FIRST RESULTS ON THE SECURITY CULTURE SURVEY

de-RSE’24, 05. – 07.03.2024, Universität Würzburg

Michael Meinel, Institut für Softwaretechnologie, Berlin

About Me

German Aerospace Center (DLR)

- Research in Aeronatuics, Space, Transportation, Energy, Security, and Digitalisation
- 50+ Institutes, 30+ Sites in Germany, 10 000+ Employees

Sustainable Software Engineering

- Group within „Institute for Software Technology“
- „Classical RSE work“
  - Guidelines: rse.dlr.de
  - Tools + Trainings
  - Support + Consulting

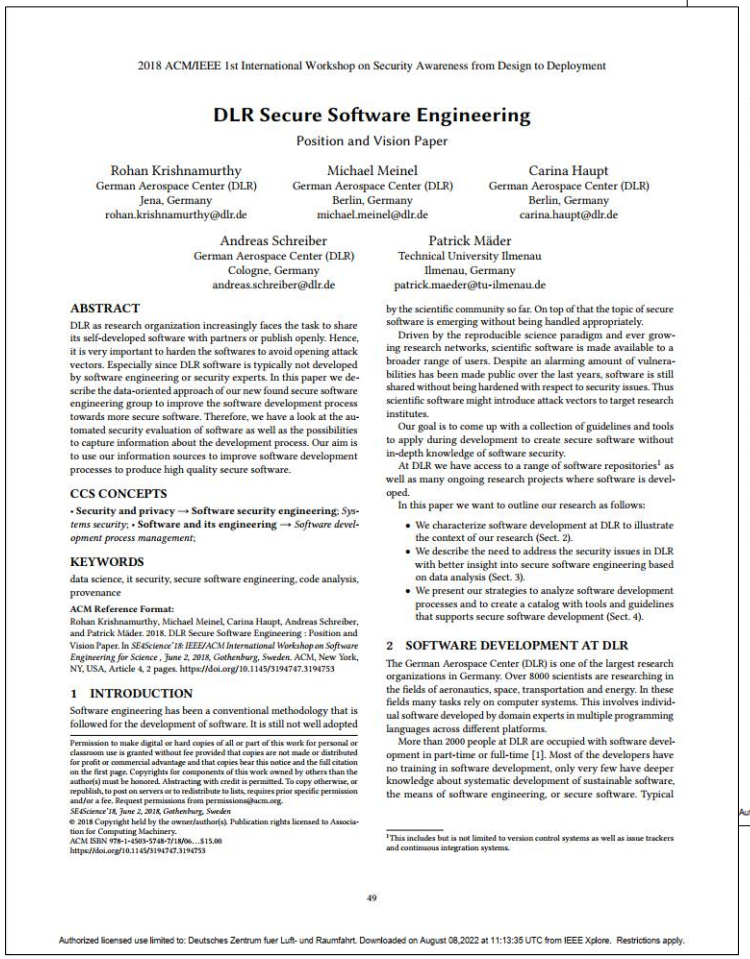
FIRST, SOME HISTORY!

DLR Secure Software Engineering

2018 ACM/IEEE 1st International Workshop on Security Awareness from Design to Deployment

Position and Vision Paper

- Published in 2018 at ICSE Workshop „SEAD'18“
- Possible Collaboration ideas with newly founded Institute for Data Science (DW)
  - Data driven, automated security audition
  - Adoption of Software Engineering methods and processes to increase security
- DW is now operational



2018 ACM/IEEE 1st International Workshop on Security Awareness from Design to Deployment

DLR Secure Software Engineering

Position and Vision Paper

Rohan Krishnamurthy
German Aerospace Center (DLR)
Jena, Germany
rohan.krishnamurthy@dlr.de

Michael Meinel
German Aerospace Center (DLR)
Berlin, Germany
michael.meinel@dlr.de

Carina Haupt
German Aerospace Center (DLR)
Berlin, Germany
carina.haupt@dlr.de

Andreas Schreiber
German Aerospace Center (DLR)
Cologne, Germany
andreas.schreiber@dlr.de

Patrick Mäder
Technical University Ilmenau
Ilmenau, Germany
patrick.maeder@tu-ilmenau.de

ABSTRACT

DLR as research organization increasingly faces the task to share its self-developed software with partners or publish openly. Hence, it is very important to harden the softwares to avoid opening attack vectors. Especially since DLR software is typically not developed by software engineering or security experts. In this paper we describe the data-oriented approach of our new found secure software engineering group to improve the software development process towards more secure software. Therefore, we have a look at the automated security evaluation of software as well as the possibilities to capture information about the development process. Our aim is to use our information sources to improve software development processes to produce high quality secure software.

