Supporting Under-Resourced Software Engineering Job Seekers through Facilitating Online Collaboration in Technical Interview Preparation

Chris Brown (PI), Sang Won Lee (Co-PI)
dcbrown@vt.edu, sangwonlee@vt.edu
Virginia Tech

Abstract

To obtain a software engineering (SE) position, job-seeking programmers must complete a *technical interview* that involves solving a coding challenge while communicating their thought process to potential employers. While this process is widely used in industry, research shows tech interviews are stressful endeavors and incorporate bias, leading to interview underperformance and exclusion of underrepresented candidates. Thus, the proposed work aims to support under-resourced job seekers by facilitating *social interactivity* in technical interview preparation (FASIT). To evaluate this approach, we will explore current preparation methods and implement FASIT as a resource to enable online collaboration among job seekers for technical interview practice, aiming to mitigate bias and bring about more inclusive SE hiring processes.

1 Introduction

A *technical interview*, or tech interview, is a cognitively and socially demanding evaluation process for software engineers where candidates solve programming problems by writing code or pseudocode on a whiteboard or other non-programming environment (e.g., Google Docs [22] or Collabedit [17]) while communicating their problem-solving strategy by thinking aloud to potential employers. However, technical interviews often invoke stress in participants [7] and incorporate bias based on unfair criteria, such as candidates' ability to commit time and resources to preparing for assessments [4]. Ultimately, this leads to a "*leaky pipeline*" in software engineering (SE) hiring where qualified candidates, usually from underrepresented backgrounds, fail to attain positions in industry [32].

SE candidates rarely face authentic technical interview settings—*simultaneous coding while thinking aloud with an observer*—in Computer Science curricula, programming bootcamps, and industry. To that end, preparing for tech interviews can be time-consuming and stressful. Developers indicated that the time and effort needed to be competitive in SE hiring is frustrating [4]. For instance, the AlgoDaily blog recommends practicing 2–3 months to prepare for one interview, and longer for less experienced programmers [1]. Most existing interview preparation resources heavily focus on individual practice of typical programming challenges, providing limited feedback to users and ignoring think aloud skills–despite the fact that employers expect sound communication from candidates [21]. Otherwise, social features are often placed behind financial barriers.[1]

Hence, technical interviews are favorably inclined to hire candidates with more time and finances to prepare—a bias that can also lead to privileged candidates duping employers by gaming SE hiring processes [35]. To that end, we propose *FAcilitating Social Interactivity in Technical interview preparation (FASIT)* as a scalable approach for job seekers

[1]For example, coaching sessions for the technical interview prep resource Exponent (`https://www.tryexponent.com/coaching`) typically cost $1,000 for the recommended five sessions.

to exchange feedback on coding skills and think aloud practice in an ecologically valid setting. We will explore the approach of FASIT (pronounced "faucet") and investigate the effects of technical interview practice in an authentic rehearsal environment, where candidates receive feedback from a community of peers who will review interview practice recordings. To design and evaluate FASIT, we seek to understand current practices in tech interview preparation and implement a social computing system. This research aims to address the leaky pipeline by introducing a free and collaborative technical interview help system for under-resourced candidates to receive valuable feedback on authentic practice. Further, we will findings from our research to provide implications for job seekers and guidelines for employers to motivate more inclusive hiring practices.

2 Research Plan

RQ1: How do candidates prepare for technical interviews? To improve tech interview preparation, we first seek to understand current practice techniques adopted by SE job seekers. Our first task consists of a mixed methods study to explore how candidates prepare for technical interviews. To gain insight into technical interview preparation in practice, we will deploy an online survey and conduct semi-structured interviews, recruiting under-resourced populations of job seekers to participate, investigating prevalent practice approaches and challenges faced by candidates when preparing for tech interviews. The outcomes of RQ1 aim to provide insight into the resources and processes used by candidates to prepare for SE hiring assessments and to motivate the design of FASIT.

Expected Results: We anticipate the majority of SE candidates prepare for tech interviews in unauthentic environments. For instance, resources like LeetCode [30] and HackerRank [23] provide ample coding problems for users to try against solutions, but prioritize individual practice and lack support for combined coding and think aloud rehearsal.

