often teaching workshop like formats in research institutions.

- There are outreach issues. We emphasize that there are two dimensions to this: First it is important that we inform students that workshops exist, and then, the more important part, we also need to motivate people to invest the time for a workshop. [18]
- Adaptation of material to the target audience has been identified as a time consuming task.
- Organization and preparation is a challenge, since currently no standardized formats exist.
- Expectation management of students. Existing knowledge of students is often diverse.
- Language barriers. This can range from the use of technical jargon up to the disparities of you teaching in a foreign language.
- Setting up a feedback loop that facilitates a reflection of the workshop for the teacher.
- staying up-to-date with fast-moving RSE topics.
- Understanding the difficulties of students [19, 162].
- Carpentries retrospective [169].

What mindset makes up a good teacher Irrespective of where people come from they need to have the proper mindset to properly foster aspiring RSEs.

- "If you want to go fast, go alone; if you want to go far, go together?"
- Not every "good" scientist wants to become a "good" software engineer, too!
- The Carpentries philosophy [40]

Where do we get our teachers from The community discussion brought about the need for a mixture of people, thereby the education of aspiring RSEs will involve people from close domain sciences or experienced RSEs and people that have respective additional skills to teach RSE competencies to the new generation. In that respect, this follows the carpentries model that offers certifications but is still open to non-certified trainers. We highlight and emphasize, that since a topic like RSE education, is constantly evolving, trainers strongly require the opportunity to and the recognition to educate themselves. Therefore our teachers will be sourced from the workplace but there will also be certified RSE Trainers. (FIXME: in classical university speak, these would be people who have done their habilitation on that topic, right?)

We propose to create common Infrastructures that can be utilized in this ongoing effort to professionalize the RSE education further, to easily share education resources across the country. (FIXME: DETAIL ME FURTHER!!!!)

Organization of teaching

Certificates With the ever-growing demand for RSEs in science it could be helpful for people to earn certificates for skills needed in certain RSE positions. This would possibly make hiring easier and could incentivise researchers to go through proper courses on these skills instead of learning on the go. For certain skills it would also be good for finding jobs outside academia, e.g. in industry where certain practices are already state-of-the-art. However, these certificates are only helpful if there is a certain level of standardization, which would require a central authority or collaboration between multiple stakeholders to decide on contents and allow participating institutions to issue these certificates. Additionally, it can be excluding capable applicants who already use these skills but never got a certificate for it.

The possible types of certificates can of course differ. The HPC skills and certification Appendix explores current efforts at creating a HPC certification program for both academic and industry RSEs. Course attendance sheets and digital tokens [72, 21, 48, 99] are another option (see Appendix: Micro-accreditation).

Having certificates provides finally a clear understanding of which tasks an RSE can perform and thereby helps defining the career path and the job description. A big demand for specialized RSEs will certainly come from the newly established RSE centers at research institutions that require skilled people to fill their vacant positions. And using the certificates, the demand can now be satisfied with people offering this skill.

Some exemplary skills for which courses are already held are version control tools like git, HPC topics like multithreading, MPI and GPU computations, FAIR principles.

A possible graduation path within the classical university structures We have put forward the idea that familiarity with research is a prerequisite for an RSE in order to be able to work effectively in the research space and in collaboration with researchers. In this particular example, we consider a path into RSE via a traditional university route involving Bachelors and Masters degree studies that include an RSE element. However, we recognise that there are other routes into an RSE career and these are increasing. For example, some RSEs come from an industry background, others may come through apprenticeship or similar programmes. In both cases, gaining knowledge of the research lifecycle and understanding the ways that researchers work towards solutions to research challenges is something that can be developed on-the-job alongside training opportunities and the chance to work directly with researchers. This leads naturally to the question, whether it is possible to become an RSE without a home research domain. With software being a core element of the research process in so many different domains, it is not helpful for everybody working in RSE to have a background in computing or software engineering. Indeed, we consider it much more useful if new graduates looking to work in the RSE space come from a wide range of different domains. Expertise, beyond software development skills, in another research domain can be an important element of an RSE team being able to support RSE projects in that domain. Assuming for a moment, that people have done their masters studies in a particular domain, e.g. from the natural sciences, and that we can assume that the lectures to that point contain a mixture of domain specific content and RSE specific content (A good starting point for an RSE in the natural sciences), then we come to the question of the topic of the masters thesis. In order for young RSEs to get their research experience we believe it is necessary that already in their master's thesis young RSE students are given computational research tasks that can be solved with the RSE specific skills of that domain. This gives them a Master's degree of a ${DOMAIN}-RSE that has learnt in their lectures a research domain specific part and a software engineering specific part, and enabled them to get a first dip into actual science in their master's thesis. Of course, the next question for their future is whether a master's degree enables them to really be effective parts of a research group. While we accept this is something of a generalisation, we argue that this is most likely not the case since undertaking a PhD provides a much more extensive set of research training and experience that can be vital for a researcher's future. Research environments different internationally but in many cases there are formal barriers in the research landscape that require a PhD (e.g. eligibility for funding). Hence a PhD is required to actively participate in science and why we argue the regular RSE should do a PhD that on one end combines knowledge of a research domain with software engineering heavy task such that both pillars are suitably covered and they were able to observe how research functions.

Specialised Master's Programs On the other hand, when pursuing a PhD, scientists are increasingly required to do RSE-type work as part of their research as data and computation are becoming part of research tasks in a huge range of fields. It is not uncommon for researchers to be faced with RSE topics for the first time, because it has not been part of their academic curricula. Many are faced with a steep learning curve that requires them to invest a huge amount of time to catch up. Naturally, many would only invest as much as necessary to get the job done regardless of whether the solution is sustainable or not. Support from RSEs is one way to resolve this challenge. Another would be to lay more effective foundations for future RSE work at a much earlier stage in undergraduate/postgraduate studies. We see scope for establishing dedicated RSE Master's programmes which specialise in developing RSE skills and practices. Some universities already offer dedicated master's programs in some domains. Examples would be Computational Sciences in Engineering (CSE) or Bioinformatics (see Appendix: Bioinformatics skills and certification). Where appropriate similar programs should also be established in other domains.

Required Next steps

Implementation Strategies

- Ideally over time scientific software engineering becomes part of the curricula at universities.

Academic Considerations

- Awareness of existing teaching programs
- "Branded" Add-on courses
- External Institutions provide resources
- fully recognized in the academic system. Students get ECTS points.
- Bachelor/Master specializations

Broader Considerations

- Instilling more respect for people that want to educate themselves for digital competencies
- Outreach to people that now have the feeling that they require this training.

Conclusion

We have indentified the RSE has an individual that contributes to research teams with their knowledge about digital tools. Then we have defined generic core competencies from the pillars of Software Engineering, Research and Team processes. We fleshed them out with some possible specializations of RSEs. Given the competencies and a demand(FIXME: Do the calculation) in the research landscape for them we moved on to define who the teachers are for this new field. We closed with a discussion of possible structures and organization forms that educate new generations of RSEs in more structured programs than what is available today(FIXME: this is currently aspirational). Therefore this closes the gap, that the research landscape requires RSEs, but there are no structures where these persons are educated, by detailing the career path that a young person might want to take to become an RSE.(FIXME: also aspirational…)

Appendix

An applied example curriculum

An example of a possible career path

- We can follow Kim, who has been the protagonist of the original deRSE Paper.