CCS CONCEPTS

• **Security and privacy → Software security engineering**; *Systems security*; • **Software and its engineering →** *Software development process management*;

KEYWORDS

data science, it security, secure software engineering, code analysis, provenance

ACM Reference Format:
Rohan Krishnamurthy, Michael Meinel, Carina Haupt, Andreas Schreiber, and Patrick Mäder. 2018. DLR Secure Software Engineering : Position and Vision Paper. In *SE4Science'18: IEEE/ACM International Workshop on Software Engineering for Science , June 2, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3194747.3194753

1 INTRODUCTION

Software engineering has been a conventional methodology that is followed for the development of software. It is still not well adopted by the scientific community so far. On top of that the topic of secure software is emerging without being handled appropriately.

Driven by the reproducible science paradigm and ever growing research networks, scientific software is made available to a broader range of users. Despite an alarming amount of vulnerabilities has been made public over the last years, software is still shared without being hardened with respect to security issues. Thus scientific software might introduce attack vectors to target research institutes.

Our goal is to come up with a collection of guidelines and tools to apply during development to create secure software without in-depth knowledge of software security.

At DLR we have access to a range of software repositories[1] as well as many ongoing research projects where software is developed.

In this paper we want to outline our research as follows:

- We characterize software development at DLR to illustrate the context of our research (Sect. 2).
- We describe the need to address the security issues in DLR with better insight into secure software engineering based on data analysis (Sect. 3).
- We present our strategies to analyze software development processes and to create a catalog with tools and guidelines that supports secure software development (Sect. 4).

2 SOFTWARE DEVELOPMENT AT DLR

The German Aerospace Center (DLR) is one of the largest research organizations in Germany. Over 8000 scientists are researching in the fields of aeronautics, space, transportation and energy. In these fields many tasks rely on computer systems. This involves individual software developed by domain experts in multiple programming languages across different platforms.

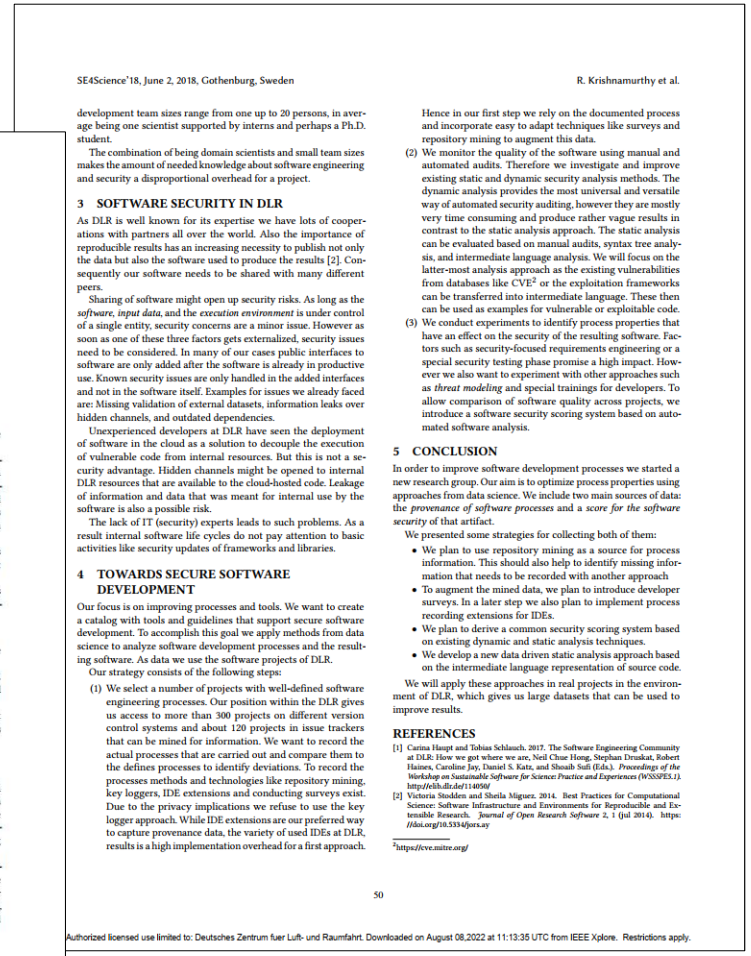
More than 2000 people at DLR are occupied with software development in part-time or full-time [1]. Most of the developers have no training in software development, only very few have deeper knowledge about systematic development of sustainable software, the means of software engineering, or secure software. Typical development team sizes range from one up to 20 persons, in average being one scientist supported by interns and perhaps a Ph.D. student.

