Note: This cover sheet is not to exceed one page.

ALFRED P. SLOAN FOUNDATION

sloan.org | proposal guidelines

PROPOSAL COVER SHEET

PROJECT TITLE: Designing and Implementing Automated Solutions to Promote Software Engineering Best Practices in Research Software Development

Principal Investigator

Name Chris Brown
Title Assistant Professor
Address 4105 Gilbert, 220 Gilbert St, Blacksburg, VA
Phone 540-231-9176
Email dcbrown@vt.edu

Grantee Organization: Virginia Tech
Amount Requested: $140,173
Requested Start Date: 05/01/2024
Requested End Date: 08/31/2025
Project URL: N/A

Co-Principal InvestigatorS/Co-Project LeadS

N/A

Project Goal

The project aims to improve the development and maintenance of research software by designing, implementing, and evaluating a novel sociotechnical system to recommend empirically verified software engineering (SE) behaviors to research software engineers.

Objectives

The proposed work aims to answer the following research questions: **(RQ1)** What design affordances are needed to overcome development challenges from the perspective of research software engineers? **(RQ2)** How do various modes of interaction affect the awareness of SE best practices among research software engineers?

Proposed Activities

The proposed research activities include: two participatory design workshops (one in-person and one virtual) to understand challenges research software engineers face and co-design technical solutions to increase awareness of SE best practices; the development of the devised solutions in a multimodal interactive system; and a preliminary user study to evaluate the usability and the effectiveness of the implemented system.

Expected Products

The research findings will be published as research papers in academic conferences, workshops, and journals related to SE and human-computer interaction. We will also disseminate our results to software engineering and research software-focused venues and organizations. The tangible results from the foundation support will include an interactive system to increase access and awareness of software engineering best practices validated by empirical evidence to research software engineers. The developed system will be made open source and publicly available in a project repository.

Expected Outcomes

The proposed work will incur new knowledge on research software engineers' perspectives of challenges and solutions to combat identified problems in research software development, and provide insight into the design of interactive automated systems to improve SE practices to support productive and robust research software development.

1 What is the research question and why is it important?

Software is increasingly critical for discovery and problem-solving across science and engineering. The use of software allows researchers to efficiently store and analyze large datasets, perform complex calculations, and develop models and simulations across domains for scientific investigation [56]. For example, the Python programming language has been shown to "push the boundaries" of research in scientific disciplines such as biology, bioinformatics, chemistry, physics, astronomy, and earth sciences [59]. Over 90% of researchers rely on software for their work, with most claiming it would not be practical or require considerably more effort to conduct their research without it [24].

Software engineering (SE), or the processes, methods, and tools to support the development and maintenance of software [43], is crucial for producing high quality applications. However, SE is a complex activity that relies on the efforts of programmers to design, implement, test, maintain, and innovate software programs. To support SE tasks, tools and processes informed by researchers and practitioners have been introduced to improve the development and maintenance of software and help developers complete programming tasks more effectively and efficiently [18]. However, despite empirical evidence supporting the benefits of SE practices, prior work shows programmers often avoid useful development behaviors (*e.g.,* [26, 53])—and suggests this software-research "crisis" can be detrimental to software [20].

Similarly, prior work suggests research software engineers rarely adopt useful practices shown to benefit software development, such as defined SE processes [19], version control systems [25], documentation [54], and testing [29]. Moreover, research software engineers report the primary pain points they face are technical problems related to developing, maintaining, testing, and debugging code for research software [62]. This leads to scientists spending "*far too much time wrestling with software, instead of doing research*" [63]. Further, avoiding SE best practices can have severe consequences in research software, including security vulnerabilities [38] or incorrectly reported research findings [55].

Research software engineers often have limited knowledge [16] and lack time and opportunities [15] to learn SE concepts. Further, access to software engineering research is also limited, inhibiting the adoption of useful practices for programmers [20]. To that end, this project seeks to increase awareness of beneficial SE practices for research software engineers. The proposed work aims to answer the following research questions:

RQ1: What design affordances are needed to overcome development challenges from the perspective of research software engineers?

RQ2: How do various modes of interaction affect the awareness of SE best practices among research software engineers?

To answer these research questions, the proposed work aims to design and implement an automated system to increase awareness of empirically verified SE best practices for research software engineers—individuals who "regularly use expertise in programming to advance research" [3]. The efforts to answer RQ1 will involve organizing two participatory design workshops to uncover challenges and co-design sociotechnical solutions to combat problems research software engineers face developing and maintaining research software. To answer RQ2, we will implement the proposed solution devised by workshop participants in an automated system. Further, we aim to conduct a preliminary evaluation to gain feedback on the system and understand effective modes of interaction for improving the adoption of SE best practices among research software engineers.

This work aims to mitigate the "software chasm" [30] between software engineering and scientific computing by designing and implementing a novel system to increase awareness of empirically verified best practices for software development in research software engineering communities. Through this system, we aim to promote the health of computational research projects and support research software engineers. The successful implementation of this project will contribute scientist-designed solutions for effective communication of empirically verified SE practices and research findings with multimodal interactions between academic experts and research software developers–with the goal of promoting good SE practices for research software engineers to enhance the development and quality of research software.

2 What is the state of the research on this question

Researchers have outlined challenges in research software engineering. Weise and colleagues outline three major types of challenges in research software community: technical, social, and scientific—noting technical-related challenges constitute over 70% of problems [62]. Existing literature has surveyed research software engineers to report challenges faced in the development of research software [15, 24, 16, 22, 40, 17]. Previous research has also explored challenges research software engineers face in specific phases of software development, such as requirements [33], design [44], and testing [28, 29]. Segal posits research software engineers often adopt inappropriate development models, leading to challenges with establishing requirements, testing, and code quality [50]. We aim to extend this work by understanding challenges research software engineers face and designing solutions to combat these problems by increasing awareness of beneficial behaviors in research software communities.

Prior work also offers solutions to improve research SE. Nanthaamornphong et al. found test-driven development—the process of writing test cases before writing code—enhanced the quality of research software [39]. Similarly, Pitt-Francis et al. show agile methodologies benefit computational biology software [42]. Existing systems have been shown to support research software development practices. For instance, research online platforms such as mailing lists [65], social Q&A websites (*i.e.,* Stack Exchange[1]) [60], and social media (*i.e.,* X, formerly Twitter[2]) [51], are valuable for curating knowledge and forming communities of practice among scientific data scientists. While these platforms have been shown to enhance software development in research contexts, crowd-sourced

[1] https://stackexchange.com/

[2] https://twitter.com/

responses lack verifiability and empirical evidence. For example, prior work shows responses on Stack Overflow, a programming-related Q&A website in the Stack Exchange network, can be unreliable [66]. The proposed work aims to further investigate solutions to improve research software engineering by providing a sociotechnical system to increase awareness of SE practices verified by empirical research methods.

PI's Prior Work Connecting SE research and practice is becoming increasingly critical as society grows more dependent on technology.[3] PI Brown's prior work has explored various methods to enhance the behavior and decision-making of software engineers in development contexts. For example, he has investigated the effectiveness of peer interactions—the process of software engineers learning about tools from in-person conversations with colleagues [8] and how developers perceive different styles of behavioral recommendations [10]. He also explored limitations in automated development tool recommendations, noting a lack of social context and interrupting developer workflows lead to ineffective recommendations [9]. For his doctoral work, Brown introduced *developer recommendation choice architectures*, a framework that uses concepts from nudge theory to improve the design of automated recommendations for useful developer behaviors [11, 14]. These studies will motivate the design of the system to increase awareness of useful SE behaviors for research software engineers.

3 What is the research project? What are its goals and its methodology?

The goal of the proposed work is to investigate the design and implementation of interactive systems to enhance the awareness and adoption of beneficial SE practices for research software engineers. This project will consist of two main research activities (RAs) corresponding with our research questions. A project overview is provided in Figure 1.

[3] https://www.computer.org/digital-library/magazines/so/cfp-connecting-research-practice

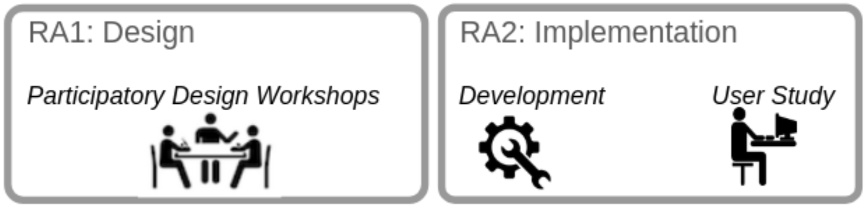

Figure 1: Overview of the proposed research

3.1 [RA1] Designing Solutions to Increase SE Best Practices in research software engineering

For our first research activity, we will conduct participatory design (PD) workshops involving research software engineers to inform the design of an automated bot to support the knowledge transfer of empirical software engineering findings to scientific practitioners. The participatory design approach will provide us with an opportunity to engage with researchers who maintain and develop research software without a background in SE to understand their experiences and challenges. Further, it allows individuals who would use an interactive system to enhance research software engineering to play a critical role in designing it [48]. The successful completion of this task will provide preliminary insights into collaboratively designed technical solutions to enhance awareness of beneficial SE practices and research in research SE contexts.