RQ2: What design affordances are needed to make technical interview preparation authentic and collaborative? Our second task will implement FASIT for authentic and socially interactive interview practice. We will leverage our results from the first task to address gaps in candidate needs and existing tech interview prep resources. We will design FASIT as a multimodal system–providing mechanisms for users to solve realistic interview challenges with code and a means to communicate about them with other job seekers asynchronously. To that end, we will explore technological features necessary for authentic tech interview practice, such as live coding and audio/screen recording capabilities. Further, we will examine social features to foster collaboration for helping candidates improve via feedback on recordings and the effects of *spectatorship*, or watching the tech interview performance of other. To evaluate FASIT, we will conduct a controlled experiment to observe users completing real-world interview prep tasks, comparing coding and think aloud performance against commonly used resources derived from RQ1, and collect feedback from participants to generate implications for job seekers and employers.

Expected Results: We expect participants will have more developed communication skills while maintaining similar technical performance with FASIT compared to existing problem-focused resources. We also anticipate that social aspects of our system—feedback exchange and spectatorship—will provide more valuable feedback to users on their technical interview practice through self-reflection and collaboration with peers.

3 Broader Impacts and Research Merit

Technical interviews are biased towards candidates with the means to prepare, and aspiring software engineers rarely encounter authentic environments to practice for evaluations. Thus, the proposed work aims to address the leaky pipeline and support underresourced job seekers by contributing FASIT, a collaborative platform for technical interview preparation. Our research will involve job seekers with diverse backgrounds (i.e. race, gender, etc.) and experiences (i.e. students, seasoned candidates, and non-CS workers in career transition). This work seeks to broaden participation in SE by providing a novel resource for users to collaboratively rehearse skills necessary to succeed in interviews. We posit FASIT is a first step towards promoting inclusion in industry by enhancing interview outcomes through useful feedback on essential skills and mitigating issues such as stage fright. This approach also has the potential to benefit employers–reducing interview costs, approximately $22,750 per hire [20], generating a dataset of interview recordings to be further analyzed, and motivating inclusive hiring practices. The proposed work will advance knowledge in SE and HCI by understanding cognitively and socially complex processes through a technical intervention that facilitates collaboration and self-reflection, motivating the design of future interactive systems for programmers.

Data Policy. The output of this research will be submitted for publication to relevant venues. Data collected from participants will be stored securely and presented anonymously in aggregate. We will maintain a project website with updates on this work and the code for FASIT will be open source and publicly available in a GitHub repository.

4 Related Work

Prior work analyzed developer comments on Hacker News [4] and Glassdoor [6] to find concerns and frustrations with hiring processes. Ford et al. present mismatched expectations between candidates and interviewers [21]. Behroozi et al. used eye trackers [2, 3, 7] to explore the impact of privacy on interview performance, finding a significant increase of stress and decrease of performance in traditional interview settings—particularly for minorities, as no female-identifying participant passed the public interview but all succeeded privately without interviewer presence [7]. Moreover, studies show factors outside of performance, such as gender [16], race [32], and visible [31] and hidden [18] disabilities, impact hiring decisions. To that end, companies are lauded for "hiring without whiteboards" [34] and technical interview replacements have been proposed, including a speed-dating approach for underrepresented job seekers [19] and a former website offering candidates opportunities to skip interviews with paid sponsorships [24].

PIs' Prior Work. PI Brown investigated *asynchronous recorded interviews*, removing interviewer presence with screencasts of coding and think aloud, to find asynchronicity reduced stress and improved communication and technical performance [5]. Brown has expertise in conducting user studies [9, 11] and evaluating developed tools [10, 12, 33]. Co-PI Lee explored the impact of *spectatorship* and observing others in the context of creative writing, motivating the application of self-reflection and collaboration for technical interview rehearsal [13]. Further, Lee has implemented and evaluated various social computing systems in programming contexts [14, 15, 25, 26, 28] and beyond [8, 13, 27, 29].

References

[1] AlgoDaily. How to prepare for a technical interview. `https://algodaily.com/lessons/how-to-prepare-for-a-technical-interview`.

[2] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin. Dazed: Measuring the cognitive load of solving technical interview problems at the whiteboard. In *International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE NIER)*, pages 93–96, 2018.

[3] M. Behroozi and C. Parnin. Can we predict stressful technical interview settings through eye-tracking? In *Proceedings of the Workshop on Eye Movements in Programming*, pages 1–5, 2018.

[4] M. Behroozi, C. Parnin, and T. Barik. Hiring is broken: What do developers say about technical interviews? In *Visual Languages & Human-Centric Computing (VL/HCC)*, pages 1–9, 2019.

[5] M. Behroozi, C. Parnin, and C. 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)*. ACM, 2022.

[6] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin. Debugging hiring: What went right and what went wrong in the technical interview process. In *International Conference on Software Engineering: Software Engineering in Society (ICSE SEIS)*, 2020.