The combination of being domain scientists and small team sizes makes the amount of needed knowledge about software engineering and security a disproportional overhead for a project.

[1]This includes but is not limited to version control systems as well as issue trackers and continuous integration systems.

SE4Science'18, June 2, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5748-7/18/06…$15.00
https://doi.org/10.1145/3194747.3194753



3 SOFTWARE SECURITY IN DLR

As DLR is well known for its expertise we have lots of cooperations with partners all over the world. Also the importance of reproducible results has an increasing necessity to publish not only the data but also the software used to produce the results [2]. Consequently our software needs to be shared with many different peers.

Sharing of software might open up security risks. As long as the *software*, *input data*, and the *execution environment* is under control of a single entity, security concerns are a minor issue. However as soon as one of these three factors gets externalized, security issues need to be considered. In many of our cases public interfaces to software are only added after the software is already in productive use. Known security issues are only handled in the added interfaces and not in the software itself. Examples for issues we already faced are: Missing validation of external datasets, information leaks over hidden channels, and outdated dependencies.

Unexperienced developers at DLR have seen the deployment of software in the cloud as a solution to decouple the execution of vulnerable code from internal resources. But this is not a security advantage. Hidden channels might be opened to internal DLR resources that are available to the cloud-hosted code. Leakage of information and data that was meant for internal use by the software is also a possible risk.

The lack of IT (security) experts leads to such problems. As a result internal software life cycles do not pay attention to basic activities like security updates of frameworks and libraries.

4 TOWARDS SECURE SOFTWARE DEVELOPMENT

Our focus is on improving processes and tools. We want to create a catalog with tools and guidelines that support secure software development. To accomplish this goal we apply methods from data science to analyze software development processes and the resulting software. As data we use the software projects of DLR.

Our strategy consists of the following steps:

(1) We select a number of projects with well-defined software engineering processes. Our position within the DLR gives us access to more than 300 projects on different version control systems and about 120 projects in issue trackers that can be mined for information. We want to record the actual processes that are carried out and compare them to the defines processes to identify deviations. To record the processes methods and technologies like repository mining, key loggers, IDE extensions and conducting surveys exist. Due to the privacy implications we refuse to use the key logger approach. While IDE extensions are our preferred way to capture provenance data, the variety of used IDEs at DLR, results is a high implementation overhead for a first approach. Hence in our first step we rely on the documented process and incorporate easy to adapt techniques like surveys and repository mining to augment this data.

(2) We monitor the quality of the software using manual and automated audits. Therefore we investigate and improve existing static and dynamic security analysis methods. The dynamic analysis provides the most universal and versatile way of automated security auditing, however they are mostly very time consuming and produce rather vague results in contrast to the static analysis approach. The static analysis can be evaluated based on manual audits, syntax tree analysis, and intermediate language analysis. We will focus on the latter-most analysis approach as the existing vulnerabilities from databases like CVE[2] or the exploitation frameworks can be transferred into intermediate language. These then can be used as examples for vulnerable or exploitable code.

(3) We conduct experiments to identify process properties that have an effect on the security of the resulting software. Factors such as security-focused requirements engineering or a special security testing phase promise a high impact. However we also want to experiment with other approaches such as *threat modeling* and special trainings for developers. To allow comparison of software quality across projects, we introduce a software security scoring system based on automated software analysis.

5 CONCLUSION

In order to improve software development processes we started a new research group. Our aim is to optimize process properties using approaches from data science. We include two main sources of data: the *provenance of software processes* and a *score for the software security* of that artifact.

We presented some strategies for collecting both of them:

- We plan to use repository mining as a source for process information. This should also help to identify missing information that needs to be recorded with another approach
- To augment the mined data, we plan to introduce developer surveys. In a later step we also plan to implement process recording extensions for IDEs.
- We plan to derive a common security scoring system based on existing dynamic and static analysis techniques.
- We develop a new data driven static analysis approach based on the intermediate language representation of source code.

We will apply these approaches in real projects in the environment of DLR, which gives us large datasets that can be used to improve results.

