Gaining Insight from Student's Use of Source Control

Simon Grey

University of Hull, Hull, UK,
S.Grey@hull.ac.uk

Abstract. For programmers experience of the use of software source control is a valuable professional skill and so it is vital that computer science students are exposed to source control systems before the graduate in order to enhance their employability. Additionally, there are additional benefits of students using source control for both students and teachers. For students source control provides a low risk environment and allows experimentation. For teachers source control systems can be used as a convenient medium to deliver scaffolding code, and allows the teacher access to student code in order to provide assistance to student's who need it. Additionally each student's use of source control provides valuable information about that student's behaviour and engagement with the work. This paper will begin by presenting a literature review of previous research use of source control systems by teachers and students. Following that some preliminary analysis of empirical data collected from student's interaction with subversion will be presented.

Keywords: source control, subversion, SVN, analytics, learning, teaching

1 Introduction

Experience of using source control and an appreciation of the benefits source control provides are valuable professional skills for computer science graduates. As well as its intended purpose of facilitating group work use of source control has significant potential benefits for individuals who are just learning to program. Anecdotal evidence shows that students who are learning basic programming concepts can shy away from experimentation because they are fearful that doing so may break their code. Source control can also provide students who are learning to program with a low risk environment in which to experiment freely - a vital part of the learning process - safe in the knowledge that their is a quick and easy way to undo any undesirable changes.

The remainder of this paper is split into three sections. Immediately following this one, in section 2 a literature review of publications concerning the use of teachers use of source control with their students. Following that, in section 3 a case study of use of source control on a second year graphics and simulation module is presented, together with some preliminary results. Finally, in section 4 conclusions and further work will be discussed.

2 Background

For computer science students an appreciation of the intricacies of source control is a valuable professional skill. Furthermore, in a learning environment source control can be used for more than just source code management.

This section provides an overview of published research concerning the use of source control management tools by students. In the next section 2.1 an overview over the various source control management tools will be presented.

2.1 Source Control Management Tools

The primary purpose of source control management tools is to enable multiple software developers to work on the same code base at the same time. As well as editing source code they also offer functionality to enable developers to get the latest changes, undo changes or merge changes to files that have been edited by two developers simultaneously. They serve as a record through time of the state of the source code.

Although Koc and Tansel identify four models for version control [11] two of those models are most prevalent. They are the *client/server* or *centralized* model and the *distributed* model. The centralized version control systems (CVCS) such as Subversion and CVS. Git and Mercurial are examples of distributed version control systems (DVCS).

The CVCS model uses the concept of a centralized, *golden copy* of the repository that exists on a central server. Individual developers *check out* a *working copy* of the golden copy. When changes are made they are integrated with the central golden copy via a *commit* command. Changes can be pulled from the golden copy down to the working copy by performing an *update*.

In the DVCS model each developer *branches* a local copy of the repository. Commits are made locally, and later the changes can be *pushed* back to the original branch. In recent years there has been a move away from the CVCS model and towards the DVCS model [1, 11] with Brindescu et al reporting that in a survey of 820 developers 65% use DVCS and 35% CVCS, and that the most popular source control solutions were Git (DVCS, 52%) and SVN(CVCS, 20%)[1].

2.2 Usage of Source Code Management Tools

Source control management tools have been used in a variety of ways in education, beyond the original purpose of facilitating group work. Of course, using source control to help facilitate student group work is still a valuable exercise. Beginning with group work this section presents a literature review of ways in which teachers have used source control in their teaching.

Group Work

Helping to facilitate group work is probably the most natural usage of source control. Using source control in this way is likely to give students a more authentic experience of source control that will arguably result in more employable

students. Much of the research concerning student's use of source control centres around group work [2, 3, 7, 12–14, 17].

