Panel: Synergistic Perspectives on Collaborative Learning with Version Control Tools

Mihaela Sabin
Applied Engineering and Sciences
University of New Hampshire
Manchester, New Hampshire, U.S.A.
mihaela.sabin@unh.edu

Kevin Buffardi
Computer Science
California State University, Chico
Chico, California, U.S.A.
kbuffardi@csuchico.edu

Simon Larsén
Electrical Engineering and Computer Science
KTH: Royal Institute of Technology
Stockholm, Sweden
slarse@kth.se

Bonnie MacKellar
Division of Computer Science and Math
St. John's University
New York City, New York, U.S.A.
mackellb@stjohns.edu

***Abstract*—Version control systems are widely used in industry and form a staple of modern software development, and this panel addresses the benefits and difficulties of introducing version control in education. Version control systems such as Git, and software development platforms such as GitHub, Bitbucket and GitLab are industry standards for which there is an increasing appeal within education. In general, students and teachers are motivated to use these tools in the classroom to increase graduates' competitiveness in the job market. Although highly desirable, the adoption of version control is hindered by a steep learning curve for both teachers and students, a lack of resources tailored to specific instructional and learning needs, as well as inadequate integration and alignment with the learning outcomes in the program curricula. The panel will analyze these challenges, solicit feedback from the audience's experience with version control tools, and facilitate discussions around promising strategies that could benefit student learning and graduates' professional competencies expected in the workplace.**

***Index Terms*—version control, collaborative learning, version control tools, version control platforms, Git**

I. Description and Rationale

Developing software and using version control are intrinsically connected in the software engineering industry today. That is why code repository hosting services, such as GitHub and Bitbucket, and version control tools, such as Git, are industry standards that have started to gain popularity among computing and engineering academic programs. A recent research study [1] found that using GitHub in the classroom predicts better student learning experiences. However, the study also pointed out that teachers did not perceive some use of GitHub as favorably as the students did.

Version control is of interest to computing and engineering curricula because it has professional relevance, helps students showcase their work on popular software development platforms, and facilitates effective feedback and collaboration with code review and issue tracking tools. Using version control in a variety of courses has the benefit of developing professional competencies for computing and engineering careers. In the absence of a curriculum that scaffolds learning of these professional tools, students end up learning version control on their own and having a superficial understanding on how it works.

In the education community, the adoption of version control creates opportunities for effective collection, assessment, and targeted feedback to student work. Code development and hosting services have the advantage of building rich data sets of student assignments and projects in multiple courses over many semesters. Such data sets represent an invaluable source for assessing students' individual contributions to a team project more efficiently and accurately. Investigating research questions about the effectiveness of version control tools and platforms also benefits from usage data that version control generates [2], [3]. Managing and mining these data sets remains a challenge. A common thread among the benefits of version control is the support for collaborative learning. Designing collaborative learning with version control, however, brings considerable challenges for both teachers and students.

II. Goal of the Panel Session

The overarching question that motivates this panel is what student learning supports and instructional resources for version control are needed to improve collaborative learning. To answer this question, the panel session will:

- Examine the motivation for adopting version control tools and platforms in the classroom
- Facilitate the sharing of various perspectives on the use of version control and its impact on instruction and student learning
- Coordinate productive exchanges on potential designs of effective collaborative learning with version control tools and platforms.

III. Topics to be Covered

To frame and enable informative and compelling exchanges and interactions towards achieving the panel's goal, we propose to organize the panel's topics into three themes, as described in the following subsections. Each theme outlines specific benefits, related challenges, or potential strategies for designing collaborative learning environments with version control.

A. Version Control Across Curricula

Version control can benefit a large array of courses: from introductory courses in computing and engineering to upper-division Software Engineering and DevOps courses, as well as non-major courses or courses in which student work involves non-computing artifacts. Equipped with a sound review of version control fundamentals, students can submit their work, share and showcase artifacts, and participate in collaborative projects using version control. In introductory programming courses, for example, teachers can give students access to starter code in a remote course repository on GitHub. Students fork the starter code to their GitHub repositories, clone to their machines, manage versions locally through commits, and then push their final version to their fork and submit the fork's URL to the course management system for grading. Alternatively, tools like GitHub Classroom or RepoBee [4], [5] enable instructors to automatically generate individual student repositories from a template without needing to fork. To gently scaffold their learning, students can also use novice-friendly, GUI-based Git clients, such as GitKraken, and coordinate collaborative activities through direct communication instead of version control.