REFERENCES

[1] Carina Haupt and Tobias Schlauch. 2017. The Software Engineering Community at DLR: How we got where we are, Neil Chue Hong, Stephan Druskat, Robert Haines, Caroline Jay, Daniel S. Katz, and Sheaib Sufi (Eds.). *Proceedings of the Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE1).* http://elib.dlr.de/114050/

[2] Victoria Stodden and Sheila Miguez. 2014. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. *Journal of Open Research Software* 2, 1 (jul 2014). https://doi.org/10.5334/jors.ay

[2]https://cve.mitre.org/

Full Paper:
https://doi.org/10.23919/SEAD.2018.8472854

REVISITING SECURE SOFTWARE ENGINEERING FOR RESEARCH SOFTWARE

RSECon UK 2022,

Michael Meinel, Martin Stoffers – German Aerospace Center (DLR)

DLR

Prof. Dr. Michael Felderer joins the DLR

- New head of institute since beginning of 2023
- Brings in new topics:
  - Explicit RSE Research
  - Security Topics
  - …
- Comes with a broad network

Getting attention…

Subject: Secure Software Engineering for Research Software

Michael,

I just listened to your talk from the RSECon2022. I found it very interesting. I have interacted with a number of people from DLR in the past including Carina Haupt, Tobias Schlauch, and Stephan Druskat.

A brief introduction of me. I have been working with Software Engineering for Research Software for a long time. I have been one of the primary organizers of the SE4Science workshop series (https://se4science.org/workshops/). I am also on the Steering Committee for the US-RSE association. I have done a lot of research on adapting various software engineering practices for appropriate use in research software. I use empirical methods to conduct human-based research. I've also done some work on security and research software. I attach an editorial that will appear in the Computing in Science & Engineering magazine at some point in the future. This is definitely a topic of interest to me.

You can find more about my work on my webpage: http://carver.cs.ua.edu/

I thought I would contact you to see if there were any projects where my expertise might be useful or where we might collaborate in this area or others.

Look forward to hearing from you.

