Roles in Research Software Engineering (RSE) Consultancies

Carina Haupt* and Martin Stoffers†
* *Institute for Software Technology*, *German Aerospace Center (DLR)*, Berlin, Germany
carina.haupt@dlr.de
† *Institute for Software Technology*, *German Aerospace Center (DLR)*, Cologne, Germany
martin.stoffers@dlr.de

***Abstract*—Consultation services are an helpful tool to support scientists in developing software. Different types of knowledge are required to perform an efficient and effective consultation. Based on our experience, by providing a consultation service for 18 research centers with over 43,000 employees, we defined five roles and show their relevance on three consultations. Being aware of these roles and trying to cover them when setting up a consultancy, is an important step towards a good consultancy.**

I. Introduction

Using and developing software increasingly became part of the daily work of most scientists [1], [2]. Especially the development requires knowledge outside their domain, such as efficient usage of software engineering tools, developing architecture design based on their requirements, and publishing their code as Open Source Software (OSS).

Consultations are one way to transfer knowledge about software engineering from software experts to domain scientists. Allowing scientist to focus on their research work by reducing their time spent on acquiring software engineering knowledge.

In 2019, the Helmholtz Association started the Helmholtz IT for Science (HIFIS) platform to support Research Software Engineering (RSE) with a high level of quality, visibility, and sustainability.[1] As part of it, HIFIS started a consultancy, offering support to over 43,000 employees from a total of 18 scientific-technical and biological-medical research centers.[2]

During the first year of service, we learned that consultations require different types of knowledge. Depending on the request, the consultant might need knowledge about the broad technical topic, a specific tool, the domain, and/or the organisation. Therefore we defined five consultant roles. In the following we will

- give a short introduction into the HIFIS consulting service (Section II),
- describe the different consultant roles (Section III),
- show their relevance on selected consultations (Section IV),
- and conclude their necessity (Section V).

II. HIFIS Consultancy

The HIFIS consultancy was officially started in July 2020. Any scientist part of a German Helmholtz center can request a consultation by filling out a short survey and a group of consultants will support them with their request.[3] If necessary, external experts are involved. These can be from the HIFIS platform or from the participating Helmholtz centers. The service is financed by the HIFIS platform and therefore the consultation is free of charge for the requester.

From July to December 2020, 24 consultation requests were handled by a team of seven consultants from four different centers. At its start the service was actively advertised only at four Helmholtz centers, allowing a soft launch. The requests can be categorized in two categories regarding their scale: Those requiring one or two calls followed by a summary and additional information provided by email, and those requiring more resources. The first type is typically related to topics such as open source, software architecture, and programming, while the second category is related to broader software engineering topics such as change management, documentation, developer on-boarding, automated testing, and refactoring. A closer look at the request categories and types will be subject to a study in the end of 2021.

III. Consultant roles

Consultancies require different types of knowledge. Therefore we defined five consultant roles which might be required in a consultation. A consultant can cover more than one role.

A. Role: Technical domain

What is the best way to achieve X? Broad knowledge about the respective technical field is needed to answer such a question. The consultant should know about existing solutions and their advantages and disadvantages. Additionally, they should be able to present them as well as give a suggestion which one is an appropriate solution.

B. Role: Tool

How do I do this with tool X? In-depth knowledge about the tool is necessary to give an helpful answer to such questions. In-depth hereby means knowing how to handle use-cases outside of the tools tutorials and how-to documentation as well as how to debug occurring errors.

[1] https://www.hifis.net/
[2] https://www.helmholtz.de/en/about-us/
[3] https://www.hifis.net/services/software/consulting.html

C. Role: Scientific domain

Domain knowledge means knowledge about the scientific application domain of the consultation request, whereby here IT is explicitly excluded. Domain knowledge can be helpful in several regards. First, to understand the underlying problem which needs to be solved. Second, to know about typical methods, tools, or processes used in the domain. An efficient consultation should not only result in a good solution, but also an appropriate one for its application domain.

D. Role: Organisation

Organisations offer tools, services, communities (i.e. Software Engineering Community at the German Aerospace Center (DLR) [3]), and official contacts for specific topics. Also they have different requirements (i.e. Software Engineering Guideline at DLR [4]) and recommendations. Requesters might, but do not necessarily do, know about these. Ideally the consultant works for the same organisation or at least has some knowledge about the previously mentioned aspects to make the consultation efficient and incorporated.

