Promoting Software Engineering Best Practices for Research Software Engineers at Virginia Tech

Center for Digital Research and Scholarship, University Libraries Collaborative Research Grant

Chris Brown (PI)
Assistant Professor
Computer Science
Virginia Tech
dcbrown@vt.edu

Jonathan Bradley (Co-PI)
Assistant Director of Learning Environments and Innovative Technologies
University Libraries
Virginia Tech
bradley1@vt.edu

1 Introduction

Software is increasingly used in science and engineering to support discovery and problem-solving. These scripts and systems allow researchers to store and analyze datasets, perform complex calculations, and develop models and simulations for scientific investigations [15]. Prior work suggests at least 90% of researchers rely on software for their work, with most claiming that research progress would not be possible or would require considerably more effort to conduct their research without it [6].

Software engineering (SE), or the processes, methods, and tools to support the development and maintenance of software [11], is crucial for producing high quality applications. To support SE tasks, tools and processes informed by empirical SE research have been introduced and evaluated. These efforts provide evidence promoting practices to support the design, implementation, testing, innovation, and maintenance of software to help programmers complete software development tasks more effectively and efficiently [3].

Problem Statement. Despite evidence supporting SE practices, prior work shows programmers often avoid useful development behaviors in practice (*e.g.,* [8, 12]). Prior work suggests this software-research "crisis" can be detrimental to software [5]. Further, *research software engineers*—individuals who write code to support their research, with or without a formal Computer Science or SE background—rarely adopt useful practices shown to benefit software development, such as defined SE processes [4], version control systems [7], documentation [13], and testing [9]. Studies show the primary pain points research software engineers face are technical problems related to developing, maintaining, testing, and debugging code for research software [16]—all areas of empirical SE research.

Research software engineers often have limited knowledge of SE concepts [2], without time and opportunities to learn [1]. Further, access to SE research is also limited, inhibiting the adoption of useful practices for programmers [5]. To that end, this project seeks to increase awareness of beneficial SE practices for research software engineers. This leads to scientists spending "*far too much time wrestling with software, instead of doing research*" [17]. Further, avoiding SE best practices can have severe consequences specific to research software, including security vulnerabilities [10] or incorrectly reported research findings [14].

Project Overview. To this end, the proposed work aims to build on preliminary work to be conducted across the Fall and Spring semester to promote evidence-based SE practices among research software engineers across departments at Virginia Tech. Through this initial project, we will: (a) explore challenges research software engineers face with developing and maintaining research software; (b) conduct participatory design workshops to motivate the design of automated systems to promote evidence-based practices in research software development; (c) implement a preliminary large language model (LLM)-based system based on the co-designed prototypes; and (d) conduct a preliminary user study of the developed system. The proposed work aims to extend this initial effort by disseminating our findings to research software engineers at Virginia Tech. The main research question we aim to answer is: *RQ: How do research software engineers perceive an automated system to promote evidence-based SE practices?*

We will answer this question by conducting a public workshop hosted by University Libraries at Virginia Tech to share our findings. University Libraries has longstanding partnerships and infrastructure to conduct high-impact learning experiences for members of the Virginia Tech community.[1] The PI also has expertise in exploring methods to improve the behavior, productivity,

[1] `https://lib.vt.edu/research-teaching/instruction1.html`

and decision-making of software engineers (see PI Brown's CV). Co-PI Bradley is the Assistant Director of Learning Environments and Innovative Technologies for the University Libraries, overseeing six emerging technologies spaces in Newman Library. The goal of this work is to support research software development—promoting evidence-based practices and novel tools to help reduce frustration and enhance the quality of code used for scientific discovery and innovation.