2.3 Individual Work

Even though source control has been designed with groups of developers in mind, using source control also has benefits for student's working as individuals. The body of research into student's use of source control also includes a good amount of student's using their own individual repositories [4–6, 9, 16, 18, 19]. Storing work in a remote repository, rather than on the student's own hardware, or a thumbstick means that there will likely be a robust back up process in place. This protects students from loss of work through hardware failure. Additionally, and perhaps more importantly for students who are just learning how to write code use of source control allows freedom of experimentation. Anecdotal evidence tells stories of students who have broken working code, and have had to spend a lot of time trying to get back to a functional state. Not only do these students view this time as wasted, but they are also reluctant to make further changes or experiment with their code for fear of breaking it again.

2.4 Delivery of Material

Source control can also offer a more appropriate mechanism for delivery of course material to computer science students that is usually available in a VLE. Clifton et al give a comprehensive account of the benefits of using Subversion as part of course management[2]. When working with code based solutions spread across multiple files there seems to be pedagogic value in providing a good deal of scaffolding in order to remove barriers such as downloading, unzipping, linking and compiling a solution that separate a student from the practical application of learning outcomes they should be considering. This can be achieved by committing solutions into individual student repositories. It should be noted, however, that this treatment robs students of an opportunity to work with source control more deeply. Glassy required students to perform additional tasks such creating their repositories and importing code into them citing that students gained extra benefits because *"it forces students to become acquainted with all stages of version control usage."*[4].

2.5 Delivery of Assistance

Kertész describes, among others, the benefits of using source control to enable fast formative feedback from both student peers and instructors [9, **?**]. This is supported by anecdotal evidence suggests that there is an added benefit of source control for students seeking assistance from instructors. If a student experiences a programming problem they are able to get help either by email or in person. Students effective use of source control means that instructors also have access to that source code and are better able to determine exactly what problem the student is facing and provide guidance as is deemed appropriate.

2.6 Submission of Assessments

An extremely popular use of source control is to enable students to submit their assessed work for marking [2, 3, 5–8, 10, 12, 14, 15, 17–19]. In some cases subversion was also used to deliver formal feedback to students [2]. Additionally Gregorio and González-Barahona discus to potential for integrating plagiarism detection tools into student's software repositories.

2.7 Monitoring and Analytics

Often source control management systems include comprehensive logs of activity affecting the repositories of source code. This generates huge amounts of data for analysis. Monitoring students use of source control and analysing the data generated has been a focus of many studies [3, 4, 7, 9, 10, 13–16, 18]. Ganapathy et al [3] and Kim et al [10] present visualisation of student activity to instructors as a way of monitoring activity, however instructors commented that whilst this information was useful it was sometimes misleading - the number of commits was not a reliable proxy for student effort but sometimes a lack of commits was cause for concern. Jones[7] describes a simple process of performing a weekly manual review of student activity in groups using existing built in monitoring tools for Subversion. Novak et al[18] present a system for tracking student's interactions with Git based source control and visualize them using GitLab.

Mierle et al[16] mined student CVS repositories looking for performance indicators. They were unable to use metrics gathered from student source control to accurately predict performance, but did find a correlation between student performance and lines-of-code written. This result will be reaffirmed later in sections 3.2. Ljubovic and Nosovic[15] report a correlation between lines of code as reported by analysis software and time students reported they spent working on that code.

2.8 Summary

To summarize the key choices when choosing a source control solution have been outline in subsection 2.1 mostly centring around a decision between centralised and distributed source control. In subsection 2.2 a variety of ways to use source control are discussed with reference to literature. Aside from the main functions of managing source code and facilitating group work key usages include delivery of course materials and assistance, submission of assessment and monitoring student engagement. In the next section 3 a case study will be presented including preliminary analysis of data from three student cohorts' interaction with source control.

3 Case Study

This section will present data collected from a second year module in Simulation and 3D Graphics collected from three separate cohorts during 2014, 2015

and 2016. Subsection 3.1 includes background information concerning the setup of source control system and how it was used, as well as the tool chain used to extract data. Following that in subsection 3.2 some preliminary results are presented and analysed.