HPC skills and certification

As an area that generally requires a range of advanced skills, High Performance Computing (HPC) is one field where there is ongoing work to identify relevant sets of skills for HPC practitioners and potential paths to develop these skills. The HPC Certification Forum [147] has developed a competence standard for HPC that defines a range of skills and how they are related in the context of a skill tree [82, 83]. This competence standard is currently being built upon by the CASTIEL 2 [46] project in collaboration with initiatives funded by the European High Performance Computing Joint Undertaking (EuroHPC JU) to create a framework for HPC certification [56].

Also looking at pathways and how different skills are related, the UNIVERSE-HPC project [152], funded under the UK's ExCALIBUR research programme [47], is looking to understand and develop training pathways to support the development of specialist skills in the HPC and exascale domains. The project is gathering open source training materials to develop curricula that support the training pathways that are underpinned by high-quality training materials.

Bioinformatics skills and certification

Bioinformatics is another field that actively works on developing skill trees. The Bioinformatics Core Competencies [103, 158, 159], the BioExcel competency framework [98], the PerMedCoE competency framework [95], the Research Data Management and Data Stewardship competence framework [38] and the ELIXIR Data Stewardship Competency Framework for Life Sciences [128] are examples of grassroot

efforts aiming at defining the set of skills of various bioinformatics specialties, one of them as a taxonomy [103]. These frameworks eventually converged into the EMBL-EBI Competency Hub [41, 94], where typical RSE and bioinformatician profiles at different levels of seniority can be queried (e.g. Junior RSE, Senior Computational Chemist) and compared against one another (e.g. Junior vs. Senior RSE).

Competencies can be divided into more fine-grained building blocks: knowledge, skills and abilities (KSAs). They can be organized in a taxonomy, and are also transferable, i.e. a KSA can be a prerequisite to multiple competencies. The Mastery Rubric for Bioinformatics [150] and the ELIXIR Data Stewardship Competency Framework for Life Sciences [128] are examples of KSA frameworks for bioinformatics curricula.

The Curriculum Task Force of the International Society for Computational Biology (ISCB) curates a database of degrees and certificates in bioinformatics [75, 103]. The database includes Bachelor and Master's degree programs and specializations, Ph.D. programs, and certificates from graduate schools.

Micro-accreditation

RSEs are subject to life-long learning. As such, digital certificates and learner badges [72] are one possibility for experienced RSEs to showcase that they posess a certain technical skill. For example, the Software Carpentries minted digital badges in 2012 [166, 163] as a form of institutional accreditation. Despite initial plans to create skill-specific badges [164], learner badges were ultimately abandoned in 2013 [167] and recycled as participation certificates [141]. Instructor badges were introduced instead [167], which are now mandatory tokens to lead a Carpentry workshop or vote in council elections [145]. Ireland's Professional Development Framework [39] provides accreditation to higher education teachers who successfully complete training on the National Forum's Open Courses via digital badges [108]. The Extreme Science and Engineering Discovery Environment organization delivers badges to incentivise participation in HPC training events [79, 126]. IBM delivers badges to promote continuous learning and provide micro-credentials to its staff and customers [87, 86] There are 1360 badges as of July 2023 [71] and 1 million badges were issued as of July 2018 [34]; they are recognized by a few academic institutions [106, 105, 14] and in some cases are convertible to graduate credit [87].

The Open Source Software Security Mobilization Plan [148] is proposing that code repositories and recruiting sites work on recognising digital badges certifying RSE skills in secure software development. Some code repositories already feature an infrastructure to automatically issue digital tokens, from personal badges measuring contributions [129, 125] or community work (e.g. outreach efforts, workshop attendance and package maintenance in Fedora [146]), to project-specific badges [109, 151, 90] that illustrate best practices, such as high code coverage or code quality, or signal commitment to diversity, equity, and inclusion [22, 23].

FIXME or remove me:

- an argument could be made for having less metrics (GitHub allows users to hide their badges)
- which institution would create RSE-badges? how would this institution drive adoption of RSE-badges?
- downside of fine-grained badges: number of badges can grow quickly
- badges can also fade away, i.e. have diminishing relevance or usefulness over time (e.g. old workshop participation badge, or technologies that have since become less relevant in the RSE field)
- reviews [177, 63, 1, 93] and case studies [102, 15, 70] on digital badges

References

[1] Kamrul Ahsan et al. "Implementation of micro-credentials in higher education: A systematic literature review". In: *Education and Information Technologies* (Mar. 2023). DOI: 10.1007/s10639-023-11739-z.

[2] Ben Akoh. "Determinants of Mobile Learning in Indigenous/Cultural Contexts: The Phenomenon in Canadian First Nations". In: *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing* (Dublin, Ireland, July 3–6, 2017). Ed. by Arthur Tatnall and Mary Webb. Vol. 515. IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer International Publishing, 2017, pp. 24–34. ISBN: 978-3-319-74310-3. DOI: 10.1007/978-3-319-74310-3_4.

[3] José Luis Ambite et al. "BD2K Training Coordinating Center's ERuDIte: The Educational Resource Discovery Index for Data Science". In: *IEEE Transactions on Emerging Topics in Computing* 9.1 (Jan. 2021), pp. 316–328. DOI: 10.1109/tetc.2019.2903466.

[4] Franziska Appel and Axel Loewe. *Forschungssoftware - Nachhaltige Entwicklung und Unterstützung*. IAMO Policy Brief 42. Halle (Saale), Saxony-Anhalt, Germany, 2021. URL: https://hdl.handle.net/10419/240937.

[5] Julen Astigarraga and Verónica Cruz-Alonso. "¡Se puede entender cómo funcionan Git y GitHub!" In: *Ecosistemas* 31.1 (Apr. 2022), p. 2332. ISSN: 1697-2473. DOI: 10.7818/ecos.2332.

[6] Finn Bacall et al. "Making Bioinformatics Training Events and Material More Discoverable Using TeSS, the ELIXIR Training Portal". In: *Current Protocols* 3.2 (Feb. 2023). DOI: 10.1002/cpz1.682.

[7] Stevie A. Bain, Thomas R. Meagher, and Daniel Barker. "Design, delivery and evaluation of a bioinformatics education workshop for 13-16-year-olds". In: *Journal of Biological Education* 56.5 (Dec. 2020), pp. 570–580. DOI: 10.1080/00219266.2020.1858932.

[8] James Baker et al. "Library Carpentry: Software Skills Training for Library Professionals". In: *Liber Quarterly* 26.3 (Nov. 2016), pp. 141–162. ISSN: 2213-056X. DOI: 10.18352/lq.10176.