3.1.1 PD Workshops

Two full-day PD workshops will be offered to ten research software engineers actively developing software for scientific research. The workshops will involve a total of 20 participants (12 per workshop). We will run one in-person workshop on campus at Virginia Tech in Blacksburg, VA. The in-person workshop will consist of a mix of research software engineers from various departments at Virginia Tech and other regional schools in Virginia. Another virtual workshop will be conducted remotely for research software engineers from departments at any institution in the United States. The virtual workshop will provide a more diverse pool of participants and broaden the impact of our work beyond just our region. The workshop details will be advertised to institutional department listservs, research software engineering communities, and personal contacts to recruit in-person and virtual participants. We will obtain human subjects research approval on the PD workshop protocols from our university Institutional Review Board (IRB) before commencing this study.

Participants will be compensated for participating in the workshops. In the application process, we will ask potential participants to provide demographic information, an explanation of how they implement software for their research, and a brief description of their reason for participating in the workshop and how they can contribute to its success. If we receive more applications than available spots, the project team will employ purposive sampling and review the applications to choose candidates from diverse demographics who would benefit from the workshop and could provide useful input for automated solutions to enhance research software engineering. The virtual workshop will be held over Zoom. The workshops will be beneficial to the participants in raising awareness of challenges in research software engineering and brainstorming solutions to enhance the current practice.

3.1.2 Procedure

The PD workshops will consist of three phases for data collection correlating with the three-stage participatory design methodology [57]:

1. Initial Exploration To align participants in the context of the workshops, we will begin by having participants share their experiences developing and maintaining code for scientific research. This session will be semi-structured and moderated by a project team member. The moderator will facilitate follow-up group discussions to understand and reflect upon the experiences and identified challenges research software engineers face. In particular, we will ask participants to share their experiences in research software engineering, the challenges they face in development tasks for scientific programs, and problem-solving strategies used to overcome the discussed challenges. The initial exploration session will be recorded for the research team to further analyze retroactively.

2. Discovery Process After the initial exploration, participants will take part in a semi-structured session where research software engineers co-design solutions to overcome the described challenges and enhance their development tasks. In particular, participants will be asked to think about the ways in which interactive systems can be used to increase awareness of beneficial software engineering behaviors and relevant software engineering research findings to support development tasks. The goal will be to design a system to overcome the aforementioned challenges faced in the first phase based on participants' experiences and perspectives. We will also inform participants to consider multiple modes of interaction for the system. The moderator will set the goals of the group, observe and maintain communication between participants, be in charge of the time, prevent disruptive and rude participants, and manage co-design activities and materials. Examples of collaborative PD strategies we plan include sketching, note-taking, role-playing, and polling or voting mechanisms to get immediate real-time feedback from participants on various topics. The initial part of the discovery process will be group-based brainstorming where participants iteratively design and discuss ideas for the design of a system. For the virtual workshop the moderator will manage virtual functionalities to facilitate co-design tactics on Zoom (*i.e.,* virtual whiteboards). All materials and notes will be shared with the participants and collected by the research team for further analysis. The generated ideas will be categorized and serve as a basis for the final stage.

3. Prototyping During the discovery process, participants will be tasked with brainstorming ideas for preliminary solution(s) to support research software engineers completing development tasks. The final stage of our PD method will be prototyping the co-designed solutions using various prototyping tools. We will provide mechanisms such as paper prototypes, sticky notes, or sketching tools for participants to prototype solutions during the in-person workshops. For the virtual and in-person workshops, we will provide collaborative prototyping tools for participants to use, such as Balsamiq[4] or Figma.[5] After the initial prototyping phase, we will conduct iterations of discovery and prototyping for participants to improve upon and refine their prototype designs. The final outcomes participants create can be in various formats, *i.e.,* storyboards, written descriptions, or mock-up prototypes, to present the design of interactive systems for promoting beneficial SE practices for research software engineers.

3.1.3 Data Collection and Analysis

We will record audio and video for all activities during the PD workshops and document—both collecting and taking pictures of—artifacts, prototypes, notes, etc. generated by participants. We will employ multiple methods to evaluate the devised solutions from the PD workshop participants. First, to conclude the workshops we will distribute an exit survey to debrief participants. The survey will collect general qualitative (*i.e.,* open-ended questions) and quantitative (*i.e.,* Likert-scale questions) feedback on participants' experiences in the workshop. The exit survey will also gather insights on the devised solutions to support research software engineering created during the PD sessions—collecting specific insights on the design of automated systems to support knowledge transfer during programming tasks. Multiple researchers will independently identify recurring themes and underlying values through a thematic analysis of comments made by participants in the exit survey and transcribed recordings of the PD activities.

3.2 [RA2] Implementing Solutions to Increase SE Best Practices in research software engineering

Using our findings from RA1, our second research activity will produce a sociotechnical system to transfer SE knowledge and findings to research software engineers. The goal of the system will be to increase awareness of scholarly findings regarding SE best practices

[4] `https://balsamiq.com/`
[5] `https://www.figma.com/`

in research development. In addition, we will evaluate the implemented system through a user study to gain feedback on the usability and effectiveness of the platform. This task will contribute a novel system that promotes beneficial SE behaviors for research software engineers and provide preliminary insights into the impact of different interaction modalities on the behavior of research software engineers.

3.2.1 System Development

Based on the identified challenges and the ideated results from the PD workshops, we plan to create an interactive system to notify research software engineers of beneficial development behaviors while completing development tasks. The research team will synthesize the prototyped outputs and recurring themes from participants in both workshops to inform the design of our system. The guidelines derived from the PD workshops will be considered to develop features for supporting research software engineers. To accomplish this, we have several design goals for our system.

Multi-modal Interaction. To effectively support the complexities of research software engineering, we envision implementing multiple modes of interactions in our system. Multimodal designs in user interfaces have been shown to enhance learning [64] and engagement [47] in learning environments. Further, we aim to understand which modes of interaction are most effective for promoting beneficial SE behaviors of research software engineers by comparing various mechanisms within our system. The modes of interaction will be derived from the findings from the PD workshop and motivated by prior work. For example, we anticipate integrating various forms of interaction in our system, such as synchronous (*i.e.,* live chat) and asynchronous (*i.e.,* email) forms of communication with experts in the SE domain.

AI-Powered. To automatically increase awareness of SE best practices in research software development, our tool will also be powered by a large language model (LLM) trained on SE research publications. Software engineers frequently seek help from LLMs (*i.e.*, ChatGPT[6]) for various SE tasks [5, 21, 61]. LLMs have also been shown to be effective for text summarization tasks [2]. This project will further explore LLM usage in SE by providing automated insight from scholarly materials reporting empirical evidence supporting best practices in software development. Similar platforms have been implemented to automatically synthesize research output in the sciences [49], medicine [45], and legal [46] domains. Our proposed work will apply similar concepts in the research software domain, using LLMs to summarize SE research findings and beneficial development practices for research software engineers.

These guidelines will be provided to participants in the PD workshop to guide their co-designing efforts. Specific details for the system development are available in Appendix E.1. Here, we provide examples of possible systems that could be suggested from participants in the PD workshop based on these guidelines: (1) An online platform for research software engineers to ask questions and receive feedback from SE experts, researchers, and AI agents; (2) An automated bot that responds to questions on existing programming-related Q&A websites, such as Stack Overflow; (3) A plugin for an integrated development environment (IDE) that recommends useful SE behaviors to research software engineers while they are actively writing code for research software.

3.2.2 Preliminary Evaluation

We will evaluate the usability of our system and elicit feedback from research software engineers through a user study. This preliminary study will follow empirical SE guidelines for conducting controlled tool experiments with human subjects [32].

Recruitment. We will recruit research software engineers from various departments at Virginia Tech to complete tasks and interact with our implemented system. Potential subjects will be recruited via emails to departmental listervs and personal contacts, and participants will be selected through convenience sampling based on availability. The

[6] `https://chat.openai.com`

study will receive IRB approval before starting participant recruitment. Individuals who agree to participate will be compensated for their time to complete the study.

Pre-Task Study Procedures. Before completing the study, we will collect informed consent from participants to complete the study and record the session. In addition, we will collect demographic information from participants via a brief questionnaire to gain insight into participants background and experiences with developing and maintaining research software. Participants will also be informed they can leave the study or decline any part of the protocol without penalty, and the total time to complete the study should take no more than one hour.

Task Design. Participants will complete software development and maintenance tasks with our system to observe its usability and effectiveness in supporting research software development activities. The study tasks will be specific to research SE and based on the identified challenges mentioned by participants in the exploration phase of the PD workshops (RA1). We will employ a think aloud protocol to gather participants' thoughts while using the system, which will be used for further analysis.