3.1 Setup

At the University of Hull students are provided with Subversion repositories despite it not being the most popular solution, or even version control model. Several factors were considered when choosing an introductory level VCS[1]. Polling student opinion gave no clear consensus. However as a centralized solution Subversion offers a clearer picture of student engagement through interaction with the repository. The DVCS model would allow students to gain all the advantages of using a VCS and making local commits without pushing changes to the server. This may also put the student's work at risk if they do not back up their local copy with the same regularity as the centralised repositories are backed up. Finally, it is perceived that Subversion is easier to understand than a DVCS, especially in the context of an individual developer rather than a group. Using a product called VisualSVN [2] allows adminstration and integration of student repositories using existing Microsoft based services including student authentication through Active Directory. Although we chose to host and administer our own source control solution it is worth acknowledging there are advantages to outsourcing this work. Lawrance et al [12] in particular describe the ease of using GitHub to provide repositories for education.

For each cohort a repository was created for the whole module. Within this shared repository students each have their own folder. Students have read and write permissions to their own folder, but cannot read or write to folders belonging to other students. Scaffolding code for laboratory assignments is committed to every students folder using a script. Laboratory assignments are guided and prompt students at to commit at appropriate times. Course work submission is also performed through the same repository. Additionally, within each student folder there was a read only folder that was used for delivery of feedback.

The laboratory work and assessment students complete is the same each year and so the years are directly comparable. As a measure against students copying each others work each student is given an individual data driven specification for their assessment created using their student ID as a random seed. This is commited to the read only folder in their repositories.

3.2 Results and Analysis

Data is extracted from the repositories using the StatSVN tool [3]. StatSVN is one of the tools suggested by Ljubovic et al [15]. StatSVN extracted data from

[1] Students are exposed to Team Foundation Server in later year as a project lifecycle management tool providing more than just version control.
[2] https://www.visualsvn.com/
[3] http://statsvn.org/

CSharp and shader code within the directory structure and ignored any output directories. Without explicitly ignoring these directories the results are skewed by students who committed intermediate files generated by the compilation process to the repository. Initially StatSVN was used to collect the Lines of Code changed by each student. It is worth noting that this metric is cumulative over all commits. That is, if a student were to change the *same line* ten times they would record 10 lines of code changed.

Students who failed to engage with assessment were removed as they have no mark. Students who interacted with source control outside of the normal semester, for example because they did a resit or a fresh attempt were also removed. A total of 185 students were remaining who between them made 11,605 commits changing 561,000 lines of code.