[9] Gabriel Balaban et al. "Ten simple rules for quick and dirty scientific programming". In: *PLOS Computational Biology* 17.3 (Mar. 2021). Ed. by Scott Markel, e1008549. DOI: 10.1371/journal.pcbi.1008549.

[10] Michelle Barker et al. "Introducing the FAIR Principles for research software". In: *Scientific Data* 9.1 (Oct. 2022), p. 622. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01710-x. URL: https://doi.org/10.1038/s41597-022-01710-x.

[11] Susan M. Baxter et al. "Scientific Software Development Is Not an Oxymoron". In: *PLoS Computational Biology* 2.9 (2006), e87. DOI: 10.1371/journal.pcbi.0020087.

[12] Niall Beard et al. "TeSS: a platform for discovering life-science training opportunities". In: *Bioinformatics* 36.10 (Feb. 2020). Ed. by Jonathan Wren, pp. 3290–3291. DOI: 10.1093/bioinformatics/btaa047.

[13] Matthew D. Beckman et al. "Implementing Version Control With Git and GitHub as a Learning Objective in Statistics and Data Science Courses". In: *Journal of Statistics and Data Science Education* 29.sup1 (Jan. 2021), S132–S144. DOI: 10.1080/10691898.2020.1848485.

[14] Bluefield State. *Bluefield State College Now Offering IBM Badge Certification*. URL: https://bluefieldstate.edu/community/news-and-events/bluefield-state-college-now-offering-ibm-badge-certification (visited on 07/03/2023).

[15] Oriol Borrás-Gené. "Use of digital badges for training in digital skills within higher education". In: *2018 International Symposium on Computers in Education (SIIE)* (Jerez, Spain, Sept. 19–21, 2018). IEEE, Sept. 2018. ISBN: 978-1-5386-6505-3. DOI: 10.1109/siie.2018.8586734.

[16] Sophie Brouat et al. "What unique knowledge and experiences do healthcare professionals have working in clinical informatics?" In: *Informatics in Medicine Unlocked* 32 (2022), p. 101014. ISSN: 2352-9148. DOI: 10.1016/j.imu.2022.101014.

[17] Jeffrey C. Carver and Ian A. Cosden. *INnovative Training Enabled by a Research Software Engineering Community of Trainers (INTERSECT)*. Zenodo. Aug. 2020. DOI: 10.5281/zenodo.4281287.

[18] CASTIEL and EuroCC Network. "Best Practice Guide: How to Find New Attendees for Training Courses". In: *Training Best Practice Seminar (CASTIEL)* (Jan. 20, 2022). EuroCC, Jan. 2022. URL: https://www.eurocc-access.eu/wp-content/uploads/2022/04/20220401_Best_Practice_Guide-Training_Best_Practice_Seminar_final.pdf.

[19] Oihane Cereceda and Danielle E. A. Quinn. "A graduate student perspective on overcoming barriers to interacting with open-source software". In: *FACETS* 5.1 (Jan. 2020). Ed. by Debra Clendinneng, pp. 289–303. DOI: 10.1139/facets-2019-0020.

[20] Mine Çetinkaya-Rundel and Colin Rundel. "Infrastructure and Tools for Teaching Computing Throughout the Statistical Curriculum". In: *The American Statistician* 72.1 (Jan. 2018), pp. 58–65. DOI: 10.1080/00031305.2017.1397549.

[21] Borhene Chakroun and James Keevy. *Digital credentialing: Implications for the recognition of learning across borders*. Tech. rep. ED-2018/WS/29. Paris, France: UNESCO, 2018. DOI: 10.54675/sabo8911.

[22] CHAOSS. *Overview of D&I Badging*. CHAOSS Community Handbook. Nov. 2020. URL: https://handbook.chaoss.community/badging/ (visited on 06/23/2023).

[23] Demetris Cheatham. *Announcing the All In CHAOSS DEI Badging pilot initiative*. GitHub Blog. June 2023. URL: https://github.blog/2023-06-07-announcing-the-all-in-chaoss-dei-badging-pilot-initiative/ (visited on 06/23/2023).

[24] Neil Chue Hong. "Minimal information for reusable scientific software". In: *2nd Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE2)* (New Orleans, Louisiana, USA, Nov. 16, 2014). figshare, July 2014. DOI: 10.6084/m9.figshare.1112528.

[25] Neil Chue Hong, Brian Hole, and Samuel Moore. "Software Papers: improving the reusability and sustainability of scientific software". In: *1st Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE1)* (Denver, Colorado, USA, Nov. 17, 2013). figshare, Sept. 2013. DOI: 10.6084/m9.figshare.795303.

[26] Sarah Clarke and Candace Norton. *Current Librarian RDM Training*. Sept. 2019. DOI: 10.17605/OSF.IO/8SJR5.

[27] Mélanie Clément-Fontaine et al. *Encouraging a wider usage of software derived from research*. Research Report. Comité pour la science ouverte, Nov. 2019. DOI: 10.52949/4. Translation of *Note d'opportunité sur la valorisation des logiciels issus de la recherche*. Research Report. Comité pour la science ouverte, Nov. 2019. DOI: 10.52949/17.

[28] CodeRefinery. *Coderefinery*. URL: https://coderefinery.org (visited on 06/16/2023).

[29] Jez Cope and James Baker. "Library Carpentry: Software Skills Training for Library Professionals". In: *International Journal of Digital Curation* 12.2 (May 2018), pp. 266–273. ISSN: 1746-8256. DOI: 10.2218/ijdc.v12i2.576.

[30] Ian A. Cosden, Kenton McHenry, and Daniel S. Katz. "Research Software Engineers: Career Entry Points and Training Gaps". In: *Computing in Science & Engineering* 24.6 (Nov. 2022), pp. 14–21. DOI: 10.1109/mcse.2023.3258630.

[31] Tom Crick, Benjamin A. Hall, and Samin Ishtiaq. "Reproducibility in Research: Systems, Infrastructure, Culture". In: *Journal of Open Research Software* 5.1 (Nov. 2017), p. 32. DOI: 10.5334/jors.73.

[32] Stephen Crouch et al. "The Software Sustainability Institute: Changing Research Software Attitudes and Practices". In: *Computing in Science & Engineering* 15.6 (Nov. 2013), pp. 74–80. ISSN: 1558-366X. DOI: 10.1109/mcse.2013.133.

[33] Michael R. Crusoe and C. Titus Brown. "Walking the Talk: Adopting and Adapting Sustainable Scientific Software Development processes in a Small Biology Lab". In: *Journal of Open Research Software* 4.1 (Nov. 2016), e44. DOI: 10.5334/jors.35.

[34] James Daniels. *IBM issues One Millionth badge*. IBM Training and Skills Blog. July 2018. URL: https://web.archive.org/web/20201121104150/https://www.ibm.com/blogs/ibm-training/ibm-issues-one-millionth-badge/ (visited on 07/03/2023).

[35] *Data Carpentry*. URL: https://datacarpentry.org/ (visited on 07/25/2023).