Debriefing. We will debrief participants after the study tasks to evaluate the usability of our system. We will integrate System Usability Scale (SUS) questionnaires [7] and collect open-ended feedback via post-surveys and semi-structured interviews to gain insight on the system and how it can be improved. In particular, we aim to understand the usefulness of the system, the quality of AI-generated content, and modes of communication through the system. We will use the collected feedback to iteratively refine our platform for future evaluations and usage. Participants will also be asked to speculate about desired features to increase the usability and effectiveness of the implemented system. The post-interviews will be recorded, transcribed, and qualitatively analyzed by the research team to derive themes regarding effective modes of interaction for improving SE practices for research software engineers and motivation for future tools and resources to support the development and maintenance of computational research software.

3.2.3 Data Collection and Analysis.

All study sessions will be recorded and transcribed by the research team. The recordings will be analyzed retroactively to evaluate participants' performance in completing the study tasks. We will also use thematic analysis techniques, such as open coding, to derive themes from the qualitative feedback provided in participants' think-aloud and debriefing. Our findings aim to provide insight into effective interaction methods and motivate the design of future tools and resources to support research software engineers.

4 Who are the key members of the research team?

PI Brown has investigated methods to improve the behavior, productivity, and decision-making of software engineers. He has expertise developing and evaluating development tools and bots to improve SE activities—such as recommending static analysis tools [9], improving code reviews [41], enhancing students' SE behaviors on coding projects [13], generating API specifications [58], and supporting code navigation [52]. In addition, he has extensive experience investigating various SE concepts and tools through empirical user studies [4, 10, 12, 23, 41]. For the proposed work, PI Brown will lead the efforts for the participatory design workshops (RA1), oversee the system prototype development and evaluation (RA2), recruit and supervise student researchers contributing to this project, disseminate findings, and ensure the project goals are met in a timely manner.

A **graduate research assistant** (**GRA**) will be responsible for leading the data collection and analysis for the PD workshops, the development of the proposed system, and the user study to obtain preliminary feedback on the system. The GRA will be a graduate student in Computer Science at Virginia Tech. They must have experience designing and implementing software using a high-level programming language. Additional desired skills include: familiarity with AI concepts, specifically collecting and training data for LLMs; and prior research experience. **Undergraduate research interns** will complete data analysis tasks with the GRA for the PD workshops and user study, and aid in the implementation and testing of the prototype system.

5 What is the work plan?

The estimated duration of this project is 16 months (*i.e.,* May 1, 2024–August 31, 2025). A projected timeline of the proposed research activities is presented below.

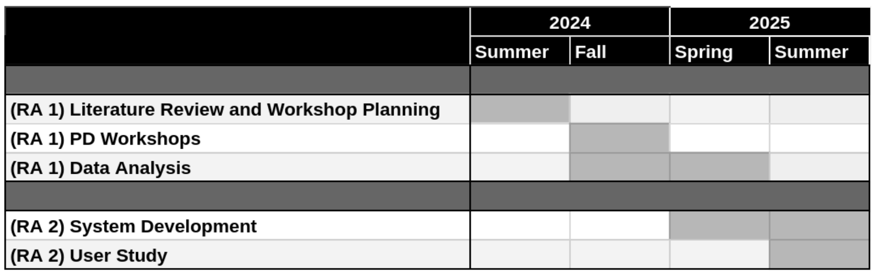

Figure 2: Timeline for the proposed research

The first project milestone will be the PD workshops (RA1) during the Fall 2024 semester. The success of this milestone will be measured by the co-designed output generated by workshop participants and feedback provided in the exit survey. Our next milestone will be the implementation of the knowledge transfer system. In January 2025, we will begin outlining requirements and devising preliminary architectural and user interface designs for the system based on findings from RA1. Throughout the following months, we will iteratively develop the system and train the model to be incorporated into our tool. The system will be developed using a high-level programming language (*i.e.,* Python, Java, etc.) and powered by the Llama LLM.[7] We aim to have the prototype complete by late April - early May, and will reserve time to test the system and make any necessary improvements. The code for the system will be stored in a public GitHub repository and deployed locally for the preliminary evaluation. The success of this milestone will be measured by feedback in the user study—scheduled to take place in Summer 2025. We will plan the evaluation and perform pilots in June to conduct the study in July.

[7] `https://llama.meta.com/`

6 What will be the output from the research project?

Our findings will be submitted for publication in relevant peer-reviewed academic journals, conferences, and workshops related to SE and HCI. Potential venues include Computer-Supported Cooperative Work (CSCW), Human Factors in Computing Systems (CHI), the International Conference on Software Engineering (ICSE), and Foundations of Software Engineering (FSE). These paper submissions will be led by the GRA in collaboration with undergraduate interns and the PI. We will also target research software-focused conferences, such as the Sustainable Scientific Software Conference (S3C), the Research Software Engineering Conference (RSECon), US Research Software Engineer Association (US-RSE) conference and working groups, and the Improving Scientific Software Conference. We will also make our findings publicly available via blog posts, technical reports, and contributions to research software-focused organizations (*i.e.,* US-RSE [3], Better Scientific Software,[8] and the Society of Research Software Engineering[9]).

The implemented system will be made open source and publicly available in a GitHub repository for researchers and developers to use and extend. We will also provide video tutorials and training materials to demonstrate the usage and capabilities of our system for users. Lastly, the study materials from our research tasks will be made available in a public repository, pending IRB approval, to support the replicability of our work. The success of this work will be constituted by the publication of research findings, the dissemination of findings in non-academic venues, the release of a public repository containing the source code and documentation for our system, and the productivity of undergraduate and graduate researchers (*i.e.,* completing degree milestones). The proposed work will advance knowledge on research software engineers' perspectives of challenges and solutions to combat problems in research software development and provide insight into the design of systems to enhance the SE behaviors of scientific programmers.

[8] `https://bssw.io/`
[9] `https://society-rse.org/`

7 What is the summary justification for the requested funding?

The total projected cost for the proposed work is $140,173 for a 16-month period (see Budget and Detailed Budget Justification). The budget requested will be used to support: summer and academic year salary as well as tuition, fees, and benefits for a graduate research assistant from the Department of Computer Science at Virginia Tech, who will work on the project; summer funding for an undergraduate research intern through the Multicultural Academic Opportunities Program at Virginia Tech (see Diversity, Equity, and Inclusion Plan); travel for the research team; compensation for participants who complete research activities; and materials and supplies to carry out the proposed research.

The funding from the Sloan Foundation will provide the resources needed to complete the proposed research and support the desired research output of this project. Namely, the grant will provide financial support for the researchers to complete the research activities, the publication and dissemination of research findings to the broader research community, and the design, development, and deployment of a co-designed sociotechnical system to enhance awareness of SE best practices among research software engineers.

8 What other sources of research support has the proposer applied for or have in hand to support the research team?

No other support has been requested for this project.

9 What is the status and output of current and/or previous Sloan grants?

PI Brown has no prior support from the Sloan Foundation.

Diversity, Equity, and Inclusion Plan

The PI is committed to promoting the diversity and inclusion of underrepresented individuals in computing throughout the completion of the proposed work. For this project, he will incorporate the following strategies to advance diversity, equity, and inclusion through the research activities and among the research team.

9.1 Research Activities

Research Methods. PI Brown will ensure the participatory design workshops (RA1) and prototype system user study (RA2) incorporate a diverse set of participants. The workshop recruitment process will involve a brief application process to collect potential participants' demographic information and experiences. In the application review process, we will target a diverse group of workshop participants based on gender, race and ethnicity, scientific domain, and research SE experience. The virtual workshop will also enroll participants from different institutions to gain further perspectives from research software engineers beyond the PI's university and locale. In addition, the moderator for the workshops will establish an inclusive and equitable environment—ensuring that all participants are able to contribute to the discussions and co-designing sessions..

Similarly, the PI will ensure the participant sample for the user study to evaluate the prototype co-designed system is diverse with regard to the race/ethnicity, gender, and experiences of participants. Diverse perspectives on the design and usability of the system will promote the design of inclusive solutions to improve SE practices, and ultimately product quality, for research software engineers from diverse backgrounds.

Dissemination. The research findings will be disseminated to various venues—shared in academic and practitioner-focused settings. We will also make targeted efforts to make our results publicly available for equitable access through diverse means (*i.e.*, blog posts, technical reports, public lectures, and social media).

9.2 Research Team

MAOP Summer Interns. The budget incorporates funding to support undergraduate student researchers through the Multicultural Academic Opportunities Program (MAOP) Summer Research Internship program. This program hosts an average of 40 interns from around the United States, primarily from underrepresented backgrounds, during the summer as they engage in research in a variety of fields. The program also provide GRE classes and a variety of professional development workshops for student participants to create a recruitment pipelines for STEM graduate programs from outstanding undergraduates from minority groups.

PI Brown has experience mentoring students through this program, advising three undergraduate students on various projects related to software engineering for Summer 2023. The SRI students spent 10 weeks working on a research project administered and advised by Dr. Brown, culminating in research posters at the Virginia Tech Summer Undergraduate Research Symposium. The students worked on projects spanning a variety of topics exploring the usability of debugging tools, Artificial Intelligence (AI) debugging techniques via tools such as ChatGPT, and challenges in video game development. PI Brown has budgeted funds to continue partnering with the MAOP program. He plans to recruit two undergraduate research interns from minority backgrounds through this program to contribute to the proposed work for Summer 2024 and Summer 2025.