2 Proposed Project

Workshop. The proposed work will implement a one-day workshop through University Libraries for research software engineers at Virginia Tech. The workshop will be offered in-person on campus to researchers who write code to support their work. We send emails to departmental and other relevant listervs to recruit at least 25 attendees from different fields and varying experience levels (*i.e.,*undergraduate and graduate students, postdocs, and faculty) across campus. The workshop will be organized into multiple interactive sessions to: 1) discuss challenges with research software development; 2) disseminate our findings on obstacles found through our prior work; 3) demonstrate our tool's ability to promote evidence-based SE practices; and 4) collect feedback on the tool. We will leverage University Libraries to obtain space and technical resources to conduct the workshop activities (*i.e.,* meeting room, A/V capabilities, etc.) and support the promotion of our workshop across campus. Funding from the collaborative research grant will also go to support the organization and implementation of the workshop (see Table 1).

Deliverables. The first project deliverable will be the completion of the *preliminary work*, conducting two participatory design workshops (Fall 2024) and producing a tool to recommend evidence-based SE practices to research software engineers (Spring 2025). The next milestone is an *interactive workshop* to present our findings and demonstrate the capabilities of our LLM-based system to support research software development (late Spring 2025). The full-day workshop will consist of talks from the PI and graduate student researcher, and discussions among workshop participants. The Co-PI will help organize the workshop. Success will be measured through the number of participants to enroll in the workshop and an exit survey. The last deliverables will involve *sharing our findings* in research, industry, and educational contexts through various means to promote empirically supported SE practices in research software development (Summer 2025).

Based on feedback from workshop participants, we envision future work to enhance our tool and develop new systems to provide infrastructure to promote evidence-based practices in research SE. We will pursue external funding opportunities (*i.e.,* National Science Foundation, Better Scientific Software (BSSw) Fellowship, and Sloan Foundation solicitations). We will also submit our findings from the preliminary work and workshop activities for publication at SE, human-computer interaction, and scientific software-focused academic venues (*i.e.,* Foundations of Software Engineering (FSE), International Conference on Software Engineering (ICSE), Computer-Supported Cooperative Work (CSCW), Research Software Engineering Conference (RSECon), etc.).

Table 1: Budget Breakdown

Item	**Amount**	**Justification**
Graduate Research Assistantship	$6,769	25% graduate research assistantship (GRA)
Travel	$2,100	Travel for PI, Co-PI, and graduate student to disseminate findings
Workshop Materials and Supplies	$1,131	Materials, food and refreshments, videographer/photographer, etc.
Total:	**$10,000**	

IDENTIFYING INFORMATION:

NAME: Brown, Chris

ORCID iD: https://orcid.org/0000-0002-6036-4733

POSITION TITLE: Assistant Professor

PRIMARY ORGANIZATION AND LOCATION: Virginia Tech, Blacksburg, Virginia, United States

Professional Preparation:

ORGANIZATION AND LOCATION	DEGREE (if applicable)	RECEIPT DATE	FIELD OF STUDY
North Carolina State University, Raleigh, North Carolina, United States	PHD	04/2021	Computer Science
North Carolina State University, Raleigh, North Carolina, United States	MS	05/2017	Computer Science
Duke University, Durham, North Carolina, United States	BS	09/2013	Computer Science

Appointments and Positions

2021 - present	Assistant Professor, Virginia Tech, College of Engineering, Blacksburg, Virginia, United States
2020 - 2020	Instructor of Record, North Carolina State University, College of Engineering, Raleigh, North Carolina, United States
2017 - 2017	Quality Engineering Intern, Red Hat, Raleigh, North Carolina, United States
2017 - 2017	Quality Engineering Intern, Red Hat, Raleigh, NC, US
2016 - 2021	Graduate Research Assistant, North Carolina State University, College of Engineering, Raleigh, North Carolina, United States
2016 - 2016	Software Quality Engineer Intern, Blackbaud, Charleston, South Carolina, United States
2015 - 2016	Graduate Teaching Assistant, North Carolina State University, College of Engineering, Raleigh, North Carolina, United States
2013 - 2015	Python Developer, Bank of America, Charlotte, North Carolina, United States