Best,

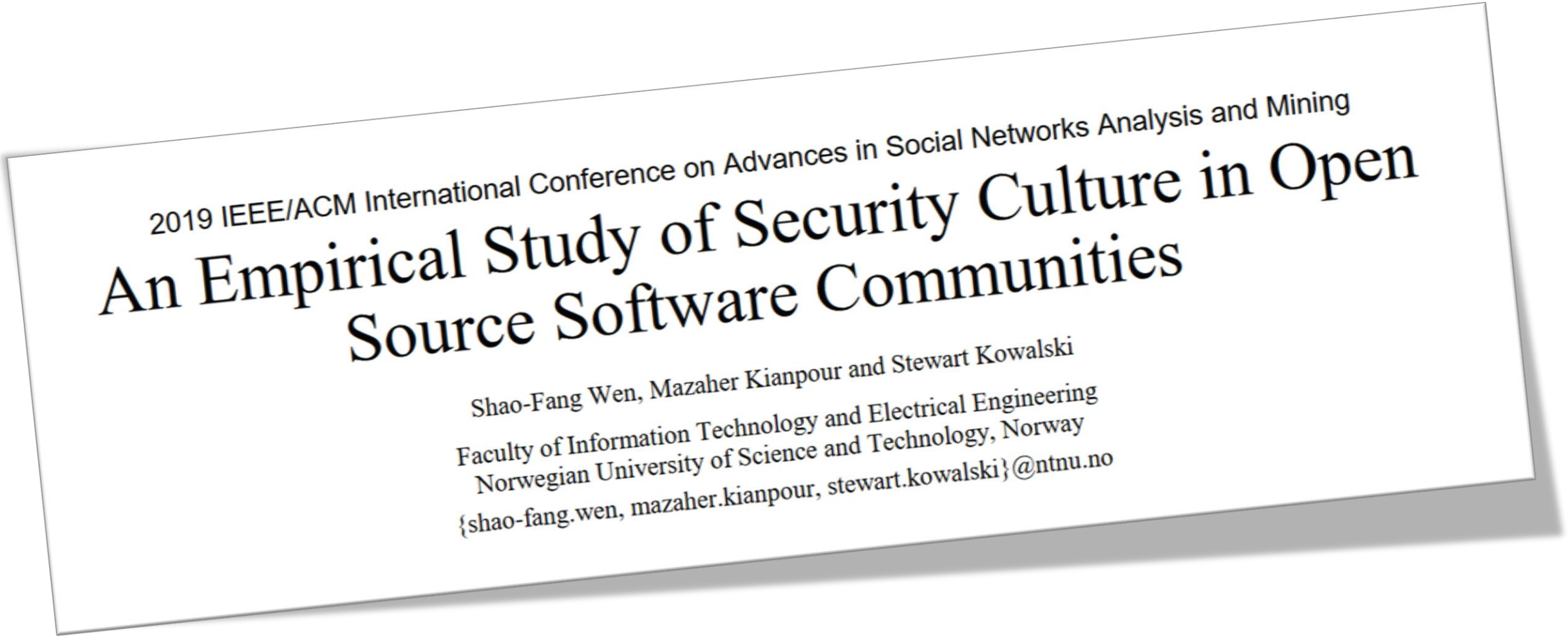
-- Jeff

So now, we‘re ready to go…

- Some kind of expertise in security topics ✓
- A great new head of institute interested in Secure Software Engineering ✓
- An even greater team and network, that helps me do my research ✓
- Interesting and interested collaboration partners ✓
- A good starting point to begin research ✗

Where to look at?

Jeff Carver, UA et. al.: „Let´s replicate this empirical study, but for RSEs“



2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

An Empirical Study of Security Culture in Open Source Software Communities

Shao-Fang Wen, Mazaher Kianpour and Stewart Kowalski

Faculty of Information Technology and Electrical Engineering
Norwegian University of Science and Technology, Norway

{shao-fang.wen, mazaher.kianpour, stewart.kowalski}@ntnu.no

Acknoledgements

- Thanks to Jeff Carver et. al. for preparing this study and teaming up with us!
  - Demographic questions were developed by UA staff
  - Additional Vingettes were added by UA
  - We were allowed to add our own questions
    (about EU Cyber Resilience Act and legal restrictions)

- Thanks to Michael Felderer for supporting the idea and the approach!

- But most of all: Thank you for participating!
  - Survey ran from mid December to mid January (approx. 5 weeks)
  - Survey server was down for the whole first week in 2024
  - Yet, we got 66 full and 79 partial answers (i.e., those who at least filled in the first half)

Disclaimer

- I‘m only allowed to show our (German) results…
  Well, I don‘t have the other data yet.

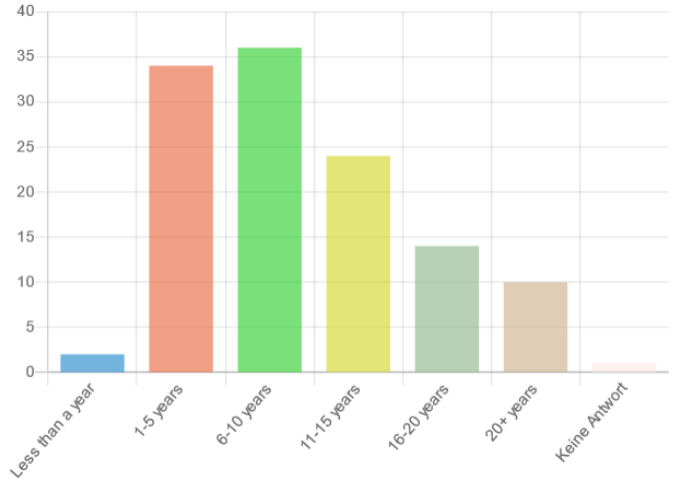
- I just started in evaluating the data (using Jupyter).
  Numbers and diagrams are not yet cross-checked and hardly reproducible.

- This slide deck was finished just about midnight…
  (... other stuff more important, like the HERMES and Reproducibility WS)

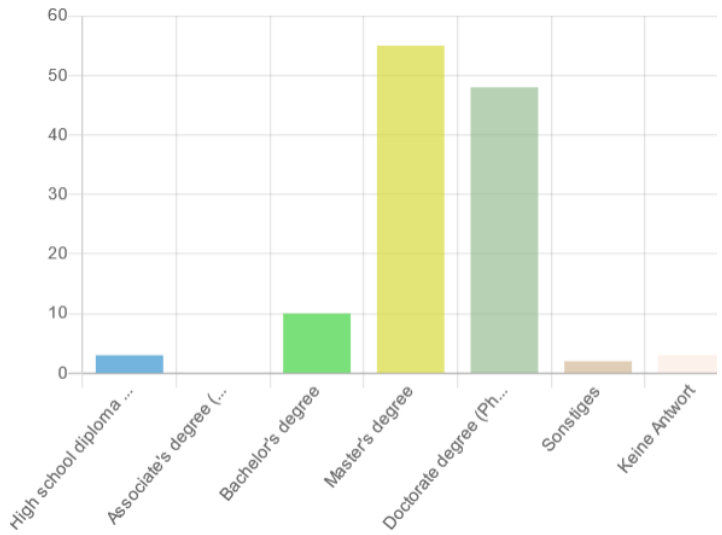
- Nevertheless, lets see the results we have…

Who did participate?