E. Role: Organisational

This role is required to be covered for each consultation. The consultation itself follows a defined process. This process has to be known and followed to ensure not only an efficient consultation, but also a correct documentation. The documentation is the basis for yearly reports, quality control, and scientific analysis. While each HIFIS consultant knows the process well, external experts, which might take over other roles of the consultation, do not.

IV. Consultations

In the following, three real life consultations are presented to show the relevance of the roles described in Section III. While these are hand picked examples, they are representative of the consultations performed by the HIFIS consultancy. The consultations are abstracted and anonymized.

A. Consulting: Embedded architecture

Requester Alex and their group needed support regarding architectural design of an embedded systems software project. Two consultants supported them. HIFIS consultant Blake has the *Organisational* role. External expert Charlie provides its knowledge about embedded systems tools (especially IDEs) and therewith takes on the *Tool* role. Both inhabit the *Technical Domain* role as they provide their shared knowledge about architectural design for embedded systems. Sharing this role allowed them to combine their knowledge, discuss ideas, and developing an better approach.

B. Consulting: Open Source Software publishing

Requester Drew planned on publishing their software as Open Source Software. The HIFIS consultant Ezra is an expert in this field and is able to take over the roles *Technical domain* and *Tool* additionally to the default *Organisational* role. Being from the same organisation as Drew, Ezra takes over the *Organisation* role as well. Ezra knows about the OSS publishing process, their license and platform recommendations as well as who to ask for permission for their organisation. In similar consultation, where the requester was from another organisation, Ezra could only explain that these topics should be addressed, but not give specific advice.

C. Consulting: Signal processing

Requester Francis and their group started to develop a software for signal processing and want to discuss their plans for their architecture with an expert. Luckily HIFIS consultant Gill does not only know about architecture design but about signal processing as well. This domain knowledge was beneficial when evaluating the proposed software architectures. By being from the same organisation as Francis, Gill was able to point out further aspects that needed to be considered. Therewith Gill unites the *Technical domain*, *Scientific domain*, *Organisation*, and *Organisational*.

V. Conclusion

There are five roles in consultancies: *Technical domain*, *Tool*, *Scientific domain*, *Organisation*, and *Organisational* (see Section III). Not all of them are relevant for each consultation.

- *Organisational* has to be covered in all consultancies due to it's nature.
- *Technical domain* and/or *Tool* occur in most consultancies, since the need of support in these fields is a typical reason to appeal to an IT consultation service,
- *Scientific domain* and *Organisation* are mandatory in certain cases, but can immensely improve the quality of the consultation in many more cases. Therewith, they can be seen as supporting roles to the *Technical domain* and/or *Tool* roles.

Roles can be covered by a single consultant, distributed via several consultants and/or experts, or even shared. Important is that they are covered when needed.

Depending on your consultancy context, the roles can have different scopes. If you support only one research institute, technical and tool knowledge might need to cover a narrower field and knowledge of the scientific domain and organisation may be more important but also naturally existent.

Considering these roles from the beginning influences the structure and personal selection of a consultancy as well as the level of service that can be provided. For us this meant to

- extract the consultation process as a separate task,
- acquire external experts from as many organisations and scientific domains as possible,
- and teach each other requirements and recommendations from each others organisations.

Conclusively, we think that the five described roles should be considered when setting up a consultancy, hiring consultants, and defining an consultation process.

REFERENCES

[1] U. Nangia and D. S. Katz, "Track 1 Paper: Surveying the U.S. National Postdoctoral Association Regarding Software Use and Training in Research." Zenodo, Jun. 2017, This paper was submitted to WSSSPE5.1 - http://wssspe.researchcomputing.org.uk/wssspe5-1/ The final accepted version is https://doi.org/10.6084/m9.figshare.5328442. [Online]. Available: https://doi.org/10.5281/zenodo.814220

[2] J. Howison and J. Bullard, "Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature," *Journal of the Association for Information Science and Technology*, vol. 67, no. 9, pp. 2137–2155, 2016. [Online]. Available: http://dx.doi.org/10.1002/asi.23538

[3] C. Haupt and T. Schlauch, "The software engineering community at dlr: How we got where we are," in *Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE5.1)*, N. C. Hong, S. Druskat, R. Haines, C. Jay, D. S. Katz, and S. Sufi, Eds., September 2017. [Online]. Available: https://elib.dlr.de/114050/

[4] C. Haupt, M. Meinel, and T. Schlauch, "The software engineering initiative of DLR: Overcome the obstacles and develop sustainable software," in *2018 IEEE/ACM 13th International Workshop on Software Engineering for Science (SE4Science)*, 2018, pp. 16–19.