[7] M. Behroozi, S. Shirolkar, T. Barik, and C. Parnin. Does stress impact technical interview performance? In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 481–492, New York, NY, USA, 2020. Association for Computing Machinery.

[8] M. M. Bhuiyan, C. A. Bautista Isaza, T. Mitra, and S. W. Lee. Othertube: Facilitating content discovery and reflection by exchanging youtube recommendations with strangers. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Carrera and S. W. Lee. Watch me write: Exploring the effects of revealing creative writing process through writing replay. In *Creativity and Cognition*, Camp;C '22, page 146–160, New York, NY, USA, 2022. Association for Computing Machinery.

[14] Y. Chen, J. Herskovitz, G. Matute, A. Wang, S. W. Lee, W. S. Lasecki, and S. Oney. Edcode: Towards personalized support at scale for remote assistance in cs education. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–5. IEEE, 2020.

[15] Y. Chen, S. W. Lee, Y. Xie, Y. Yang, W. S. Lasecki, and S. Oney. Codeon: On-demand software development assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 6220–6231, New York, NY, USA, 2017. Association for Computing Machinery.

[16] D. Chinn and T. VanDeGrift. Gender and diversity in hiring software professionals: what do students say? In *Proceedings of the Fourth International Workshop on Computing Education Research*, pages 39–50, 2008.

[17] collabedit. `https://collabedit.com/`.

[18] A. Das. How tech interviews fail those with hidden disabilities. LinkedIn, 2020. `https://www.linkedin.com/pulse/how-tech-interviews-fail-those-hidden-disabilities-avik-das/`.

[19] T. R. Dillahunt, J. Lam, A. Lu, and E. Wheeler. Designing future employment applications for underserved job seekers: A speed dating study. In *Designing Interactive Systems*, DIS, pages 33–44, 2018.

[20] N. Doctor. The hidden cost of hiring software engineers — $22,750/hire. Qualified Blog, 2016. `https://www.qualified.io/blog/posts/the-hidden-cost-of-hiring-software-engineers`.

[21] D. Ford, T. Barik, L. Rand-Pickett, and C. Parnin. The tech-talk balance: What technical interviewers expect from technical candidates. In *International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE, pages 43–48, 2017.

[22] Google Docs. `https://www.google.com/docs/about/`.

[23] HackerRank. `https://www.hackerrank.com/`.

[24] L. Lawrence. Site allowing you to 'skip the interview' launches, then promptly shuts down. Protocol, 2021. `https://www.protocol.com/site-allowing-you-to-skip-the-interview-launches-promptly-shuts-down`.

[25] S. W. Lee and G. Essl. Live coding the mobile music instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 493–498, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.

[26] S. W. Lee and G. Essl. Communication, control, and state sharing in collaborative live coding. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 263–268, London, United Kingdom, June 2014. Goldsmiths, University of London.

[27] S. W. Lee, R. Krosnick, S. Y. Park, B. Keelean, S. Vaidya, S. D. O'Keefe, and W. S. Lasecki. Exploring real-time collaboration in crowd-powered systems through a UI design tool. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW):104:1–104:23, Nov. 2018.

[28] S. W. Lee, A. Willette, D. Koutra, and W. S. Lasecki. The effect of social interaction on facilitating audience participation in a live music performance. In *Proceedings of the 2019 on Creativity and Cognition*, C&C '19, page 108–120, New York, NY, USA, 2019. Association for Computing Machinery.

[29] S. W. Lee, Y. Zhang, I. Wong, Y. Y., S. O'Keefe, and W. Lasecki. SketchExpress: Remixing animations for more effective crowd-powered prototyping of interactive interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, UIST. ACM, 2017.

[30] LeetCode. `https://leetcode.com/`.

[31] Office of Disability Employment. Focus on ability: Interviewing applicants with disabilities. U.S. Department of Labor. `https://www.dol.gov/agencies/odep/publications/fact-sheets/focus-on-ability-interviewing-applicants-with-disabilities`.

[32] A. Scott, F. K. Klein, F. McAlear, A. Martin, and S. Koshy. The leaky tech pipeline: A comprehensive framework for understanding and addressing the lack of diversity across the tech ecosystem. *Kapor Center for Social Impact*. `https://leakytechpipeline.com/wp-content/themes/kapor/pdf/KC18001_report_v6.pdf`.

[33] 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.

[34] L. Tan. Hiring without whiteboards. `https://github.com/poteto/hiring-without-whiteboards`.

[35] J. Tsang. Why interviewing in tech is broken (and the direction it is going). Evermore, 2021. `https://jonathantsang.github.io/co-op3/`.