[36] Alan Davies, Julia Mueller, and Georgina Moulton. "Core competencies for clinical informaticians: A systematic review". In: *International Journal of Medical Informatics* 141 (Sept. 2020), p. 104237. DOI: 10.1016/j.ijmedinf.2020.104237.

[37] Alan Davies et al. "Creation of a core competency framework for clinical informatics: From genesis to maintaining relevance". In: *International Journal of Medical Informatics* 168 (Dec. 2022), p. 104905. DOI: 10.1016/j.ijmedinf.2022.104905.

[38] Yuri Demchenko and Lennart Stoy. "Research Data Management and Data Stewardship Competences in University Curriculum". In: *Proceedings of the 2001 IEEE Global Engineering Education Conference (EDUCON)* (Vienna, Austria, Apr. 21–23, 2021). Ed. by Thomas Klinger, Christian Kollmitzer, and Andreas Pester. Red Hook, New York, USA: IEEE, Apr. 2021. ISBN: 978-1-7281-8478-4. DOI: 10.1109/educon46332.2021.9453956.

[39] Roisin Donnelly and Terry Maguire. "Ireland's Higher Education Teachers Have a National Professional Development Framework, Now What?" In: *Transforming our World Through Design, Diversity and Education*. Ed. by Gerald Craddock et al. Vol. 256. Studies in Health Technology and Informatics. IOS Press, 2018, pp. 655–666. ISBN: 978-1-61499-923-2. DOI: 10.3233/978-1-61499-923-2-655.

[40] Beth M. Duckles. *Value of Software Carpentry to Instructors Report*. Tech. rep. Jan. 2016. URL: https://software-carpentry.org/files/bib/duckles-instructor-engagement-2016.pdf.

[41] EMBL-EBI. *Competency Hub*. URL: https://competency.ebi.ac.uk/ (visited on 07/20/2023).

[42] ENCCS. *ENCCS Instructor Training*. Training Material to Train the Trainer. EuroCC National Competence Center Sweden, Nov. 2022. URL: https://www.eurocc-access.eu/wp-content/uploads/2022/12/EuroCC_NCC_Sweden_instructor_training_workshop.pdf.

[43] ENCCS. *ENCCS Instructor Training*. URL: https://enccs.github.io/instructor-training/ (visited on 06/16/2023).

[44] ENCCS. *ENCCS lessons*. URL: https://enccs.github.io/instructor-training/enccs-lessons/ (visited on 06/16/2023).
[45] EuroCC. *EuroCC Training*. URL: https://www.eurocc-access.eu/services/training/ (visited on 06/16/2023).
[46] EuroHPC Joint Undertaking. *EuroCC 2 and CASTIEL 2: Promoting HPC to boost digital skills, jobs and industrial competitiveness in Europe*. URL: https://eurohpc-ju.europa.eu/eurocc-2-and-castiel-2-promoting-hpc-boost-digital-skills-jobs-and-industrial-competitiveness-europe-2023-02-03_en (visited on 06/16/2023).
[47] ExCALIBUR. *Exascale Computing ALgorithms & Infrastructures Benefiting UK Research*. URL: https://excalibur.ac.uk (visited on 06/16/2023).
[48] Joseph Fanfarelli, Stephanie Vie, and Rudy McDaniel. "Understanding digital badges through feedback, reward, and narrative". In: *Communication Design Quarterly* 3.3 (June 2015), pp. 56–60. DOI: 10.1145/2792989.2792998.
[49] Lisa Federer, Sarah C. Clarke, and Maryam Zaringhalam. *Developing the Librarian Workforce for Data Science and Open Science*. Workshop report. National Library of Medicine, Jan. 2020. DOI: 10.31219/osf.io/uycax.
[50] J. Fehr et al. "Sustainable Research Software Hand-Over". In: *Journal of Open Research Software* 9.1 (Apr. 2021), p. 5. DOI: 10.5334/jors.307.
[51] Karl Fogel. *Producing Open Source Software. How to Run a Successful Free Software Project*. 1st ed. O'Reilly Media, Oct. 2005. ISBN: 978-0-596-00759-1. URL: https://www.oreilly.com/library/view/producing-open-source/0596007590/.
[52] Karl Fogel. *Producing Open Source Software. How to Run a Successful Free Software Project*. 2nd ed. https://www.producingoss.com, Jan. 2017.
[53] David Form and Fran Lewitter. "Ten Simple Rules for Teaching Bioinformatics at the High School Level". In: *PLoS Computational Biology* 7.10 (Oct. 2011), e1002243. DOI: 10.1371/journal.pcbi.1002243.
[54] Leyla Garcia et al. "Ten simple rules for making training materials FAIR". In: *PLOS Computational Biology* 16.5 (May 2020). Ed. by Scott Markel, e1007854. DOI: 10.1371/journal.pcbi.1007854.
[55] Reed M. Gardner et al. "Core Content for the Subspecialty of Clinical Informatics". In: *Journal of the American Medical Informatics Association* 16.2 (Mar. 2009), pp. 153–157. DOI: 10.1197/jamia.m3045.
[56] Governing Board of the EuroHPC Joint Undertaking. *Amending the Joint Undertaking's Work Programme and Budget for the year 2023 (Work Programme and Budget Amendment no. 1)*. EuroHPC JU Decision No 03/2023. EuroHPC Joint Undertaking, Mar. 2023. URL: https://eurohpc-ju.europa.eu/system/files/2023-03/Decision%203.2023.-%201st%20Amendment%20WP%202023.pdf.
[57] Alan Grossfield. "How to be a Good Member of a Scientific Software Community [Article v1.0]". In: *Living Journal of Computational Molecular Science* 3.1 (Jan. 2022), p. 1473. DOI: 10.33011/livecoms.3.1.1473.
[58] Morane Gruenpeter et al. "Defining Research Software: a controversial discussion". In: (Sept. 2021). DOI: 10.5281/zenodo.5504016.
[59] Carla Guillen and Carmen Navarrete. *Introduction to C++*. URL: https://doku.lrz.de/2022-09-21-introduction-to-c++-hcpb2s22-11497182.html (visited on 06/16/2023).
[60] Kim T. Gurwitz et al. "A framework to assess the quality and impact of bioinformatics training across ELIXIR". In: *PLOS Computational Biology* 16.7 (July 2020). Ed. by Francis Ouellette, e1007976. DOI: 10.1371/journal.pcbi.1007976.
[61] Mario Haim. "Gütekriterien und Handlungsempfehlungen für die Entwicklung von Forschungssoftware in der Kommunikations- und Medienwissenschaft". In: *Medien & Kommunikationswissenschaft* 69.1 (2021), pp. 65–79. DOI: 10.5771/1615-634x-2021-1-65.
[62] Jo Erskine Hannay et al. "How do scientists develop and use scientific software?" In: *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering* (Vancouver, British Columbia, Canada, May 23–23, 2009). IEEE, May 2009. ISBN: 978-1-4244-3737-5. DOI: 10.1109/secse.2009.5069155.
[63] Anna Hansch, Christopher Newman, and Thomas Schildhauer. "Fostering Engagement with Gamification: Review of Current Practices on Online Learning Platforms". In: *SSRN Electronic Journal* (Nov. 2015). DOI: 10.2139/ssrn.2694736.
[64] Helmholtz. *Helmholtz AI*. URL: https://www.helmholtz.ai/ (visited on 06/16/2023).