Inclusive Work Environment. PI Brown identifies as an African American male, and he will recruit a diverse team to complete the research activities. He currently leads a diverse team of students made up of graduate and undergraduate students from varying genders and races/ethnicities.[10] The PI will recruit a diverse team to contribute to the execution of the research activities—conducting experiments and developing the prototype system. He will also work to foster an inclusive environment where students can feel comfortable actively engaging in the proposed research activities.

[10] `https://code-world-no-blanket.github.io/team.html`

Budget and Budget Justification

The project period is for a duration of 16 months, 05/01/2024 – 08/31/2025.

Personnel. PI Brown is a junior faculty at Virginia Tech and has a 9-month appointment with a base salary of $130,495 (AY), and requests 1.00 months of summer salary during year one, and 1.20 months, during year 2. The total salary requested each year is $14,499, and $17,399.

The PI is responsible for overall project execution at Virginia Tech as well as supervision of the students. Virginia Tech defines a year as 8/10/xxxx – 8/9/xxxx. An escalation factor of 5% is included and occurs December 1st each year.

Other Personnel. One graduate student from the Department of Computer Science will work on the project each year for an equivalent of 12 total months in year one, and 4 months in year two. The monthly stipend is $2,736, per the University's approved monthly stipend table, and based on a 20-hour workweek. The combined total student stipend budgeted each year is $32,832 and $10,944. An escalation factor of 5% is included and occurs August 15th each year. The graduate student will lead the efforts for the system development and support data collection and analysis for the participatory design workshops and tool evaluation user study.

An undergraduate student is also budgeted in the amount of $4,500 each year. The Multicultural Academic Opportunities Program (MAOP) at Virginia Tech sponsors a summer research internship program for underrepresented undergraduate students to gain research experience and participate in professional development and networking activities. Students will complete a 10-week internship and receive a $3,000 stipend, on-campus housing, and a meal plan. The total cost for the program is $9,000 per student. Faculty mentors are expected to contribute $4,500 per student, and the remaining costs are covered by MAOP and Virginia Tech.

Fringe Benefits. Fringe Benefits are calculated in accordance with Virginia Tech's federally negotiated fringe rate agreement which is available at `http://osp.vt.edu/resources/rates.html`.

FRINGE RATES	Through 6/30/24	On/After 7/1/24
SMR Faculty/Wage		
Employee	6.34%	6.65%
GRA	9.19%	9.83%

The fringe benefits total $3,936 in year one, and $2,233 in year two.

Annual negotiations with the Office of Naval Research (ONR) result in fixed rates for Employee Benefits covering the period July 1st - June 30th. Benefits include: Fee Waivers, Workman's Compensation, Retirement, Unemployment, FICA, Life Insurance, Hospitalization and Educational Leave. A copy of Virginia Tech's federally negotiated fringe rate agreement is available at: `http://osp.vt.edu/resources/rates.html`

Travel. Travel in the amount of $3,000 is budgeted each year. This will allow the PI and/or students to travel to conferences/workshops to present research findings. Anticipated budget items include conference registration, airfare, local transportation, hotel/lodging, and meals.

The University follows the Commonwealth of Virginia travel policy and procedures which provide for reimbursement of "reasonable" cost in connection with official travel. As a State agency the University is obliged to reimburse travel costs in conformance with State policy. Reimbursement in compliance with this policy is consistent with the requirements of Federal Acquisition Regulation (FAR) 31.3.

Other Direct Costs. $2,000 is budgeted in year one for resources needed to complete the proposed work, such as a laptop and monitor for student researchers to carry out the research activities.

Human subject payments in the amount of $2,400 in year one and $500 in year two is budgeted. This will allow for 24 participants at $100 each for the participatory design workshop and 25 participants at $20 each for the user study evaluation.

Tuition of $18,081 is budgeted for the entire project period, and based on the 2023-2024 tuition fees for an engineering student at the Blacksburg, Virginia campus. All sponsored program proposals that include graduate student stipends in the budget must also include tuition and technology and library fees for the same timeframe (AY) that the student(s) will be on GRA stipends. Academic year tuition plus technology, library, and engineering fee is budgeted for engineering students. The amount includes a 4.9% escalation factor each year, which occurs on August 15th. The Virginia Tech current estimating and cost procedures is available at: `https://osp.vt.edu/content/dam/osp_vt_edu/institutional-data/costestimating.pdf`.

Total Direct Costs $119,824

Indirect Costs. Indirect Cost Rates are fixed annually through agreement with the Office of Naval Research (ONR). The Commonwealth of Virginia legislative action obligates the University to recover indirect costs wherever possible. The sponsor rate of 20% TDC has been used, for a total of $24,948. A copy of the agreement is available at: `http://osp.vt.edu/resources/rates.html`

Total Direct and Indirect Costs $140,173

Alfred P Sloan Foundation
3-Year Budget Template

Principal Investigator: Chris Brown | **Department or Research Unit:** Computer Science

Organization: Virginia Tech | **Project Dates:** 5/1/2024 to 8/31/2025

	Title/Role	Monthly Base	Fringe Rate	Year 1 FTEs	Year 1 Funds Requested	Year 1 Non-Sloan Funds	Year 2 FTEs	Year 2 Funds Requested	Year 2 Non-Sloan Funds	Year 3 FTEs	Year 3 Funds Requested	Year 3 Non-Sloan Funds	Total Funds Requested	Total Non-Sloan Funds	TOTAL PROJECT COST
Merit/Cost of Living Increases: 0.0%															
A. SENIOR PERSONNEL: PI/PD, Co-PI'S, Faculty and Other Senior Associates															
FirstName LastName	**Title/Role**														
1. Chris Brown	Assistant Professor	$14,499	6.34%	1.00	$14,499	$0	0.00	$0	$0	0.00	$0	$0	**$14,499**	**$0**	**$14,499**
2. Chris Brown	Assistant Professor	$14,499	6.65%	0.00	$0	$0	1.20	$17,399	$0	0.00	$0	$0	**$17,399**	**$0**	**$17,399**
3.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
4.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
5.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
6.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
7.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
8.		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
TOTAL SENIOR PERSONNEL					**$14,499**	**$0**		**$17,399**	**$0**		**$0**	**$0**	**$31,898**	**$0**	**$31,898**
B. OTHER PERSONNEL (SHOW NUMBERS IN BOXES)															
1. 0 Post Doctoral Associates		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
2. 1 Graduate Students		$2,736	9.19%	12.00	$32,832	$0	0.00	$0	$0	0.00	$0	$0	**$32,832**	**$0**	**$32,832**
3. 1 Undergraduate Students		$1,500	0.00%	3.00	$4,500	$0	3.00	$4,500	$0	0.00	$0	$0	**$9,000**	**$0**	**$9,000**
4. 0 Secretarial - Clerical (If Charged Directly)		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
5. 1 Other (specify)	Graduate Student (Yea	$2,736	9.83%	0.00	$0	$0	4.00	$10,944	$0	0.00	$0	$0	**$10,944**	**$0**	**$10,944**
6. 0 Other (specify)		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
7. 0 Other (specify)		$0	0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
8. 0 Other (Not having the same base salary)			0.00%	0.00	$0	$0	0.00	$0	$0	0.00	$0	$0	**$0**	**$0**	**$0**
TOTAL OTHER PERSONNEL					**$37,332**	**$0**		**$15,444**	**$0**		**$0**	**$0**	**$52,776**	**$0**	**$52,776**
TOTAL SALARIES AND WAGES (A+B)					**$51,831**	**$0**		**$32,843**	**$0**		**$0**	**$0**	**$84,674**	**$0**	**$84,674**
C. FRINGE BENEFITS (AUTOMATICALLY CALCULATED BASED ON ENTERED RATES)															
1. Senior Personnel					**$919**	$0		**$1,157**	$0		**$0**	$0	**$2,076**	**$0**	**$2,076**
2. Other Personnel					**$3,017**	$0		**$1,076**	$0		**$0**	$0	**$4,093**	**$0**	**$4,093**
TOTAL FRINGE BENEFITS					**$3,936**	**$0**		**$2,233**	**$0**		**$0**	**$0**	**$6,169**	**$0**	**$6,169**
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A+B+C)					**$55,767**	**$0**		**$35,076**	**$0**		**$0**	**$0**	**$90,843**	**$0**	**$90,843**
D. TRAVEL															
1. Domestic (Incl. U.S. Possessions)					$3,000	$0		$3,000	$0		$0	$0	**$6,000**	**$0**	**$6,000**
2. Foreign					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
TOTAL TRAVEL					**$3,000**	**$0**		**$3,000**	**$0**		**$0**	**$0**	**$6,000**	**$0**	**$6,000**
E. PARTICIPANT SUPPORT COSTS															
1. Stipends					$2,400	$0		$500	$0		$0	$0	**$2,900**	**$0**	**$2,900**
2. Travel					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
3. Subsistence					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
4. Other					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
49 TOTAL NUMBER OF PARTICIPANTS					**$2,400**	**$0**		**$500**	**$0**		**$0**	**$0**	**$2,900**	**$0**	**$2,900**
F. OTHER DIRECT COSTS															
1. Materials and Supplies					$2,000	$0		$0	$0		$0	$0	**$2,000**	**$0**	**$2,000**
2. Publication Costs / Documentation / Dissemination					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
3. Honoraria (list # and amt. each)					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
4. Consultant Services					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
5. Computer Services					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
6. Equipment (specify)					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
7. Other					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
TOTAL OTHER DIRECT COSTS					**$2,000**	**$0**		**$0**	**$0**		**$0**	**$0**	**$2,000**	**$0**	**$2,000**
G. OVERHEAD-EXEMPT COSTS															
1. Tuition (requires exemption)					$16,000	$0		$2,081	$0		$0	$0	**$18,081**	**$0**	**$18,081**
2. Other					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
TOTAL OVERHEAD-EXEMPT COSTS					**$16,000**	**$0**		**$2,081**	**$0**		**$0**	**$0**	**$18,081**	**$0**	**$18,081**
H. TOTAL DIRECT COSTS (A THROUGH G)					**$79,167**	**$0**		**$40,657**	**$0**		**$0**	**$0**	**$119,824**	**$0**	**$119,824**
I. INDIRECT COSTS															
For grants $50,000 and under, the Foundation does not allow indirect costs.															
The Foundation does not allow indirect costs on tuition.															
Modified Total Direct Costs					**$63,167**	**$0**		**$38,576**	**$0**		**$0**	**$0**	**$101,743**	**$0**	**$101,743**
Overhead Rate:			20.00%		**$12,633**	**$0**		**$7,715**	**$0**		**$0**	**$0**	**$20,349**	**$0**	**$20,349**
J. SUBAWARDS															
Organization															
1.					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
2.					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
3.					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
4.					$0	$0		$0	$0		$0	$0	**$0**	**$0**	**$0**
TOTAL SUBAWARDS					**$0**	**$0**		**$0**	**$0**		**$0**	**$0**	**$0**	**$0**	**$0**
K. OVERHEAD ON SUBAWARD DIRECT COSTS					**$0**	**$0**		**$0**	**$0**		**$0**	**$0**	**$0**	**$0**	**$0**