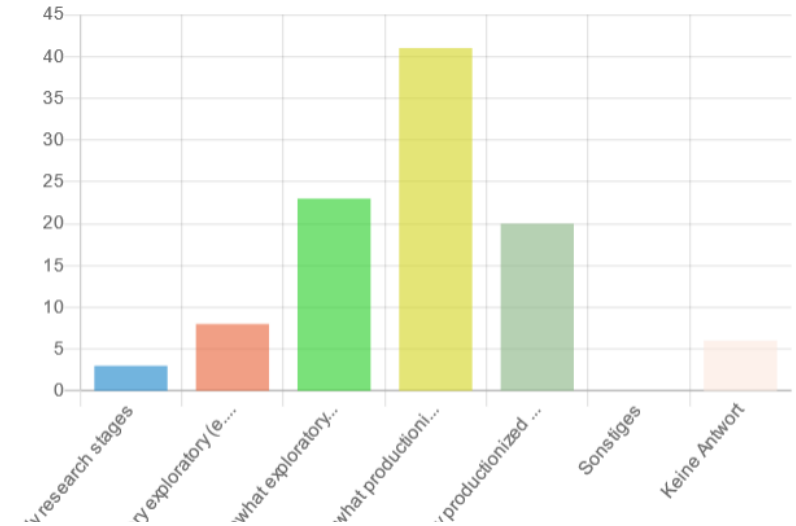
How many years of professional experience do you have?



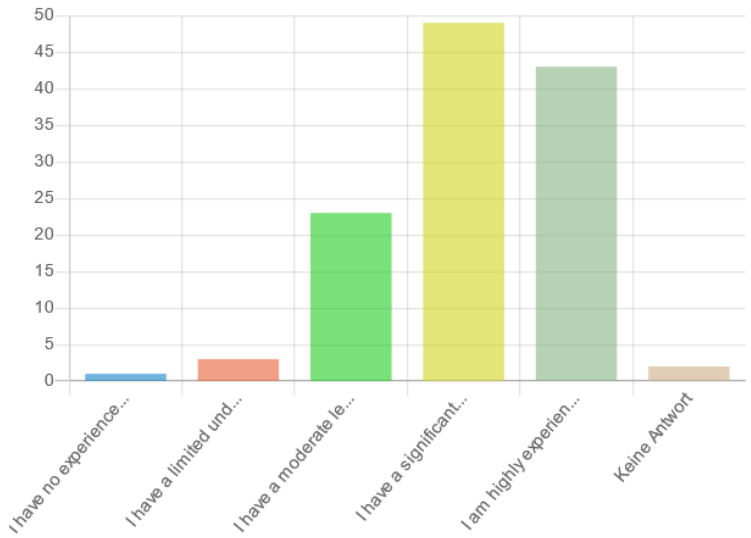
What is the highest level of education you have completed?



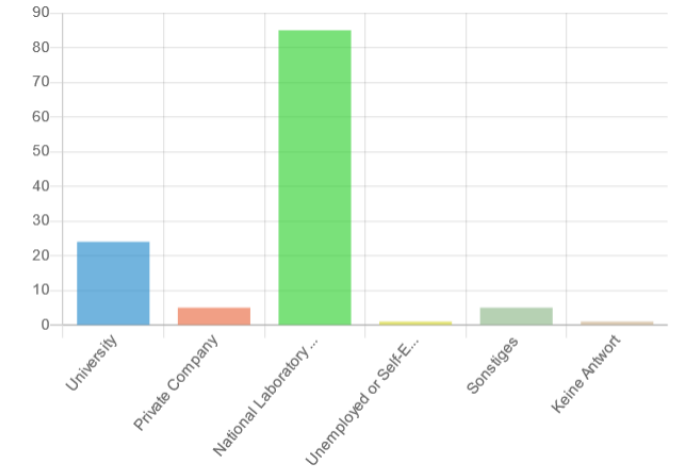
Which of the following best describes the status of your project?