Products

Products Most Closely Related to the Proposed Project

1. Franke Lucas, Liang Huayu, Brantly Aaron, Davis James C, Brown Chris. A First Look at the General Data Protection Regulation (GDPR) in Open-Source Software. 2024.
2. Brown C, Parnin C. Understanding the impact of GitHub suggested changes on recommendations between developers. Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering; 08 1 20; Virtual Event USA. New York, NY, USA: ACM; c2020. Available from: https://dl.acm.org/doi/10.1145/3368089.3409722 DOI: 10.1145/3368089.3409722

3. Khalid S, Brown C. Software Engineering Approaches Adopted By Blockchain Developers. 2023 Tenth International Conference on Software Defined Systems (SDS). 2023 Tenth International Conference on Software Defined Systems (SDS); ; San Antonio, TX, USA. IEEE; c2023. Available from: https://ieeexplore.ieee.org/document/10329007/ DOI: 10.1109/SDS59856.2023.10329007
4. Brown C, Parnin C. Nudging Students Toward Better Software Engineering Behaviors. 2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE). 2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE); ; Madrid, Spain. IEEE; c2021. Available from: https://ieeexplore.ieee.org/document/9474399/ DOI: 10.1109/BotSE52550.2021.00010
5. Brown C, Parnin C. Comparing Different Developer Behavior Recommendation Styles. Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. ICSE '20: 42nd International Conference on Software Engineering; 27 0 20; Seoul Republic of Korea. New York, NY, USA: ACM; c2020. Available from: https://dl.acm.org/doi/10.1145/3387940.3391481 DOI: 10.1145/3387940.3391481

Other Significant Products, Whether or Not Related to the Proposed Project

1. Haroon Sabaat, Brown Chris, Gulzar Muhammad Ali. DeSQL: Interactive Debugging of SQL in Data-Intensive Scalable Computing. Foundations of Software Engineering; 2024; New York, NY, USA: Association for Computing Machinery; c2024.
2. Behroozi Mahnaz, Parnin Chris, Brown Chris. Asynchronous technical interviews: Reducing the effect of supervised think-aloud on communication ability. Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering; 2022; ACM; c2022.
3. Minkyuk Ko, Dibyendu Brinto Bose, Hemayet Ahmed Chowdhury, Mohammed Seyam, Chris Brown. Exploring the Barriers and Factors that Influence Debugger Usage for Students. Visual Languages and Human Centric Computing; 2023; c2023.
4. Palvannan N, Brown C. Suggestion Bot: Analyzing the Impact of Automated Suggested Changes on Code Reviews. 2023 IEEE/ACM 5th International Workshop on Bots in Software Engineering (BotSE). 2023 IEEE/ACM 5th International Workshop on Bots in Software Engineering (BotSE); ; Melbourne, Australia. IEEE; c2023. Available from: https://ieeexplore.ieee.org/document/10190399/ DOI: 10.1109/BotSE59190.2023.00015
5. Anjum Haque M, Ahmad W, Lourentzou I, Brown C. FixEval: Execution-based Evaluation of Program Fixes for Programming Problems. 2023 IEEE/ACM International Workshop on Automated Program Repair (APR). 2023 IEEE/ACM International Workshop on Automated Program Repair (APR); ; Melbourne, Australia. IEEE; c2023. Available from: https://ieeexplore.ieee.org/document/10189234/ DOI: 10.1109/APR59189.2023.00009

Synergistic Activities

1. Program committee member and reviewer for software engineering and HCI-related academic workshops and conferences, such as Computer-Supported and Cooperative Work (CSCW) 2024.
2. Reviewer for software engineering-related academic journals, such as the IEEE Software Special Issue on Developing your Software Engineering Career.

3. As an invited participant to the NII Shonan Meeting on Software Developer Diversity and Inclusion (SDDI), I collaborated with other researchers to discuss and develop plans to conduct cutting edge diversity and inclusion research and broaden the participation of underrepresented groups in software engineering.
4. As an invited speaker for the It Will Never Work in Theory session at the Strange Loop 2022, I presented my research on making effective recommendations for development tools to an audience of software practitioners at the industry-focused conference.
5. As a faculty mentor for Virginia Tech Multicultural Academic Opportunities Program (MAOP) Summer Research Internship for Summer 2023, I advised three undergraduate students from minority backgrounds for a 10-week program to gain research experience conducting innovative and impactful computing research.