Sum of section I. on each subaward									
L. TOTAL COSTS (H + I + J + K)	**$91,801**	**$0**	**$48,372**	**$0**	**$0**	**$0**	**$140,173**	**$0**	**$140,173**

Version 7.3, Last Modified: 10/31/2022

A List of Citations

References

[1] Empirical research papers: An overview. https://www.jmu.edu/uwc/_files/link-library/empirical/empirical_research_article_overview.pdf.

[2] T. Ahmed and P. Devanbu. Few-shot training llms for project-specific code-summarization. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–5, 2022.

[3] U. R. S. E. Association. What is an rse? https://us-rse.org/about/what-is-an-rse/.

[4] M. Behroozi, C. Parnin, and C. Brown. Asynchronous technical interviews: Reducing the effect of supervised think-aloud on communication ability. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, page 294–305, New York, NY, USA, 2022. Association for Computing Machinery.

[5] L. Belzner, T. Gabor, and M. Wirsing. Large language model assisted software engineering: prospects, challenges, and a case study. In *International Conference on Bridging the Gap between AI and Reality*, pages 355–374. Springer, 2023.

[6] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[7] J. Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[8] C. Brown, J. Middleton, E. Sharma, and E. Murphy-Hill. How software users recommend tools to each other. In *Visual Languages and Human-Centric Computing*, 2017.

[9] C. Brown and C. Parnin. Sorry to bother you: designing bots for effective recommendations. In *Proceedings of the 1st International Workshop on Bots in Software Engineering*, pages 54–58. IEEE Press, 2019.

[10] C. Brown and C. Parnin. Comparing different developer behavior recommendation styles. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20, page 78–85, New York, NY, USA, 2020. Association for Computing Machinery.

[11] C. Brown and C. Parnin. "Sorry to bother you again: Developer recommendation choice architectures for designing effective bots". In *Proceedings of the 2nd International Workshop on Bots in Software Engineering*, BotSE 2020, Seoul, South Korea, 2020. ACM.

[12] C. Brown and C. Parnin. *Understanding the Impact of GitHub Suggested Changes on Recommendations between Developers*, page 1065–1076. Association for Computing Machinery, New York, NY, USA, 2020.

[13] C. Brown and C. Parnin. Nudging students toward better software engineering behaviors. In *3rd IEEE/ACM International Workshop on Bots in Software Engineering, BotSE@ICSE 2021, Madrid, Spain, June 4, 2021*, pages 11–15. IEEE, 2021.

[14] D. C. Brown. *Digital nudges for encouraging developer behaviors*. PhD thesis, North Carolina State University, 2021.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] B. Grewal, W. Lu, S. Nadi, and C.-P. Bezemer. Analyzing developer use of chatgpt generated code in open source github projects. 2024.

[22] J. E. Hannay, C. MacLeod, J. Singer, H. P. Langtangen, D. Pfahl, and G. Wilson. How do scientists develop and use scientific software? In *2009 ICSE workshop on software engineering for computational science and engineering*, pages 1–8. Ieee, 2009.

[23] S. Haroon, C. Brown, and M. A. Gulzar. Desql: Interactive debugging of sql in data-intensive scalable computing. In *Proceedings of the ACM International Conference on the Foundations of Software Engineering*, FSE 2024, New York, NY, USA, 2024. Association for Computing Machinery.

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

[25] 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.

[26] 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.

[27] R. R. Johnson. *User-centered technology: A rhetorical theory for computers and other mundane artifacts*. SUNY press, 1998.

[28] U. Kanewala and J. M. Bieman. Testing scientific software: A systematic literature review. *Information and software technology*, 56(10):1219–1232, 2014.

[29] 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.

[30] D. F. Kelly. A software chasm: Software engineering and scientific computing. *IEEE software*, 24(6):120–119, 2007.

[31] G. Keswani, W. Bisen, H. Padwad, Y. Wankhedkar, S. Pandey, and A. Soni. Abstractive long text summarization using large language models. *International Journal of Intelligent Systems and Applications in Engineering*, 12(12s):160–168, 2024.

[32] A. J. Ko, T. D. LaToza, and M. M. Burnett. A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Software Engineering*, 20(1):110–141, 2015.

[33] K. Kreyman and D. L. Parnas. On documenting the requirements for computer programs based on models of physical phenomena. *SQRL Report*, 1, 2002.

[34] D. Leigh. Swot analysis. *Handbook of Improving Performance in the Workplace: Volumes 1-3*, pages 115–140, 2009.

[35] H. Lipmanowicz and K. McCandless. *The surprising power of liberating structures: Simple rules to unleash a culture of innovation*. Liberating Structures Press Seattle, WA, 2013.

[36] R. Luck. Dialogue in participatory design. *Design studies*, 24(6):523–535, 2003.

[37] A. Mahajani, V. Pandya, I. Maria, and D. Sharma. A comprehensive survey on extractive and abstractive techniques for text summarization. *Ambient Communications and Computer Systems: RACCCS-2018*, pages 339–351, 2019.

[38] 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.

[39] A. Nanthaamornphong and J. C. Carver. Test-driven development in scientific software: a survey. *Software Quality Journal*, 25:343–372, 2017.

[40] L. Nguyen-Hoan, S. Flint, and R. Sankaranarayana. A survey of scientific software development. In *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*, pages 1–10, 2010.

[41] N. Palvannan and C. Brown. "Suggestion bot: Analyzing the impact of automated suggested changes on code reviews". In *Proceedings of the 5th International Workshop on Bots in Software Engineering*, BotSE 2023, Melbourne, Austrailia, 2023. ACM.

[42] J. Pitt-Francis, M. O. Bernabeu, J. Cooper, A. Garny, L. Momtahan, J. Osborne, P. Pathmanathan, B. Rodriguez, J. P. Whiteley, and D. J. Gavaghan. Chaste: using agile programming techniques to develop computational biology software. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1878):3111–3136, 2008.

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

[44] F. Queiroz and R. Spitz. The lens of the lab: Design challenges in scientific software. *International Journal of Design Management and Professional Practice*, 10(3):17–45, 2016.

[45] Med-PaLM. `https://sites.research.google/med-palm/`.

[46] Rocket Matter. `https://www.rocketmatter.com/chatgpt/`.

[47] M. Sankey, D. Birch, and M. W. Gardiner. Engaging students through multimodal learning environments: The journey continues. *Proceedings of the 27th Australasian Society for Computers in Learning in Tertiary Education*, pages 852–863, 2010.

[48] D. Schuler and A. Namioka. *Participatory design: Principles and practices*. CRC Press, 1993.

[49] Scitodate. `https://www.scitodate.com/`.