In upper-level software development courses, version control can be expanded to encompass work on collaborative development, DevOps tools for static analysis and continuous integration, and unit testing. For team projects, the Git and GitHub toolkit includes issue tracking, project Kanban boards, and pull requests of branches linked to project issues and contributed by individual team members. These more advanced features provide teachers with rich opportunities to monitor and evaluate individual student performance [6]. In courses where student work consists of non-computing artifacts, students can showcase their portfolios with Git-based tools that conceal version control complexities of a command line Git client. The GitHub Pages tool, for example, enables students to conveniently publish and host a portfolio website by just editing HTML or Markdown files. With greater adoption of various GitHub features across curricula, students gain proficiency and comfort with version control as they build a valuable portfolio that showcases their technical skills.

B. Support for Student Learning

Some students tend to have positive attitudes towards version control because of its professional appeal in addition to their fluency with computing tools and particularly with navigating the version history of documents with tools such as Google Drive or Dropbox. Nevertheless, professional version control skills requires more advanced knowledge than simply navigating automated document version histories so care should be taken to distinguish the differences. Some students perceive version control as an add-on that takes an unnecessary toll on preparing and submitting assignments and projects. Often, the use of version control in a course is not designed with learning outcomes that are explicitly stated and coherently integrated with the learning outcomes of the course. Students make use of the version control commands to develop locally and share remotely in a mechanistic fashion. For example, when completing assignments individually, students' experience with Git version control is often limited to clone, add, commit, pull, and push commands, and lack understanding of the concepts behind these commands. Without such understanding, trying to troubleshoot Git errors generates a lot of student frustration.

To instill appreciation for the professional benefits of version control, teachers need to outline the value of remote repositories with which students can showcase their work to peers or external reviewers, including potential employers. However, appreciation will not suffice if student learning and practice with version control is not adequately supported. When used in a course, version control practice needs to be gradually scaffolded and realistically scoped throughout the semester and tied into the learning outcomes of the course.

Git is extremely beneficial for managing large scale, highly distributed projects. Effective use of Git in a semester-long, relatively small team project requires non-trivial project management workflows. If students are not familiar with these workflows or show uneven level of responsible participation, then instruction should leverage the efficiencies of version control tools to support and enhance, not hinder, effective collaborative practices.

C. Building and Sharing Adequate Instructional Resources

As a result of adopting version control for student work submissions and assessment, and for team projects and collaborative work, teachers face the intimidating challenge of managing a very large volume of repositories. This challenge translates into time-consuming and non-trivial tasks to organize repositories by course and semester over time; configure students' and teams' access to different types of repositories; and set up repositories for various purposes: starter code, student submissions, solution code, student team projects, instructional resources etc. The education community, GitHub, and other organizations and vendors have created open source and commercial tools to lower the entry barrier for teachers and help them manage repositories and version control-enabled collaborations. One such example is RepoBee [4], [5], which among other things helps teachers distribute and collect assignments, organise peer review, manage issue tracking as well as extend and customize the tool with plugins. The main challenge with additional plugins and tools is their accessibility to new users and expected extensibility to meet the needs of various courses. To further mitigate the risks posed by limited

TABLE I
PANEL SESSION STRUCTURE

	Description	Time (min)
1	Introduction of panelists and panel's format	5
2	Panelists' presentations and perspectives	20
3	Diverse groups' formation based on polling and experience levels	5
4	Theme-based group discussions and reporting out	35
5	Concluding discussions and further study directions	15

experience with the use of version control, communities of practice, such as the GitHub Education Community, facilitate spaces where teachers can share, collaborate, and support each other.

IV. ANTICIPATED AUDIENCE

We anticipate that computing and engineering educators who use software projects and coding assignments in their courses will be interested in introducing or improving on the use of version control. The Frontiers in Education community values project-based learning and team work, as evidenced by the abundance of conference sessions on this important topic. The community has also expressed interest in version control and its application to educational innovations in a variety of areas: agile software engineering [7], capstone programming projects [8], curricular content management [9], test-driven development [10], collaborative courseware development [11], and measuring project team members' contributions [12].

Our target audience are computing and engineering educators and researchers. Because of the professional relevance of version control tools and platforms, we also expect high interest from industry practitioners and students. To have a robust exchange of ideas and propose valuable and practical recommendations, we intend to invite and encourage participation of anyone with interest in version control.

V. PANEL SESSION STRUCTURE

Informed by the panelists' perspectives and experiences, the panel session will offer leading questions that will be posed to the audience to stimulate participants' engagement. In Table I we have outlined how the panel session will be structured. Ample time is provided for audience interactions.