Certification:

When the individual signs the certification on behalf of themselves, they are certifying that the information is current, accurate, and complete. This includes, but is not limited to, information related to domestic and foreign appointments and positions. Misrepresentations and/or omissions may be subject to prosecution and liability pursuant to, but not limited to, 18 U.S.C. §§ 287, 1001, 1031 and 31 U.S.C. §§ 3729-3733 and 3802.

Certified by Brown, Chris in SciENcv on 2024-03-05 14:01:02

Jonathan Bradley's work involves everything from makerspace creation using 3D printers and other manufacturing technologies, to microelectronics work as part of the ongoing Smart Commons project, to web development technologies, to virtual reality and augmented reality development. He is the Assistant Director of Learning Environments and Innovative Technologies for the University Libraries and oversees the 6 emerging technologies spaces in Newman Library.

References

[1] J. Carver. Urssi conceptualization survey results. *US Research Software Sustainability Institute (URSSI)*, 2019. https://urssi.us/blog/2019/05/20/urssi-conceptualization-survey-results/.

[2] J. Carver, D. Heaton, L. Hochstein, and R. Bartlett. Self-perceptions about software engineering: A survey of scientists and engineers. *Computing in Science & Engineering*, 15(1):7–11, 2013.

[3] P. Devanbu, T. Zimmermann, and C. Bird. Belief & evidence in empirical software engineering. In *Proceedings of the 38th international conference on software engineering*, pages 108–119, 2016.

[4] N. U. Eisty, G. K. Thiruvathukal, and J. C. Carver. Use of software process in research software development: A survey. In *Proceedings of the 23rd International Conference on Evaluation and Assessment in Software Engineering*, EASE '19, page 276–282, New York, NY, USA, 2019. Association for Computing Machinery.

[5] R. L. Glass. The software-research crisis. *IEEE Software*, 11(6):42–47, 1994.

[6] S. Hettrick. softwaresaved/software_in_research_survey_2014: Software in research survey, Feb. 2018.

[7] C. Jay, R. Sanyour, and R. Haines. "not everyone can use git": Research software engineers' recommendations for scientist-centred software support (and what researchers really think of them). *Journal of Open Research Software*, June 2016.

[8] B. Johnson, Y. Song, E. Murphy-Hill, and R. Bowdidge. Why don't software developers use static analysis tools to find bugs? In *2013 35th International Conference on Software Engineering (ICSE)*, pages 672–681. IEEE, 2013.

[9] D. Kelly and R. Sanders. The challenge of testing scientific software. In *Proceedings of the 3rd annual conference of the Association for Software Testing (CAST 2008: Beyond the Boundaries)*, pages 30–36. Citeseer, 2008.

[10] R. Milewicz, J. Carver, S. Grayson, and T. Atkison. A secure future for open-source computational science and engineering. *Computing in Science & Engineering*, 24(4):65–69, 2022.

[11] R. S. Pressman. *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[12] J. Smith, L. N. Q. Do, and E. Murphy-Hill. Why can't johnny fix vulnerabilities: A usability evaluation of static analysis tools for security. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, pages 221–238, 2020.

[13] S. Smith, T. Jegatheesan, and D. Kelly. Advantages, disadvantages and misunderstandings about document driven design for scientific software. In *2016 Fourth International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering (SE-HPCCSE)*, pages 41–48. IEEE, 2016.

[14] D. A. Soergel. Rampant software errors undermine scientific results. *F1000Research*, 3, 2014.

[15] D. Soulé. Programming languages for scientific research. *LinkedIn Pulse*, 2023. https://www.linkedin.com/pulse/programming-languages-scientific-research-damien-soul%C3%A9/.

[16] I. Wiese, I. Polato, and G. Pinto. Naming the pain in developing scientific software. *IEEE Software*, 37(4):75–82, 2019.

[17] G. Wilson. Software carpentry: getting scientists to write better code by making them more productive. *Computing in science & engineering*, 8(6):66–69, 2006.