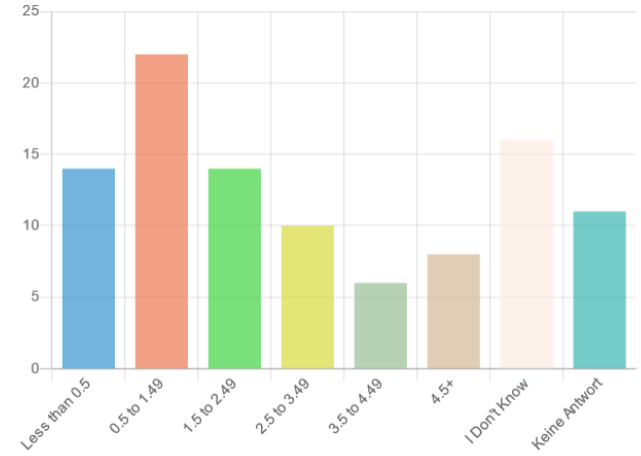
How much experience do you have with software development?



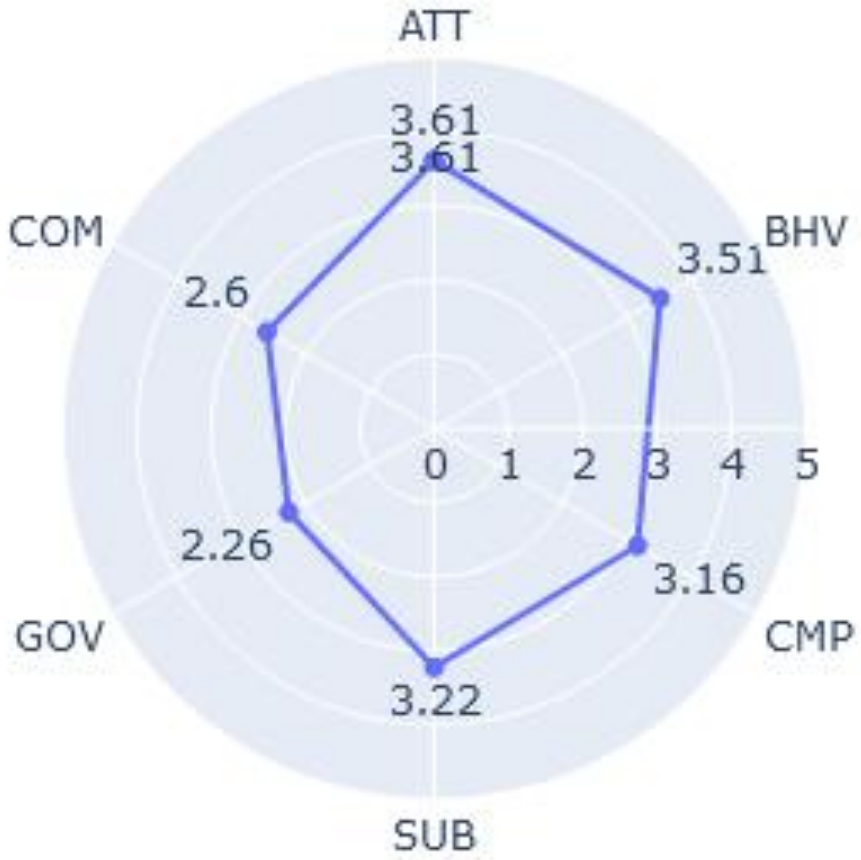
Which of the following best describes your current institution?