[50] J. Segal. Some challenges facing software engineers developing software for scientists. In *2009 ICSE workshop on software engineering for computational science and engineering*, pages 9–14. IEEE, 2009.

[51] N. Shrestha, T. Barik, and C. Parnin. Remote, but connected: How# tidytuesday provides an online community of practice for data scientists. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–31, 2021.

[52] J. Smith, C. Brown, and E. Murphy-Hill. Flower: Navigating program flow in the ide. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 19–23, Oct 2017.

[53] 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.

[54] 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.

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

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

[57] C. Spinuzzi. The methodology of participatory design. *Technical communication*, 52(2):163–174, 2005.

[58] P. Sun, C. Brown, I. Beschastnikh, and K. T. Stolee. Mining specifications from documentation using a crowd. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 275–286, 2019.

[59] D. O. Tuama. The power of python in scientific fields. *Code Institute*, 2024. `https://codeinstitute.net/global/blog/the-power-of-python-in-scientific-fields/`.

[60] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov. How social q&a sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 342–354, 2014.

[61] J. White, S. Hays, Q. Fu, J. Spencer-Smith, and D. C. Schmidt. Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. *arXiv preprint arXiv:2303.07839*, 2023.

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

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

[64] C. Yu, H. Zhang, and L. B. Smith. Learning through multimodal interaction. In *Proceedings of the Fifth International Conference on Development and Learning (ICDL'06)*, 2006.

[65] A. Zagalsky, D. M. German, M.-A. Storey, C. G. Teshima, and G. Poo-Caamaño. How the r community creates and curates knowledge: an extended study of stack overflow and mailing lists. *Empirical Software Engineering*, 23:953–986, 2018.

[66] T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan, and M. Kim. Are code examples on an online q&a forum reliable? a study of api misuse on stack overflow. In *Proceedings of the 40th international conference on software engineering*, pages 886–896, 2018.

B Curricula Vitae

BIOGRAPHICAL SKETCH

Chris Brown	Email: dcbrown@vt.edu
Assistant Professor	Web: https://chbrown13.github.io/
Department of Computer Science	Phone: 540-231-9176
College of Engineering	Office: 4105 Gilbert
Virginia Tech	Address: 220 Gilbert Street, Blacksburg, VA 24061

(a) Education & Training

North Carolina State University	Raleigh, NC	Computer Science	Ph.D., 2015-2021
North Carolina State University	Raleigh, NC	Computer Science	MS, 2015-2017
Duke University	Durham, NC	Computer Science	B.S. *with distinction*, 2009-2013

(b) Research & Professional Experience

2021 – present	Assistant Professor, Virginia Tech
2020 – 2020	Instructor of Record, NC State
2018 – 2018	Quality Engineering Intern, Red Hat, Raleigh, NC
2017 – 2017	Quality Engineering Intern, Red Hat, Raleigh, NC
2016 – 2021	Graduate Research Assistant, NC State
2016 – 2016	Software Quality Engineering Intern, Blackbaud, Charleston, SC
2015 – 2016	Graduate Teaching Assistant, NC State
2013 – 2015	Python Developer, Bank of America, Charlotte, NC

(c) Selected Publications

Most closely related

1. Chris Brown and Chris Parnin, "Sorry to bother you: designing bots for effective recommendations", in *Proceedings of the 1st International Workshop on Bots in Software Engineering* (IEEE Press, 2019) pp. 54–58.
2. Chris Brown and Chris Parnin, "Sorry to bother you again: Developer recommendation choice architectures for designing effective bots", in *Proceedings of the 2nd International Workshop on Bots in Software Engineering* (ACM, 2020).
3. Chris Brown and Chris Parnin, "Comparing different developer behavior recommendation styles", in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20 (Association for Computing Machinery, New York, NY, USA, 2020) p. 78–85.
4. Chris Brown, Justin Middleton, Esha Sharma, and Emerson Murphy-Hill, "How software users recommend tools to each other", in *Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2017)* (IEEE Press, 2017).
5. Nivishree Palvannan and Chris Brown, "Suggestion bot: Analyzing the impact of automated suggested changes on code reviews ", in *Proceedings of the 5th International Workshop on Bots in Software Engineering* (IEEE Press, 2023).

Other significant publications

6. Sabaat Haroon, Chris Brown, and Muhammad Ali Gulzar, Desql: Interactive debugging of sql in data-intensive scalable computing, in *Proceedings of the ACM International Conference on the Foundations of Software Engineering*, FSE 2024 (Association for Computing Machinery, New York, NY, USA, 2024).
7. Mahnaz Behroozi, Chris Parnin, and Chris Brown, "Asynchronous technical interviews: Reducing the effect of supervised think-aloud on communication ability", in *Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022)* (ACM, 2022) **Distinguished Paper Award**.
8. Minhyuk Ko, Dibyendu Brinto Bose, Hemayet Ahmed Chowdhury, Mohammed Seyam, and Chris Brown, Exploring the barriers and factors that influence debugger usage for students, in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2023) pp. 168–172, **Best Short Paper, Honorable Mention**.
9. Tianjia Wang, Daniel Vargas Díaz, Chris Brown, and Yan Chen, Exploring the role of ai assistants in computer science education: Methods, implications, and instructor perspectives, in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2023) pp. 92–102.
10. Md Mahim Anjum Haque, Wasi Uddin Ahmad, Ismini Lourentzou, and Chris Brown, Fixeval: Execution-based evaluation of program fixes for programming problems, in *2023 IEEE/ACM International Workshop on Automated Program Repair (APR)* (2023) pp. 11–18.

(d) Recent Synergistic Activities

1. Reviewer for software engineering-related academic journals, such as IEEE Software: *Special Issue on Developing your Software Engineering Career*, Transactions on Software Engineering (TSE), Empirical Software Engineering (EMSE), IEEE Software, PeerJ Computer Science, and IEEE Software *Special Issue on Bots in Software Engineering*
2. Program committee member for various computing conferences and workshops, including:
   - Computer-Supported Cooperative Work (CSCW): 2024
   - International Workshop on Bots in Software Engineering (BotSE): 2023, 2022, 2021
   - Foundations of Software Engineering (FSE) (*Tutorials Track*): 2023
   - ACM Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium (*Birds of a Feather track*): 2023
   - International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE): 2022
3. As an invited speaker for the It Will Never Work in Theory workshop 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.
4. 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.
5. 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.

C Conflict of Interest / Sources of Bias

There are no conflicts of interest or sources of bias to report.

Financial Interests: N/A

Management/advisory affiliations: N/A

Paid consulting: N/A

Patents: N/A

D Empirical Research Methods

Empirical research aims to observe, measure, record, and analyze data using various qualitative or quantitative methods, with the goal of generating new knowledge about phenomena in humans or the natural world [1]. Our research activities consist of two empirical research methods, participatory design workshops and a controlled user study.

D.1 Participatory Design Workshops

Participatory design (PD) is a research methodology that places stakeholders at the center of the design process. PD as a research method is characterized by user-centered involvement focusing on the design of artifacts and systems [27]. This methodology relies on various research methods—including ethnographic observations, interviews, analysis of artifacts, and qualitative analysis—to iteratively construct the design and elicit feedback from participants who will represent stakeholders and users of the end system [57]. For the proposed work, the PD method was chosen based on its effectiveness in engaging stakeholders, increasing social inclusion and empathetic design, and extracting the tacit knowledge of participants which may be difficult to formalize and describe otherwise [48, 36]. Our participatory design workshops will leverage the three stage methodology proposed by Spinuzzi [57]: *Initial exploration of work*, *Discovery process*, and *Prototyping*. An overview of this process is described in Section 3.1.2. Here, we focus on the instruments and measures incorporated into the PD workshop settings to answer our research first question—**RQ1: What design affordances are needed to overcome development challenges from the perspective of research software engineers?**

Perspective of Research Software Engineers. To understand the perspectives of research software engineers, we will recruit individuals actively developing software to support their research activities. To identify participants, we will broadly distribute an email with a demographic questionnaire to departmental listervs and personal contacts. The questionnaire will collect the background experiences and identities of potential participants, as well as gain insight into preliminary perspectives in research software development. We will use purposive sampling to target a diverse group of 24 (12 in-person and 12 virtual) research software engineers from different demographic backgrounds to participate in the PD workshops. Participants (in-person and virtual) who successfully complete the workshop will be compensated $100 for their time.group

At the beginning of the workshop, we will provide an introductory overview of the workshop agenda, outline the expectations for the workshop, and conduct an icebreaker activity for participants to meet each other. The icebreaker will be a group activity for participants to get to know about each other. After the icebreaker, the first stage of the PD workshops will employ a qualitative techniques to gain further insight on their perspectives of research software development and specific challenges faced with constructing and maintaining research-focused software.