[65] Helmholtz. *Helmholtz Federated IT Services (HIFIS)*. URL: https://hifis.net (visited on 06/16/2023).
[66] Helmholtz. *Helmholtz Imaging*. URL: https://helmholtz-imaging.de/ (visited on 06/16/2023).
[67] Helmholtz. *Helmholtz Information & Data Science Academy (HIDA)*. URL: https://www.helmholtz-hida.de (visited on 06/16/2023).
[68] Helmholtz. *Helmholtz Metadata Collaboration (HMC)*. URL: https://helmholtz-metadaten.de (visited on 06/16/2023).
[69] Simon Hettrick. *A not-so-brief history of Research Software Engineers*. Software Susitainability Institute. Aug. 2016. URL: https://www.software.ac.uk/blog/2016-08-17-not-so-brief-history-research-software-engineers-0 (visited on 06/16/2023).
[70] Ross Higashi and Christian D. Schunn. "Perceived relevance of digital badges predicts longitudinal change in program engagement". In: *Journal of Educational Psychology* 112.5 (July 2020), pp. 1020–1041. DOI: 10.1037/edu0000401.
[71] IBM. *IBM Training / Badges*. URL: https://www.ibm.com/training/search?query=*&trainingType=Badge (visited on 07/03/2023).
[72] Dirk Ifenthaler, Nicole Bellin-Mularski, and Dana-Kristin Mah, eds. *Foundation of Digital Badges and Micro-Credentials. Demonstrating and Recognizing Knowledge and Competencies*. Cham, Switzerland: Springer International Publishing, June 2016. ISBN: 978-3-319-15425-1. DOI: 10.1007/978-3-319-15425-1.
[73] Klaus Iglberger. *Modern C++ Software Design*. URL: https://doku.lrz.de/2022-10-26-modern-c++-software-design-hcpa1w22-11497188.html (visited on 06/16/2023).
[74] Institute for Innovation Research and Management. *DigiFlex - Digital Flexibilization Program for an Individualized Freshman Year*. URL: https://www.ifi-ge.de/en/projects/current-projects/translate-to-englisch-digiflex/ (visited on 07/03/2023).
[75] International Society for Computational Biology. *Degree & Certificate Programs in Bioinformatics*. URL: https://www.iscb.org/iscb-degree-certificate-programs (visited on 07/20/2023).
[76] INTERSECT. *Existing Research Software Engineering Training Material*. 2023. URL: https://intersect-training.org/training-links/ (visited on 07/12/2023).
[77] Damien Irving et al. *Research Software Engineering with Python. Building software that makes research possible*. CRC Press, 2021. ISBN: 978-1-003-14348-2. DOI: 10.1201/9781003143482.
[78] Mike Jackson, Steve Crouch, and Rob Baxter. *Software Evaluation: Criteria-Based Assessment*. Software Evaluation Guide. Software Sustainability Institute, Nov. 2011. URL: https://software.ac.uk/sites/default/files/SSI-SoftwareEvaluationCriteria.pdf.
[79] Sandra Kappes and Vincent C. Betro. "Using Mozilla badges to certify XSEDE users and promote training". In: *Proceedings of the 2015 XSEDE Conference on Scientific Advancements Enabled by Enhanced Cyberinfrastructure - XSEDE '15* (St. Louis, Missouri, USA, July 26–30, 2015). New York, New York, USA: ACM Press, July 2015. ISBN: 978-1-4503-3720-5. DOI: 10.1145/2792745.2792759.
[80] Matthias Katerbow and Georg Feulner. *Recommendations on the Development, Use and Provision of Research Software*. Zenodo. 2018. DOI: 10.5281/zenodo.1172988.
[81] "SusTrainable: Promoting Sustainability as a Fundamental Driver in Software Development Training and Education". Revised lecture notes. In: (Radboud University Nijmegen, Nijmegen, The Netherlands, Nov. 1–5, 2021). Ed. by Pieter Koopman, Mart Lubbers, and João Paulo Fernandes. arXiv, Apr. 2022. DOI: 10.48550/arXiv.2204.13993.
[82] Julian Kunkel et al. "The HPC Certification Forum: Toward a Globally Acknowledged HPC Certification". In: *Computing in Science & Engineering* 22.4 (2020), pp. 110–114. ISSN: 1521-9615. DOI: 10.1109/MCSE.2020.2996073.
[83] Julian Martin Kunkel et al. "One Year HPC Certification Forum in Retrospective". In: *The Journal of Computational Science Education* 11.1 (Jan. 2020), pp. 29–35. ISSN: 2153-4136. DOI: 10.22369/issn.2153-4136/11/1/6.
[84] Iva Laginja et al. "Connecting the astronomical testbed community - the CAOTIC project: optimized teaching methods for software version control concepts". In: *Adaptive Optics Systems VIII* (Montréal, Québec, Canada, July 17–23, 2022). Ed. by Laura Schreiber, Dirk Schmidt, and Elise Vernet. Vol. 12185. International Society for Optics and Photonics. Bellingham, Washington, USA: SPIE, Aug. 2022, 121853A. ISBN: 9781510653528. DOI: 10.1117/12.2629483.
[85] Carl Landwehr et al. "Software Systems Engineering programmes a capability approach". In: *Journal of Systems and Software* 125 (2017), pp. 354–364. ISSN: 0164-1212. DOI: 10.1016/j.jss.2016.12.016.

[86] David Leaser. *IBM Digital Badge Program Overview*. slideshare. July 2019. URL: https://www.slideshare.net/DavidLeaser/ibm-digital-badge-program-overview.

[87] David Leaser, Kemi Jona, and Sean Gallagher. "Connecting Workplace Learning and Academic Credentials via Digital Badges". In: *New Directions for Community Colleges* 2020.189 (Feb. 2020), pp. 39–51. ISSN: 1536-0733. DOI: 10.1002/cc.20396.

[88] Michael Lee. "Happy Belly Bioinformatics: an open-source resource dedicated to helping biologists utilize bioinformatics". In: *Journal of Open Source Education* 2.19 (Sept. 2019), p. 53. DOI: 10.21105/jose.00053.

[89] Victor (Kwangchun) Lee. *Korean Translation of Software Carpentry - version 5.2*. Software Carpentry Blog. Apr. 2015. URL: https://software-carpentry.org/blog/2015/04/korean-translation-of-software-carpentry-ver-5-2.html (visited on 06/19/2023).

[90] Damien Legay, Alexandre Decan, and Tom Mens. "On the Usage of Badges in Open Source Packages on GitHub". In: *Proceedings of the 18th Edition of the Belgian-Netherlands Software Evolution Symposium (BENEVOL 2019)* (Brussels, Belgium, Nov. 28–29, 2019). Ed. by Dario Di Nucci and Coen De Roover. CEUR Workshop Proceedings. CEUR, May 2020. URL: https://ceur-ws.org/Vol-2605/9.pdf.

