Collaboration and Professionalization. The role of software and Research Software Engineers in the Digital Humanities.

Czmiel, Alexander
czmiel@bbaw.de
Berlin-Brandenburg Academy of Sciences and Humanities

Henny-Krahmer, Ulrike
ulrike.henny-krahmer@uni-rostock.de
University of Rostock

Jettka, Daniel
daniel.jettka@uni-paderborn.de
Paderborn University

Developing research software, in addition to applying digital methods, is an important part of research activities in the Digital Humanities, but still mostly takes place in a less professionalized form (see Schrade et al. 2018). It is true that in some areas of DH it is possible to rely on existing software solutions developed outside the DH itself, some of which are commercial, for certain tasks, e.g., the Oxygen XML editor (SyncRO Soft SRL 2022) for editing XML files or the Gephi software (Bastian et al. 2009) for visualizing network data. Often, however, such general tools are not appropriate for answering humanities research questions and projects, and more specialized tools are not available. This often leads digital humanities researchers to become software engineers themselves or to work closely together with research software engineers (RSE) to develop new research software.

The nature and extent of software development in the humanities can vary greatly. In the same way, the stake and importance that the activity "software engineering" has for researchers in the DH can also vary greatly. The development of research software can be part of an overall activity, e.g., when an entire project is carried out by individuals, from the humanities research question to the operationalization and technical implementation to the interpretation and processing and publication of the results. However, software development can also become the main activity for researchers, especially in larger teams working in division of labor, or when the development of research software is the focus of a project. For those who are mainly involved in the development of research software, the term Research Software Engineer has become established.

Especially because software engineering in the context of research in the humanities and in the DH is rarely carried out by software engineers or computer scientists trained for this purpose, there is still much room and a lot of opportunities for fostering professionalization in this area. The researchers themselves acquire the basics of individual programming languages to develop software through courses or additional training. For this additional competence and the additional work, which results from the development of professional software, there is however only rarely the deserved scientific acknowledgment from the scholarly community. Since neither the knowledge of a stable and maintainable software architecture nor sufficient time for tests or even adequate documentation is available, the software itself is in need for improvement and not prepared for sustainable use.

Those situations and challenges will be discussed during a workshop at the annual DHd conference taking place in March 2023. The goal of this contribution to the DH conference is to structure, visualize, and present the outcome of the workshop as a poster to aim to connect DH-RSEs on an international level as well and to raise the awareness of this topic and increase professionalization through collaboration on all levels. Professionalization becomes more important with the increasing importance of software development in the research process and as a scientific activity in general, including all areas of DH. It can be roughly described on three levels:

1. Technical Professionalization

Technical professionalization primarily concerns the areas that are important in day-to-day software development work at the code and architecture level, e.g., clean code (Martin 2009, Clean Code Developer 2022), versioning, documentation, testing, DevOps (Halstenberg et al. 2020), etc.

2. Organizational Professionalization

Organizational professionalization refers to all aspects of planning, project management, collaboration, and (self-)organization of software engineers. This includes, among other things, the formation of developer teams, the use of version management and ticket systems, methods of agile software development (Beck et al. 2001), but also software management plans (SMPs).

3. Institutional Professionalization

Institutional professionalization creates the framework for the other two areas by making software development part of DH curricula and improving training for RSEs in DH, but also by enabling career paths and collaboration opportunities. It is about providing resources and infrastructure for software development and anchoring the activity of an RSE in the academic field.

The poster will present the state of discussion and approaches to solutions in these three areas.

Bibliography

Bastian, Mathieu / Heymann, Sebastien / Jacomy, Mathieu (2009): *Gephi: an open source software for exploring and manipulating networks.* In International AAAI Conference on Web and Social Media. https://gephi.org/publications/gephi-bastian-feb09.pdf.

Beck, Kent / Beedle, Mike / van Bennekum, Arie / Cockburn, Alistair / Cunningham, Ward / Fowler, Martin / Grenning, James / Highsmith, Jim / Hunt, Andrew / Jeffries, Ron / Kern, Jon / Marick, Brian / Martin, Robert C. / Mellor, Steve / Schwaber, Ken / Sutherland, Jeff / Thomas, Dave (2001): *Manifest für Agile Softwareentwicklung*. http://agilemanifesto.org/iso/de/manifesto.html. Clean Code Developer. 2022. „Clean Code Developer.“ https://clean-code-developer.de/.

Goble, Carole (2014): *Better Software, Better Research*. Software Sustainability Institute. https://www.software.ac.uk/resources/publications/better-software-better-research.

Halstenberg, Jürgen / **Pfitzinger, Bernd** / **Jestädt, Thomas** (2020): *DevOps: Ein Überblick*. Wiesbaden: Springer.

Hettrick, Simon (2020): *The growth and professionalisation of Research Software Engineering*. https://slides.com/simonhettrick/rse-professionalisation-cw20.

Martin, Robert C. (2009): Clean Code. Refactoring, Patterns, Testen und Techniken für sauberen Code. Frechen: mitp-Verlag.

Schrade, Torsten / **Czmiel, Alexander** / **Druskat, Stephan** (2018): *Research Software Engineering und Digital Humanities. Reflexion, Kartierung, Organisation*. In DHd 2018. Book of Abstracts. https://doi.org/10.5281/zenodo.4622564.

SyncRO Soft SRL (2022): *Oxygen XML Editor*. https://www.oxygenxml.com/xml_editor.html.