Development Challenges. To understand the development challenges of research software engineers, the initial exploration phase will utilize a semi-structured focus group to gain insights from participants. A member of the research team will be a moderator for the initial exploration, and we will leverage the 1-2-4-ALL[11] liberating structure to gain qualitative insights from participants. Liberating Structures are qualitative discussion techniques to support discussions and foster engagement in a group setting [35]. We will utilize 1-2-4-ALL as a data collection method to allow self-reflection and collaborative discussion, building toward consensus or shared understanding among participants. In this case, our goal is to gain an understanding of challenges research software developers face with SE-related tasks.

Participants will be given one overarching discussion question regarding challenges faced in research software development (*i.e., What are the main challenges you face developing and maintaining software related to your research?*). In this method, participants will first reflect on the discussion prompt and write down their notes individually for (1).

[11] https://www.liberatingstructures.com/1-1-2-4-all/

Then, participants will find a partner to discuss their thoughts with another individual (2). Next, two pairs of participants will join together to further discuss challenges and consolidate a list of challenges experienced in the group (4). Finally, the groups of four will report their outcomes to the entire group (ALL) for a general discussion. The moderator will ask follow-up questions and additional insights based on the responses from participants. This activity will last approximately one hour. In the virtual workshop, the smaller groups will be randomly assigned using the Zoom breakout room functionality. For the in-person workshop, the smaller groups will self-organize and disperse in the meeting room. The initial exploration phase will contribute a set of challenges reported by research software developers. The outcomes presented and proceeding discussion in this section will be recorded for our analysis and set the stage for the next phases of the workshop.

Design Affordances to Overcome Challenges Following the initial exploration phase, the PD workshop moderator will provide a brief presentation to transition to the discovery and prototyping stages. This presentation will provide a brief overview of empirical software engineering research and broad examples of beneficial behaviors verified by science [18]. We will also provide an overview of expectations, desired outcomes, and design goals for a desired automated system for the discovery and prototyping phases. The participants can self-organize as one large group or divide into smaller breakout groups for the discovery and prototyping stages. While we aim to produce an automated system in this work, participants will be notified that they are not limited to only brainstorming solutions that are automated systems but to produce general solutions, technical and non-technical, to support research software development.

The discovery process will offer strategies for participants to brainstorm ideas and designs for solutions to enhance the adoption of empirically verified development practice in research SE. The PI and event organizers will provide materials for co-designing and discover, including writing utensils and paper for in-person participants and shared documents and Zoom whiteboards for virtual participants. Systems for prototyping and design will also be available to use (*i.e.,* Balsamiq and Figma). For the prototyping phase, participants will act out the brainstormed solutions in simulated contexts. Examples of potential collaborative PD prototyping strategies include sketching, note-taking, role-playing, and polling. After participants prototype potential solutions, we will conduct another iteration of the discover process and prototyping stages of the workshop to improve upon initial solutions. To conclude the PD sessions, participants will demonstrate their proposed co-designed solutions for the entire group. The demos will be recorded, and all artifacts produced in the discovery processes will be collected for further analysis.

Exit Survey. To conclude the PD workshop, we will distribute an online survey to participants to collect feedback and gain insight into the collaboratively designed solutions. The survey will employ statistical analysis techniques to analyze the perception of the proposed solutions and use qualitative thematic analysis techniques [6]to understand open-ended comments from participants on the workshop and designed solutions. Sample exit survey questions are presented in Appendix F.

D.2 User Study

We aim to conduct a user study to evaluate the prototype system proposed by PD workshop participants. A description of the implementation of the system is provided in Appendix E.1. Our controlled user study will evaluate the usability and effectiveness of the implemented system. We will adhere to best practices for human subjects research for empirical SE [32].[12] The user study will allow us to gain feedback on the system within a simulated context and observe the usability and effectiveness of this system. High-level details are described in Section 3.2.2. This section describes how we will answer our second research question: **RQ2: How do various modes of interaction affect the awareness of SE best practices among research software engineers?**

[12] https://www2.sigsoft.org/EmpiricalStandards/docs/standards?standard=Experiments

Research Software Engineers. To evaluate our implemented prototype, we will recruit research software engineers to participate in our study. These participants will be separate from the individuals who participated in the PD workshops. We will recruit participants through emails to departmental listervs at Virginia Tech and regional universities and laboratories in Southwest Virginia. We anticipate completing the user studies in-person, as the system may only be locally deployed for the initial prototype. The email will include a brief demographic questionnaire to gain information about the background and experiences of participants and obtain their availability to complete the study. For this effort we will use convenience sampling to schedule participants based on their availability—however, we still aim to recruit a diverse sample of participants based on research domain, gender, race/ethnicity, and years of experience. We aim to recruit at least 25 participants to complete this study, and will compensation individuals who complete the study $20 for their efforts.

Various modes of interaction. To distill the output from the PD workshops into technical requirements for our system, we will analyze the output from workshop participants. First, the research team will review the demonstration recordings of the closing session of the two PD workshops. Each researcher will delineate a list of features incorporated into the demoed prototypes. After individual analyses, we will come together as a group to outline the most popular features reported by participants and cross-reference this list with the list of challenges reported in the initial exploration of the two workshops. We will create a mapping of which features were supported by workshop participants and which challenges reported by participants, if any, they address. Using this mapping, we will conduct a brief feasibility analysis based on current industry practices to create requirements for our technical system.[13]

First, we will define the scope and objectives of the overall tool, focusing on the specific problems identified by workshop participants. Next, we will conduct a lightweight

[13] `https://www.linkedin.com/advice/1/what-key-steps-criteria-conducting-feasibility`

literature review to investigate which challenges have been addressed in prior work and how. We will also analyze gray literature to understand which technical features proposed by PD workshop participants in their prototypes have been implemented in existing tools. Using this information, we will generate a Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis [34]. This will provide insight on the advantages and disadvantages for implementing each of the outlined features from our preliminary data exploration. We will also assess the technical feasibility of implementing the features, given the expertise and knowledge of the research team—in particular the GRA who will lead the development efforts. Finally, we will consider any alternative solutions, such as features or interaction methods that were not shared by participants in the PD workshops but may be beneficial to the goals of the platform. Using these analyses, we will recommend a set of features to include into the implemented prototype system.

SE best practices. We will utilize LLMs to synthesize development practices supported by empirical SE research. We aim to summarize these findings and convey them to research software developers. The description of how we will generate recommendations for SE best practices is described in the LLM Development paragraph of Appendix E.1.

Awareness. To understand the effectiveness of our system, we aim to conduct a within-subjects study for all participants to interact with the system. To control for external factors, the research team will construct two research development tasks. These tasks will be specific to research software development and based on the experiences and challenges reported by PD workshop participants. The task will require participants to complete a development task relevant to empirical SE research concepts and research software development. We will deploy two sample projects for this study on a GitHub repository, and the two tasks will be counter-balanced between participants with regard the order of task completion and our independent variable—whether the tool is used or not. While using simulated tasks creates a controlled experimental environment, it will limit our results as participants will be unfamiliar with the provided code and system. Our future work will aim to conduct a large-scale evaluation where research software developers use the tool in their normal working contexts to understand its effectiveness and collect additional feedback.

All study sessions will be audio, video, and screen-recorded. We will analyze various dependent variables, such as time to complete the task and correctness. Our hypothesis is that the system will improve the efficiency and effectiveness of research software development tasks, based on the increased awareness of SE best practices verified by science. We will conduct pilot studies with participants from our target sample to get feedback on these tasks and iteratively improve our study protocol. We will also observe the usage of the system, including which features and modes of interaction are used by participants. We also seek to gain additional insights on the system from research software developers. We will employ a think-aloud protocol, for participants to share their thoughts while completing the tasks both with and without the tool. We will also incorporate a debriefing session consisting of a brief semi-structured interview to gain feedback on the system and user experiences. Finally, we will give participants a post-survey to complete based on the System Usability scale introduced by Brooke et al. [7]. This will allow us to gain additional insights on the usability of the system within development contexts. Finally, the debriefing survey and interview will seek insight on ways to improve the system.

E Information Products

The proposed work anticipates producing the following information products to promote the transparency, reproducibility, and replicability of the proposed work:

E.1 Software

The proposed work aims to produce a prototype software system to enhance awareness and adoption of empirically validated SE practices in research software development.

System Implementation: The design goals for our system are provided in Section 3.2.1. The prototype system details and implementation will depend on the solutions collaboratively designed by participants in the PD workshops. Generally, the system will be written in a high-level programming language (*i.e.,* Python, Java, JavaScript, etc.), have multiple types of interaction from users, and be powered by a large language model.

LLM Development. Regarding the LLM development, we aim to build, train, and fine-tune large language models to synthesize scholarly research output from empirical SE work. We plan to use Llama as the base LLM for this system, due to it being free and open-source. We will leverage this model to perform natural language summarization to condense SE research findings. Additional LLM-based functionality, such as natural language text generation and question answering, will be incorporated depending on the system design and interaction mechanisms proposed by the participatory design workshop.