[91] Felipe da Veiga Leprevost et al. "On best practices in the development of bioinformatics software". In: *Frontiers in Genetics* 5 (July 2014). DOI: 10.3389/fgene.2014.00199.

[92] *Library Carpentry*. URL: https://librarycarpentry.org/index.html (visited on 07/25/2023).

[93] Tharindu R. Liyanagunawardena, Sandra Scalzavara, and Shirley A. Williams. "Open Badges: A Systematic Review of Peer-Reviewed Published Literature (2011-2015)". In: *European Journal of Open, Distance and E-Learning* 20.2 (Dec. 2017), pp. 1–16. ISSN: 1027-5207. DOI: 10.1515/eurodl-2017-0013.

[94] Marta Lloret-Llinares and Daniel Thomas Lopez. *The EMBL-EBI Competency Hub, a tool to support training and professional development*. FEBS Network. Oct. 2022. URL: https://network.febs.org/posts/the-embl-ebi-competency-hub-a-tool-to-support-training-and-professional-development (visited on 07/20/2023).

[95] Marta Lloret-Llinares et al. "The PerMedCoE competency framework to guide the training programme". In: 29th Conference on Intelligent Systems for Molecular Biology and the 20th European Conference on Computational Biology (Virtual Event, July 25–30, 2021). July 2021. URL: https://permedcoe.eu/publication/the-permedcoe-competency-framework-to-guide-the-training-programme/.

[96] Juan M. *Stack Overflow en Español has Graduated!* The Overflow. May 2017. URL: https://stackoverflow.blog/2017/05/20/stack-overflow-en-espanol-graduated/ (visited on 06/19/2023).

[97] Marlie MacLean and Colin Miles. "Swift action needed to close the skills gap in bioinformatics". In: *Nature* 401.6748 (Sept. 1999), pp. 10–10. DOI: 10.1038/43269.

[98] Vera Matser. *BioExcel Deliverable 4.2 – Competency framework, mapping to current training & initial training plan*. Zenodo. 2016. DOI: 10.5281/zenodo.264231.

[99] Rudy McDaniel and Joseph Fanfarelli. "Building Better Digital Badges". In: *Simulation & Gaming* 47.1 (Feb. 2016), pp. 73–102. DOI: 10.1177/1046878115627138.

[100] Clare McInerney, Anna-Lena Lamprecht, and Tiziana Margaria. "Computing Camps for Girls – A First-Time Experience at the University of Limerick". In: *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing* (Dublin, Ireland, July 3–6, 2017). Ed. by Arthur Tatnall and Mary Webb. Vol. 515. IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer International Publishing, 2017, pp. 494–505. ISBN: 978-3-319-74310-3. DOI: 10.1007/978-3-319-74310-3_50.

[101] Sarah L. Morgan et al. "The ELIXIR-EXCELERATE Train-the-Trainer pilot programme: empower researchers to deliver high-quality training [version 1; peer review: 2 approved]". In: *F1000Research* 6 (Aug. 2017), p. 1557. DOI: 10.12688/f1000research.12332.1.

[102] Bradley J. Morris et al. "Comparing badges and learning goals in low- and high-stakes learning contexts". In: *Journal of Computing in Higher Education* 31.3 (May 2019), pp. 573–603. DOI: 10.1007/s12528-019-09228-9.

[103] Nicola Mulder et al. "The development and application of bioinformatics core competencies to improve bioinformatics training and education". In: *PLOS Computational Biology* 14.2 (Feb. 2018). Ed. by Olga G. Troyanskaya, e1005772. DOI: 10.1371/journal.pcbi.1005772.

[104] National Academies of Sciences, Engineering, and Medicine. *Transforming Trajectories for Women of Color in Tech*. Ed. by Evelynn Hammonds, Valerie Taylor, and Rebekah Hutton. Washington, D.C., USA: The National Academies Press, 2022. ISBN: 978-0-309-26908-7. DOI: 10.17226/26345.

[105] North Carolina Central University. *NCCU SLIS Partners with IBM Skills Academy*. URL: https://www.nccu.edu/slis/nccu-slis-partners-ibm-skills-academy (visited on 07/03/2023).

[106] Northeastern University. *Northeastern University and IBM partnership first to turn digital badges into academic credentials for learners worldwide*. News@Northeastern. Sept. 2017. URL: https://news.northeastern.edu/2017/09/25/northeastern-university-and-ibm-partnership-first-to-turn-digital-badges-into-academic-credentials-for-learners-worldwide/ (visited on 07/03/2023).

[107] Alan O'Cais and Peter Steinbach. "Expanding user communities with HPC Carpentry". In: *The Journal of Computational Science Education* 11.1 (Jan. 2020), pp. 21–25. ISSN: 2153-4136. DOI: 10.22369/issn.2153-4136/11/1/4.

[108] Open Courses. *About Open Badges*. URL: https://opencourses.ie/about-open-badges/ (visited on 07/03/2023).

[109] Open Source Security Foundation. *OpenSSF Best Practices Badge Program*. 2021. URL: https://bestpractices.coreinfrastructure.org/en (visited on 07/12/2023).

[110] Ana Oprescu et al. "Energy-driven software engineering". In: *SusTrainable: Promoting Sustainability as a Fundamental Driver in Software Development Training and Education*. Revised lecture notes (Radboud University Nijmegen, Nijmegen, The Netherlands, Nov. 1–5, 2021). Ed. by Pieter Koopman, Mart Lubbers, and João Paulo Fernandes. arXiv, Apr. 2022, pp. 27–35. DOI: 10.48550/arXiv.2204.13993.

[111] Rui Pereira et al. "Ranking programming languages by energy efficiency". In: *Science of Computer Programming* 205 (May 2021), p. 102609. DOI: 10.1016/j.scico.2021.102609.

[112] Pavel Pevzner and Ron Shamir. "Computing Has Changed Biology—Biology Education Must Catch Up". In: *Science* 325.5940 (July 2009), pp. 541–542. DOI: 10.1126/science.1173876.

[113] Gustavo Pinto and Fernando Castor. "Energy Efficiency: A New Concern for Application Software Developers". In: *Communications of the ACM* 60.12 (Nov. 2017), pp. 68–75. ISSN: 1557-7317. DOI: 10.1145/3154384.

[114] Ben Popper. *Stack Overflow in Portugese: now with less beta!* The Overflow. June 2015. URL: https://stackoverflow.blog/2015/06/04/stack-overflow-in-portuguese-now-with-less-beta/ (visited on 06/19/2023).

[115] Ben Popper. *Welcome, Nicolas Chabanovsky and Stack Overflow in Russian!* The Overflow. June 2015. URL: https://stackoverflow.blog/2015/06/11/welcome-nicolas-chabanovsky-and-stack-overflow-in-russian/ (visited on 06/19/2023).

[116] PRACE. *Partnership for Advanced Computing in Europe*. URL: https://prace-ri.eu/ (visited on 06/16/2023).