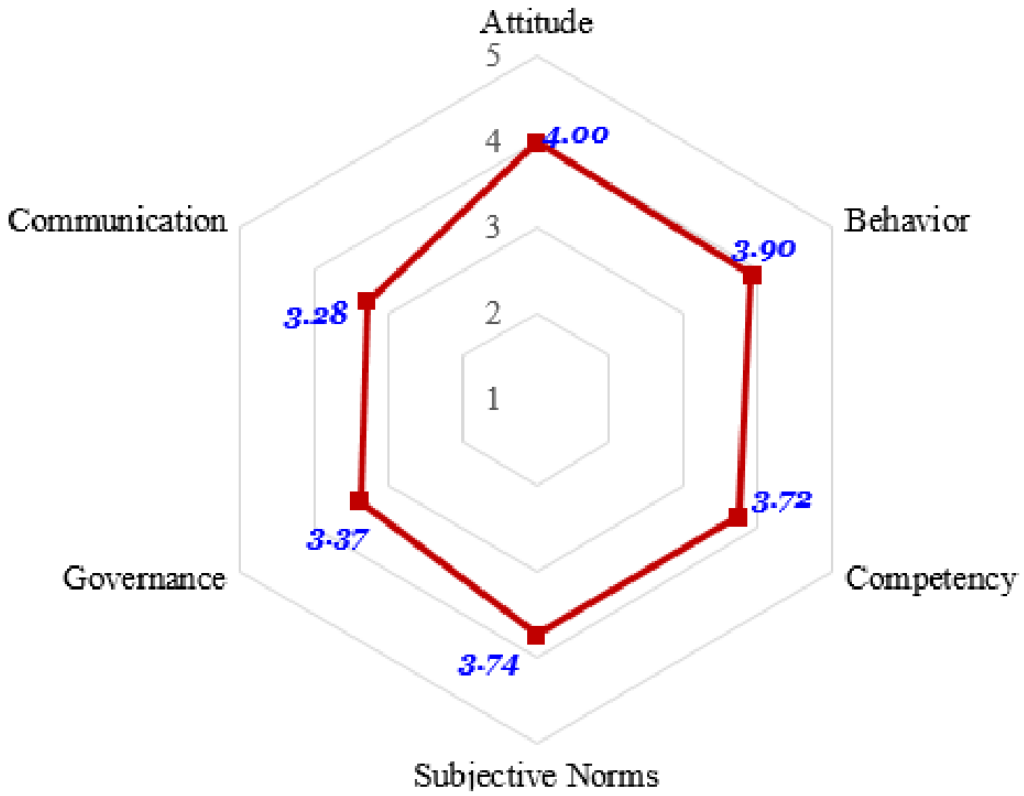


Approximately how many full-time employees (FTEs) are/were allocated to the project?



Overall results, compared to original study





https://doi.org/10.1145/3341161.3343520

Attitude

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Value	1%	3%	6%	30%	60%
	5%	13%	16%	38%	28%
Responsibility	4%	10%	30%	26%	30%
	5%	14%	23%	30%	28%
Positivity	5%	9%	20%	26%	40%
	3%	15%	30%	34%	18%

Behaviour

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Acts	2%	7%	20%	37%	33%
	0%	14%	18%	47%	22%
Compliance	2%	7%	17%	38%	36%
	3%	14%	28%	42%	14%
Risk-Taking	2%	10%	26%	31%	30%
	10%	23%	16%	32%	19%

Competency

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Knowledge	4%	9%	22%	40%	26%
	6%	22%	20%	42%	10%
Skills	3%	11%	21%	41%	24%
	9%	25%	27%	33%	6%
Effectiveness	5%	10%	19%	43%	23%
	6%	27%	22%	35%	10%

Subjective Norms

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Trust	3%	7%	11%	37%	41%
	5%	19%	35%	37%	4%
Supportiveness	6%	9%	20%	33%	32%
	9%	23%	24%	20%	4%
Expectation	9%	12%	27%	30%	21%
	9%	23%	44%	20%	4%

Governance

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Expertise	8%	14%	33%	27%	19%
	54%	22%	8%	15%	1%
Policies	10%	17%	22%	28%	23%
	54%	16%	13%	14%	3%
Implementation	8%	19%	22%	26%	25%
	24%	10%	23%	30%	13%

Communication

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
Infrastructure	13%	26%	20%	22%	19%
	35%	22%	15%	23%	5%
Codification	11%	27%	27%	21%	14%
	23%	28%	30%	17%	3%
Personalization	7%	11%	13%	38%	31%
	14%	25%	24%	29%	8%

Outlook

- Evaluate the data (maybe even based on my notebook…)
  - I shared my data, waiting for the US data to come back
  - Compare results between the communities
- Write and publish a high impact paper!
  - Maybe we‘ll see again at RSECon UK ;)
  - Kick off for our newly established Secure Software Engineering research group
- Start more targeted research towards Secure Research Software Engieering
  - You are my guineapigs and I‘m very thankful for your collaboration
  - If you are interested and have ideas, don‘t hesitate to contact me
  - One important goal is to make our RSE guidelines more secure

Impressum

Thema:	**First Results on the Security Culture Survey** de-RSE‘24 Conference, 05. – 07.03.2024, Uni Würzburg
Datum:	2024-03-05
Autor:	Michael Meinel
Institut:	DLR Institut für Softwaretechnologie
Bildquellen:	Alle Bilder „DLR (CC BY-NC-ND 3.0)“, sofern nicht anders angegeben