The panel's primary focus is to generate productive discussions on how version control could become a common practice in project-based learning. The panel session will start with introducing the panelists and the format of the panel. It will continue with brief presentations by which panelists will share their perspectives and experiences with version control. A list of leading questions will conclude the panelists' presentations. Through polling, we will gauge participants' familiarity and level of experience with version control. Depending on the audience size, we will form groups of 3-5 or 5-8 members that have different levels of familiarity and experience with version control. These groups will engage in three discussion sessions prompted by the topical themes of version control described in Section III: 1) scope of use, 2) support for student learning, and 3) instructional resources. In the last segment of the panel, the panelists will facilitate concluding discussions around lessons learned, recommended resources, and promising directions for further study.

VI. PANELISTS

The panelists bring a variety of different backgrounds, experiences, and applications of version control in educational settings. The following biographical summaries illustrate their unique and synergistic perspectives.

A. Kevin Buffardi

Kevin Buffardi is an Associate Professor of Computer Science at California State University, Chico. His background in industry as a user experience (UX) specialist complements his current research in adaptive learning technology, software engineering education, and software testing. Buffardi teaches a variety of computer science and data science courses and is actively involved in expanding computer science education in California's primary and secondary schools.

B. Simon Larsén

Simon Larsén is a master's student at KTH Royal Institute of Technology, Stockholm. He has experience of version control in education both as a student being introduced to it, and as a teaching assistant tasked with managing and automating the educator's side of it. Simon's primary contribution to version control for education is the tool RepoBee [4], [5], which is actively maintained and expanded upon.

C. Bonnie MacKellar

Dr. Bonnie MacKellar is an associate professor in computer science at St John's University and is Program Director for the undergraduate Computer Science program. Before coming to St John's, she had extensive industry experience as a software engineer. She teaches upper level software design and software engineering courses and has research interests in software engineering education, in particular, the ways that students interact in group programming projects, as well as code quality.

D. Mihaela Sabin

Mihaela Sabin is a Professor of Computer Science at University of New Hampshire. She chaired the ACM/IEEE Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology Task Force and serves as co-chair of the ABET Computing Accreditation Commission IT subcommittee, charged with updating the accreditation criteria for IT programs by 2021. Sabin has over 20 years of experience in computing education with focus on increasing student engagement and broadening participation in computing education of students from underrepresented groups.

REFERENCES

[1] C. Hsing and V. Gennarelli, "Using github in the classroom predicts student learning outcomes and classroom experiences: Findings from a survey of students and teachers," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 672–678. [Online]. Available: https://doi.org/10.1145/3287324.3287460

[2] R. Glassey, "Developing feedback analytics: Discovering feedback patterns in an introductory course," in *Proceedings of the ACM Conference on Global Computing Education*, ser. CompEd '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 37–43. [Online]. Available: https://doi.org/10.1145/3300115.3309519

[3] K. Buffardi, "Assessing individual contributions to software engineering projects with git logs and user stories," in *In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. New York, NY, USA: Association for Computing Machinery, 2020, p. 650–656. [Online]. Available: https://doi.org/10.1145/3328778.3366948

[4] S. Larsén and R. Glassey, "Repobee: Developing tool support for courses using git/github," in *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 534–540. [Online]. Available: https://doi.org/10.1145/3304221.3319784

[5] R. Glassey and S. Larsén, "Towards flexible and extensible git-based course management with repobee," in *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 537–538. [Online]. Available: https://doi.org/10.1145/3341525.3393999

[6] kbuffardi/gitlog2csv: Python tool for creating a csv spreadsheet that summarizes git commits. [Online]. Available: https://github.com/kbuffardi/gitlog2csv

[7] X. Bai, D. Pei, M. Li, and S. Li, "The devops lab platform for managing diversified projects in educating agile software engineering," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5.

[8] M. C. Murray, "Work in progress — creating a professional software development environment to support capstone programming projects," in *2011 Frontiers in Education Conference (FIE)*, 2011, pp. S4F–1–S4F–3.

[9] S. Mandala and K. A. Gary, "Distributed version control for curricular content management," in *2013 IEEE Frontiers in Education Conference (FIE)*, 2013, pp. 802–804.

[10] C. Matthies, A. Treffer, and M. Uflacker, "Prof. CI: Employing continuous integration services and github workflows to teach test-driven development," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, pp. 1–8.

[11] J. Rodriguez, C. Apsey, S. Rees, T. Boudreau, and G. Raileanu, "Courseware as code setting a new bar for transparency and collaboration," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–4.

[12] R. M. Parizi, P. Spoletini, and A. Singh, "Measuring team members' contributions in software engineering projects using git-driven technology," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018, pp. 1–5.