[117] Andreas Prlić and James B. Procter. "Ten Simple Rules for the Open Development of Scientific Software". In: *PLoS Computational Biology* 8.12 (Dec. 2012), e1002802. DOI: 10.1371/journal.pcbi.1002802.

[118] Alex Reinhart and Christopher R. Genovese. "Expanding the Scope of Statistical Computing: Training Statisticians to Be Software Engineers". In: *Journal of Statistics and Data Science Education* 29.sup1 (Jan. 2021), S7–S15. DOI: 10.1080/10691898.2020.1845109.

[119] *ReproHack Hub: Building Communities of Practice in Reproducibility*. 2023. URL: https://www.reprohack.org/ (visited on 07/25/2023).

[120] K. V. Roberts. "The publication of scientific Fortran programs". In: *Computer Physics Communications* 1.1 (July 1969), pp. 1–9. DOI: 10.1016/0010-4655(69)90011-3.

[121] RSE Competencies Toolkit team. *RSE Competencies Toolkit*. GitHub. 2023. URL: https://github.com/RSEToolkit/rse-competencies-toolkit.

[122] US-RSE Training and Education Working Group. *RSE Training Resources*. URL: https://us-rse.org/resources/training/ (visited on 07/12/2023).

[123] US-RSE Training and Education Working Group. *Skills Used by RSEs*. URL: https://us-rse.org/resources/training/skills/ (visited on 07/12/2023).

[124] RWTH Aachen University. *Algorithmic Battle*. URL: https://tcs.rwth-aachen.de/lehre/algobattle/WS2023/ (visited on 07/03/2023).

[125] Ryota Sakamoto. *GitHub Profile Trophy*. URL: https://github.com/ryo-ma/github-profile-trophy (visited on 06/14/2023).

[126] Jeff Sale. "Badges for Visualization Micro-Certification". In: *Blue Waters Webinars* (National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, Illinois, USA, Aug. 2, 2017). Blue Waters, Aug. 2017. URL: https://bluewaters.ncsa.illinois.edu/liferay-content/document-library/BW%20Webinars/Slides/Badges%20for%20Visualization%20Micro-Certification%20by%20Jeff%20Sale.pdf.

[127] Tobias Schlauch, Michael Meinel, and Carina Haupt. *DLR Software Engineering Guidelines*. Tech. rep. Aug. 2018. URL: https://elib.dlr.de/121502/.
[128] Salome Scholtens et al. *Final report: Towards FAIR data steward as profession for the lifesciences. Report of a ZonMw funded collaborative approach built on existing expertise*. Zenodo. Oct. 2019. DOI: 10.5281/zenodo.3471707.
[129] Schweinepriester. *GitHub Profile Achievements*. URL: https://github.com/Schweinepriester/github-profile-achievements (visited on 06/15/2023).
[130] Judith Segal. "Some challenges facing software engineers developing software for scientists". In: *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering* (Vancouver, British Columbia, Canada, May 23–23, 2009). IEEE, May 2009. ISBN: 978-1-4244-3737-5. DOI: 10.1109/secse.2009.5069156.
[131] Chris Shelton. "How Can We Make Computing Lessons More Inclusive?" In: *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing* (Dublin, Ireland, July 3–6, 2017). Ed. by Arthur Tatnall and Mary Webb. Vol. 515. IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer International Publishing, 2017, pp. 506–514. ISBN: 978-3-319-74310-3. DOI: 10.1007/978-3-319-74310-3_51.
[132] Raniere Silva. *Translating Software Carpentry into Portuguese*. Software Carpentry Blog. July 2014. URL: https://software-carpentry.org/blog/2014/07/translating-software-carpentry-into-portuguese.html (visited on 06/19/2023).
[133] *Software Carpentry*. URL: https://software-carpentry.org/ (visited on 07/25/2023).
[134] Stiftung Innovation in der Hochschullehre. *About Us*. URL: https://stiftung-hochschullehre.de/en/about-us/ (visited on 07/03/2023).
[135] Victoria Stodden and Sheila Miguez. "Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research". In: *Journal of Open Research Software* 2.1 (July 2014), e21. DOI: 10.5334/jors.ay.
[136] Carly Strasser et al. "Ten simple rules for funding scientific open source software". In: *PLOS Computational Biology* 18.11 (Nov. 2022). Ed. by Scott Markel, e1010627. DOI: 10.1371/journal.pcbi.1010627.
[137] SusTrainable. *SusTrainable - Promoting Sustainability as a Fundamental Driver in Software Development Training and Education*. URL: https://sustrainable.github.io/index.html (visited on 07/12/2023).
[138] Tracy K. Teal et al. "Data Carpentry: Workshops to Increase Data Literacy for Researchers". In: *International Journal of Digital Curation* 10.1 (Mar. 2015), pp. 135–143. ISSN: 1746-8256. DOI: 10.2218/ijdc.v10i1.351.
[139] The Carpentries. *Become a Carpentries Instructor*. URL: https://docs.carpentries.org/topic_folders/for_instructors/new_instructors.html (visited on 07/12/2023).
[140] The Carpentries. *Carpentries-es*. URL: https://carpentries.org/latam-tf/ (visited on 06/19/2023).
[141] The Carpentries. *Certificates for The Carpentries*. URL: https://github.com/carpentries/learner-certificates (visited on 06/19/2023).
[142] The Carpentries. *Learner Certificates*. URL: https://docs.carpentries.org/topic_folders/hosts_instructors/certificates.html (visited on 07/12/2023).
[143] The Carpentries. *The Carpentries*. URL: https://carpentries.org (visited on 06/16/2023).
[144] The Carpentries. *The Carpentries Incubator*. URL: https://carpentries-incubator.org/ (visited on 06/16/2023).
[145] The Carpentries. *What does a badge mean? Instructor Training*. URL: https://carpentries.github.io/instructor-training/aio.html#what-does-a-badge-mean (visited on 06/19/2023).
[146] The Fedora Project. *Fedora Badges*. URL: https://badges.fedoraproject.org/explore/badges (visited on 06/15/2023).
[147] The HPC Certification Forum. *Competencies*. URL: https://www.hpc-certification.org/cs/ (visited on 06/16/2023).
[148] The Linux Foundation and The Open Source Security Foundation. *The Open Source Software Security Mobilization Plan*. Tech. rep. The Linux Foundation, May 2022. URL: https://openssf.org/oss-security-mobilization-plan/.
[149] *The Missing Semester of Your CS Education*. Massachusetts Institute of Technology. URL: https://missing.csail.mit.edu/ (visited on 07/21/2023).
[150] Rochelle E. Tractenberg et al. "The Mastery Rubric for Bioinformatics: A tool to support design and evaluation of career-spanning education and training". In: *PLOS ONE* 14.11 (Nov. 2019). Ed. by Nicholas J. Provart, e0225256. DOI: 10.1371/journal.pone.0225256.