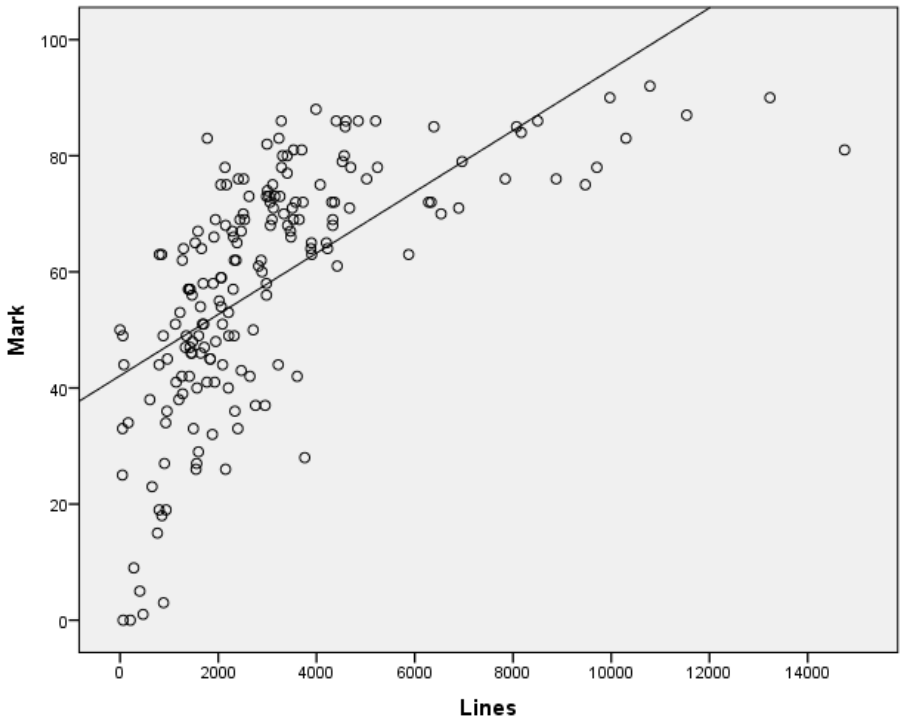

Fig. 1. Lines of Code Changed versus Mark Achieved

Figure 1 shows lines of code changed against mark achieved. A two tailed pearson correlation shows a strong positive correlation between lines of code and marks achieved of 0.64. The significance of the correlation was very significant at the 0.01 level. A linear line of best fit has been added to the graph. Although a cubic line provides a better fit, it is likely that this is due to the nature of the

course work, which is designed such that there is a greater weight towards easier functionality to ensure that all students who engage should be able to achieve something. Functionality that is more difficult to achieve carries fewer marks. This functionality is intended to challenge the more advanced students.

There appear to be a few outliers from students who wrote made far more changes that other students. Twenty students edited more than 6,000 lines of code over the course of the module. If these students are removed the pearson correlation increases to 0.70. The graph of this data is shown in figure 2.

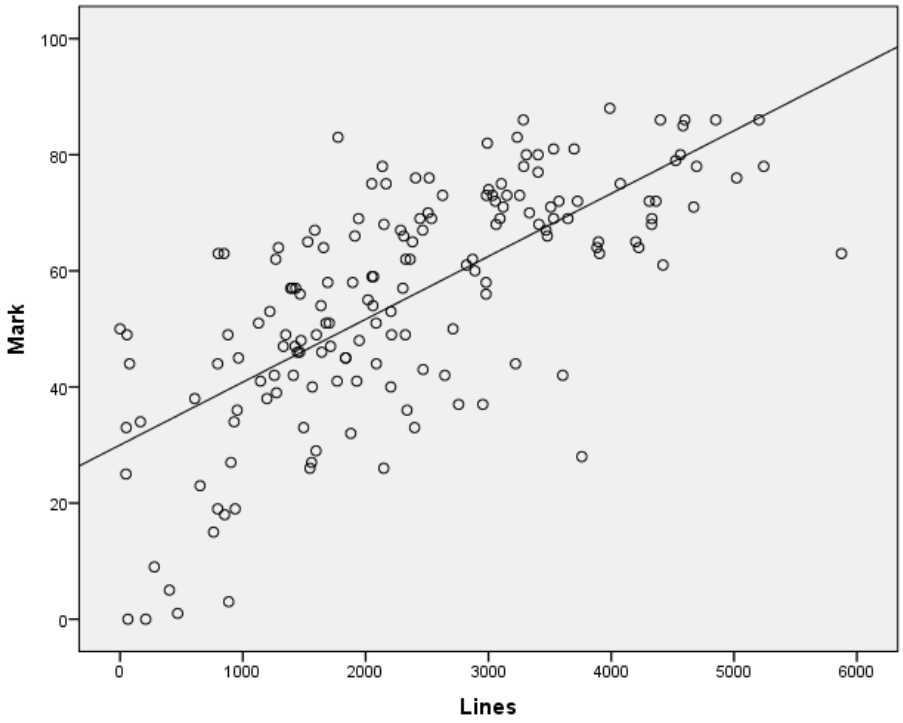

Fig. 2. Lines of Code Changed versus Mark Achieved Without Outliers

It is worth taking some time to consider why these outlier students made so many more changes than their peers. The source control logs revealed very well documented changes, some even using more advanced features of source control such as branching. These students typically worked steadily and consistently for the duration of the module. The code itself was often over-engineered for the scope of the module, indicating that these students were stretching themselves beyond the expectations of their tutors and creating their own challenges.

4 Conclusion and Further Work

Introducing source control to computer science students has clear benefits in terms of their employability as software developers. In education, however, it is proposed that there are additional benefits to using source control. Source control can provide students with a means to low risk experimentation on their code. When needed it can provide them with an easier and more productive route to assistance. It can provide a more efficient and appropriate link between instructors and students for the delivery of code based course material, a natural mechanism for submitting assessment and a means to provide feedback.