Our first step is to collect a representative dataset to train the model. The dataset will consist of recent research papers derived from scholarly libraries such as Google Scholar and the ACM and IEEE digital libraries. We will target research papers from SE-related journals and conferences, including but not limited to: the International Conference on Software Engineering (ICSE); Foundations of Software Engineering (FSE); International Conference on Cooperative Cooperative and Human Aspects of Software Engineering (CHASE); Automated Software Engineering (ASE); the International Symposium on Empirical Software Engineering and Measurement (ESEM); IEEE Software; Transactions on Software Engineering (TSE); Empirical Software Engineering (EMSE); and Transactions on Software Engineering and Methodology (TOSEM). Our dataset will incorporate a diverse set of research papers that use various empirical methods[14] to verify hypotheses and research questions related to software engineering in practice.

The GRA will formulate the dataset and conduct pre-processing steps. The dataset will consist of the titles, abstracts, and content of empirical SE research papers. Potential pre-processing tasks involve removing irrelevant metadata and sections, tokenizing the text, and removing unneeded words, special characters, and other irrelevant formatting. Once the dataset is pre-processed, we will begin training the model. The pre-processed data will be divided into training, validation, and test datasets with a balanced distribution. The Department of Computer Science at Virginia Tech also has GPU virtual machines[15] available for researchers to reserve, which we will leverage to complete the model training tasks for our system. We will use deep learning frameworks (*i.e.,* PyTorch[16] or TensorFlow[17]) to perform the training and experiment with different text summarization techniques [37]. We will perform preliminary evaluations of the summarized research output from our model, using manual evaluations with text summarization metrics—*i.e.,* Recall-Oriented Understudy for Gisting Evaluation (ROGUE) [31]—to evaluate accuracy of LLM-generated summaries.

System Availability: Upon the completion of the prototype, we will make the system publicly available. The source code and configurations for the system will be made open source and stored in a public GitHub repository under the MIT license.[18] This will allow future researchers or developers to modify, enhance, or extend the program for different purposes and different contexts. The GitHub repository will also include the dataset and

[14] `https://www2.sigsoft.org/EmpiricalStandards/docs/standards`
[15] `https://csrgpu.cs.vt.edu/`
[16] `https://pytorch.org/`
[17] `https://www.tensorflow.org/`
[18] `https://opensource.org/license/mit`

pre-processing and training scripts used for the AI models powering the system.

The computational environment and deployment strategies will depend on the outcome of the co-designed solutions in the PD workshop. For instance, a web application hosted online will have different development and execution processes compared to a plugin for an integrated development environment (*i.e.,* VS Code). We will share these details and provide specific instructions on how to download and run the system in the README file of the GitHub repository. We will also incorporate training materials, such as documentation explaining how to use the system and a demonstration video tutorial to showcase various functionalities. The GRA will maintain the system, and the PI will manage the long term storage and maintenance of the platform after the conclusion of the project.

E.2 Study Materials

The study materials for our research activities will be made publicly available with the publication of findings to support the replicability and reproducibility of the proposed work.

E.2.1 RA1: Participatory Design Workshops

- **Surveys:** The survey instruments for the PD workshops will be made available. This will include the demographic questionnaire used for the application process and the exit survey to elicit feedback on the workshop. We anticipate demographic questionnaire and exit survey responses will not be shared to protect the privacy of the workshop participants.

- **Exploration Script:** The initial exploration phase of the PD workshops will feature a semi-structured discussion investigating participants' experiences and challenges of research SE. We will make the script including a base set of discussion questions available.

- **Prototypes:** The final stage of the PD workshops involves participants working together to collaboratively design solutions to enhance empirically supported development practices in research SE. We will make the co-designed artifacts available to demonstrate the output from the PD sessions. All prototypes will be anonymized to protect the identity of participants.

E.2.2 RA2: User Study

- **Demographic Questionnaire:** The user study sessions will begin with a brief questionnaire to gain background information on participants. We will make the questionnaire publicly available.

- **Study Script:** The user study will consist of participants completing study tasks with our prototype system immediately followed by a brief debriefing to provide open-ended feedback on the system. Each session will be moderated by a member of the research team, who will use a script to outline the study protocol. The script will include a detailed description of the tasks for participants to complete and a set of interview questions to collect feedback from participants on the system.

- **Post-Survey:** Each user study session will conclude with a post-survey that will collect participants' thoughts on the usability of the system according to the SUS usability scale [7]. This survey will be made publicly available.

F Workshops, Conferences, or Other Large Meetings

The proposed work is requesting support for two participatory design workshops, one in-person at the Virginia Tech campus and one held virtually over Zoom. These workshops will be data collection mechanisms to gain insights on challenges research software engineers face and to co-design solutions to support increased awareness of SE best practices in research development contexts.

Workshop Agenda: The PD workshops will be half-day workshops for participants, lasting approximately n hours. A potential agenda is provided below for the PD workshops. Please note the time and activities are subject to change.

9:30-10:00 Welcome and Introductions

10:00-11:00 Initial Exploration

11:00-11:30 Discovery Process

11:30-12:00 Prototyping

12:00-12:15 Break

12:15-12:45 Discovery Process II

12:45-1:15 Prototyping II

1:15-1:45 Demo of Co-Designed Solutions

1:45-2:00 Concluding Remarks and Exit Survey

Workshop Invitees: Invitees will be research software developers at Virginia Tech (in-person) or at academic institutions in the United States (virtual). We aim to invite research developers from diverse backgrounds and experiences to participate in our workshops.

Letter of Invitation: Invitees will be invited to participate in the workshop via emails to departmental listservs and personal contacts. The email will be crafted to inform receivers about our workshop and recruit participants to join our research study by filling out a brief application. A sample template is provided below. Please note the content of the invitation is subject to changes.

Subject: Invitation to Participate in a Participatory Design Workshop for Research Software Engineers

Content: To Whom It May Concern:

Do you regularly use programming to support and advance your research? If so, the Code World, No Blanket research lab at Virginia Tech is excited to invite you to participate in a unique opportunity for research software engineers. We are organizing a workshop that aims to bring together research software engineers like yourself to collaboratively design solutions aimed at enhancing development practices supported by empirical research. Please find more details and a link to be considered for this opportunity below.

Event Details:

- Workshop Title: Participatory Design Workshop for Research Software Engineers
- Date: [Date]
- Time: [Start Time] - [End Time]
- Location: [On-Campus Location at Virginia Tech] or [Zoom Link]

Objective: The primary goal of this workshop is to leverage the collective knowledge and experience of research software engineers to design innovative solutions that address the challenges faced in software development, grounded in empirical research. As a participant in this study, you will utilize participatory design methods to generate actionable insights that can significantly enhance development practices in our field.

Why Participate? By participating in this workshop, you will have the opportunity to:

- Collaborate with fellow Research Software Engineers from diverse backgrounds.
- Gain insights into the latest empirical research findings in software engineering.
- Contribute your expertise towards designing practical solutions that can improve development practices.

Application:

To apply to participate in the workshop, please complete the application at the following link: [link].

Please note that participation in this workshop is limited to ensure productive and meaningful collaboration.

If you have any questions or require further information, please do not hesitate to contact [GRA contact] or Dr. Chris Brown at dcbrown@vt.edu.

Thanks,

[Name]

Invited speakers, presenters, and/or panel participants: N/A

Call for Papers: N/A

Anti-harassment Policy: The research team will enforce the following anti-harrassment policy to ensure all participants feel welcomed and encouraged to contribute to the PD workshops.

> This workshop is committed to providing a respectful and inclusive environment for all participants, regardless of gender, sexual orientation, race, ethnicity, religion, disability, age, or any other characteristic. We do not tolerate harassment of participants in any form. All participants are expected to treat each other with respect, value the views and opinions of others, and avoid engaging in behavior that can be considered inappropriate or harassment. If you are harassed or have other concerns, please alert the event organizers immediately or contact Dr. Chris Brown (dcbrown@vt.edu).

We will highlight this in the Introduction session of the workshop and the moderator will ensure the policy is upheld throughout the discussion sessions and co-design activities.

Participant Survey: Participants will be asked to complete a demographic survey to apply for participation in the PD workshops. Potential sample questions are provided below:

Demographic Survey

- Name
- Gender
- Race/Ethnicity
- Department
- University (virtual)
- Years of Programming Experience
- Programming Languages
- How do you use programming for your research? Please provide a brief explanation.
- How familiar are you with software engineering concepts?
- Do you have formal training in Computer Science or programming?
- Why do you want to participate in this workshop?
- How can you contribute to the success of this workshop?
- Please let us know if you have any other concerns:

Evaluation Plan: To evaluate the workshops, we will distribute an exit survey to participants during the closing session of the event to collect feedback on the workshop and collaboratively designed solutions. Potential exit survey questions are presented below:

Exit Survey

- On a scale of 1 (Very Dissatisfied) to 5 (Very Satisfied), please rate your overall satisfaction with the workshop.
- On a scale of 1 (Very Unlikely) to 5 (Very Likely), how likely are you to use the solutions co-designed in the workshop?
- Please explain:
- Please share other methods that can improve research software development tasks.
- How relevant were the discussion topics and co-designed solutions to your current role?
- How would you rate the collaboration and interaction among participants?
- Do you have any suggestions for how we can improve future iterations of this workshop?