[151] Asher Trockman et al. "Adding sparkle to social coding: an empirical study of repository badges in the *npm* ecosystem". In: *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)* (Gothenburg, Sweden, May 27–June 3, 2018). New York, New York, USA: ACM, May 2018, pp. 511–522. ISBN: 978-1-4503-5638-1. DOI: 10.1145/3180155.3180209.

[152] UNIVERSE HPC. *Understanding and Nurturing an Integrated Vision for Education in RSE and HPC*. URL: https://www.universe-hpc.ac.uk (visited on 06/16/2023).

[153] University of Suttgart. *digit@L – digital learning and teaching at the University of Suttgart: Boost. Skills. Support.* URL: https://www.project.uni-stuttgart.de/digital/en/ (visited on 07/03/2023).

[154] John Darrell Van Horn et al. "Democratizing data science through data science training". In: *Proceedings of the Pacific Symposium on Biocomputing 2018* (Kohala Coast, Hawaii, USA, Jan. 3–7, 2018). Ed. by Russ B. Altman et al. Singapore: World Scientific, Nov. 2017, pp. 292–303. ISBN: 978-981-3235-53-3. DOI: 10.1142/9789813235533_0027.

[155] Allegra Via et al. "A new pan-European Train-the-Trainer programme for bioinformatics: pilot results on feasibility, utility and sustainability of learning". In: *Briefings in Bioinformatics* 20.2 (Sept. 2017), pp. 405–415. DOI: 10.1093/bib/bbx112.

[156] Jing Wang. "Survival factors for Free Open Source Software projects: A multi-stage perspective". In: *European Management Journal* 30.4 (2012), pp. 352–371. ISSN: 0263-2373. DOI: https://doi.org/10.1016/j.emj.2012.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0263237312000199.

[157] Mary Webb et al. "Computer Science in the School Curriculum: Issues and Challenges". In: *Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing* (Dublin, Ireland, July 3–6, 2017). Ed. by Arthur Tatnall and Mary Webb. Vol. 515. IFIP Advances in Information and Communication Technology. Cham, Switzerland: Springer International Publishing, 2017, pp. 421–431. ISBN: 978-3-319-74310-3. DOI: 10.1007/978-3-319-74310-3_43.

[158] Lonnie Welch et al. "Applying, Evaluating and Refining Bioinformatics Core Competencies (An Update from the Curriculum Task Force of ISCB's Education Committee)". In: *PLOS Computational Biology* 12.5 (May 2016), e1004943. DOI: 10.1371/journal.pcbi.1004943.

[159] Lonnie Welch et al. "Bioinformatics Curriculum Guidelines: Toward a Definition of Core Competencies". In: *PLoS Computational Biology* 10.3 (Mar. 2014), e1003496. DOI: 10.1371/journal.pcbi.1003496.

[160] Ethan White et al. "Data Carpentry for Biologists: A semester long Data Carpentry course using ecological and other biological examples". In: *Journal of Open Source Education* 5.50 (Apr. 2022), p. 139. DOI: 10.21105/jose.00139.

[161] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3.1 (Mar. 2016), p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[162] Jason J. Williams et al. "Barriers to integration of bioinformatics into undergraduate life sciences education: A national study of US life sciences faculty uncover significant barriers to integrating bioinformatics into undergraduate instruction". In: *PLOS ONE* 14.11 (Nov. 2019). Ed. by Cesario Bianchi, e0224288. DOI: 10.1371/journal.pone.0224288.

[163] Greg Wilson. *Badges (Finalized)*. Software Carpentry Blog. Feb. 2012. URL: https://software-carpentry.org/blog/2012/02/badges-finalized.html (visited on 06/19/2023).

[164] Greg Wilson. *Badging*. Software Carpentry Blog. Jan. 2012. URL: https://software-carpentry.org/blog/2012/01/badging.html (visited on 06/19/2023).

[165] Greg Wilson. *Enseñar tecnología en comunidad. Cómo crear y dar lecciones que funcionen y construir una comunidad docente a su alrededor*. 2019. URL: https://teachtogether.tech/es/index.html (visited on 07/12/2023).

[166] Greg Wilson. *Sloan Foundation Grant to Software Carpentry and Mozilla*. Software Carpentry Blog. Jan. 2012. URL: https://software-carpentry.org/blog/2012/01/sloan-foundation-grant-to-software-carpentry-and-mozilla.html (visited on 06/19/2023).

[167] Greg Wilson. *Sloan Foundation Proposal Round 2*. Software Carpentry Blog. July 2013. URL: https://software-carpentry.org/blog/2013/07/sloan-proposal-round-2.html (visited on 06/19/2023).

[168] Greg Wilson. "Software Carpentry: Getting Scientists to Write Better Code by Making Them More Productive". In: *Computing in Science & Engineering* 8.6 (Nov. 2006), pp. 66–69. DOI: 10.1109/mcse.2006.122.

[169] Greg Wilson. "Software Carpentry: lessons learned [version 2; peer review: 3 approved]". In: *F1000Research* 3.62 (Jan. 2016). DOI: 10.12688/f1000research.3-62.v2.

[170] Greg Wilson. *Teaching Tech Together. How to Make Your Lessons Work and Build a Teaching Community around Them*. CRC Press, 2019. ISBN: 9780429330704. URL: https://www.routledge.com/Teaching-Tech-Together-How-to-Make-Your-Lessons-Work-and-Build-a-Teaching/Wilson/p/book/9780367352974.

[171] Greg Wilson. *Teaching Tech Together. How to Make Your Lessons Work and Build a Teaching Community around Them*. 2019. URL: https://teachtogether.tech/en/index.html (visited on 07/12/2023).

[172] Greg Wilson. *Translating Software Carpentry into Korean*. Software Carpentry Blog. Nov. 2014. URL: https://software-carpentry.org/blog/2014/11/korean-translation.html (visited on 06/19/2023).

[173] Greg Wilson. *Translating Software Carpentry into Spanish*. Software Carpentry Blog. June 2014. URL: https://software-carpentry.org/blog/2014/06/translating-software-carpentry-into-spanish.html (visited on 06/19/2023).

[174] Greg Wilson et al. "Best Practices for Scientific Computing". In: *PLoS Biology* 12.1 (Jan. 2014). Ed. by Jonathan A. Eisen, e1001745. DOI: 10.1371/journal.pbio.1001745.

[175] Melissa A. Wilson Sayres et al. "Bioinformatics core competencies for undergraduate life sciences education". In: *PLOS ONE* 13.6 (June 2018). Ed. by Andrew R. Dalby, e0196878. DOI: 10.1371/journal.pone.0196878.

[176] Lou Woodley et al. *The CSCCE Skills Wheel. Five core competencies and 45 skills to describe the role of the community engagement manager in STEM*. Guidelines. Center for Scientific Collaboration and Community Engagement, Jan. 2021. DOI: 10.5281/zenodo.4437294.

[177] Li Zhou et al. "Students' Perception of Using Digital Badges in Blended Learning Classrooms". In: *Sustainability* 11.7 (Apr. 2019), p. 2151. DOI: 10.3390/su11072151.