Student's use of source control also generates a great deal of data to be mined. The analysis of the data produced is an ongoing process. Future analysis will consider when students made changes both with respect to the working week, and the University trimester, and what changes students made. Although there does not yet seem to be any clear predictors for student performance there may be indicators of students who are failing to engage fully and are at risk of falling behind in their studies. With this goal, data should be extracted from source control for the first few weeks to see if there are any indicators of withdrawal early enough to make a timely intervention.

References

1. Brindescu, C., Codoban, M., Shmarkatiuk, D. D.: How Do Centralized Distributed Version Control Systems Impact Software Changes? ICSE 2014 (2014)
2. Clifton, C., Kaczmarczyk, L., Mrozek, M. Subverting the Fundamentals Sequence: Using Version Control to Enhance Course Management ACM SIGCSE, 39, 1, 86–90 (2007)
3. Assessing Collaborative Undergraduate Student Wikis and SVN with Technology-based Instrumentation: Relating Participation Patterns to Learning American Society of Engineering Education Conference, (2011)
4. Glassy, L.: Using Version Control to Observe Student Software Development Processes. Journal of Computing Sciences in Colleges 21, 3, 99–106 (2006)
5. Gregorio, R., Gonzalez-Barahona, J. Mining Student Repositories to Gain Learning Analytics. An Experience Report Global Engineering Education Conference (EDUCON) (2013)
6. Helmick, M. Integrating Online Courseware for Computer Science Courses ITiCSE'07, ACM Press, 39, 2, 146–150 (2007)
7. Jones, C. Using Subversion as an aid in evaluating individuals working on a group coding project Journal of Computer Sciences in Colleges, 25, 3, 18–23 (2010)
8. Kelleher, J. Employing Git in the Classroom WCCAIS'2014, (2014)
9. Kertész, C. Using GitHub in the Classroom - a Collaborative Learning Experience. SIITME 21, 381–386 (2015)
10. Kim, J., Shaw, E., Xu, H., Adarsh, G. V. Assisting Instructional Assessment of Undergraduate Collaborative Wiki and SVN Activities International Conference on Educational Data Mining (EDM), (2012)
11. Koc, A., Tansel, A.: A Survey of Version Control Systems. ICEME 2011 (2011)
12. Lawrance, J., Jung, S., Wiseman, C. Git on the Cloud in the Classroom SIGCSE'13, ACM Press, 639–644, (2013)

13. Liu, Y., Stroulia, E. Wong, K. Using CVS Historical Information to Understand How Students Develop Software MSR 1, 32–36 (2004)
14. Liu, S., Kim, J., Macskassy, S., Shaw, E. Predicting Group Programming Performance using SVN Activity Traces International Conference on Educations Data Mining, (2013)
15. Ljubovic, L., Nosovic, N. Repository Analysis Tools in Teaching Software Engineering IX International Symposium on Telecommunications (BIHTEL) (2012)
16. Mierle, K., Laven, K., Roweis, S., Wilson, G. Mining Student CVS Repositories for Performance Indicators ACM SIGSOFT Software Engineering Notes, 30, 4, 1–5 (2005)
17. Milentijevic, I., Vlanimir, C., Vojinovic, O. Version Control in Project Based Learning. Computers & Education 50, 1331–1338 (2008)
18. Novák, M. Biňas, M., Michalko, M., Jakab, F. Student's Progress Tracking on Programming Assignments Emerging eLearning Technologies & Applications (ICETA), 279-282 (2012)
19. Reid, K., Wilson, G. Learning by Doing: Introducing Version Control as a Way to Manage Student Assignments SIGCSE'05, ACM Press, 272-276, (2005)
20. Zagalsky, A., Feliciano, J., Storey, M., Zhao, Y., Wang, W. Emergence of GitHub as a collaborative platform for Education CSCW'15, ACM Press, 1906-1917, (2015)