Predicting Group Programming Project Performance using SVN Activity Traces

Sen Liu
USC Information Sciences Institute, 4676 Admiralty Way
Marina del Rey CA 90292
+1 213 880 8363
senliu@usc.edu

Jihie Kim
USC Information Sciences Institute, 4676 Admiralty Way
Marina del Rey CA 90292
+1 310 448 8769
jihie@isi.edu

Sofus A. Macskassy
USC Information Sciences Institute, 4676 Admiralty Way
Marina del Rey CA 90292
+1 310 448 8243
sofmac@isi.edu

Erin Shaw
USC Information Sciences Institute, 4676 Admiralty Way
Marina del Rey CA 90292
+1 310 448 9196
shaw@isi.edu

ABSTRACT

This paper presents a model for integrating student activity traces in a collaborative programming project using SVN, and relates different attributes of the SVN activities to student and team performance. We show how student participation patterns can be related to the grades of their group programming projects. Graph theory, entropy analysis and statistical techniques are applied to process and analyze data.

Keywords

Collaborative project, group project, SVN, data mining, entropy analysis, graph theory

1. INTRODUCTION

The goal of this case study is to make progress towards understanding the impact of collaboration on individual and group performance in programming courses that use a collaborative code management system, such as SVN (Subversion), which supports team-based programming projects by providing a complete history of individual programming activities. Past studies of group work have analyzed how the characteristics of team members affect group outcomes [1] and whether certain members or leaders influence performance [2]. Related work by the authors has shown that team pacing is highly correlated to group project performance [3]. Building on these results, we explore the following questions:

- Does individual coursework performance affect the group project performance?
- Does the most interactive (or influential) student affect the group project performance?
- Does even work pacing affect the group work performance?

Results indicate that when integrating components from different members, teamwork skills and usage of teamwork tools may improve the group performance; however, for implementing difficult programs, individual members' programming skills become more important. The performance of leaders or central students can affect the group performance greatly, and work pacing and management of the work throughout the project period can be an important fact for a successful team programming.

2. STUDY CONTEXT

To better prepare students for professional employment, two undergraduate computer science teachers at the University of Southern California combined a first and second year course so that students could work on an authentic project. This case study of that experiment spans a seven-week period of collaboration among students in the two classes.

2.1 Group Project Description

Students from the two courses formed 19 groups, each of which had 3 to 11 members from cs200 (freshmen) and 4 to 6 members from cs201 (sophomore). Each group designed and built a manufacturing assembly cell. The freshmen implemented the front-end and sophomores implemented the back end code. The students cooperated in designing the API between the two. The project had four subtasks: To design the project; to implement the components of the program (V0); to integrate the components (V1); and to implement non-normative case handling (V2). For teamwork planning and documenting, students made use of co-authoring tools. To manage code development, they used Apache Subversion (SVN). This work focuses on the SVN activities.

2.2 Data Description

SVN data from two semesters, 2011 spring and 2011 fall, was used for the analysis. Table 1 gives an overview of the data including students per team, number of files and number of file modifications made by each team.

Table 1. Summary of SVN data used for analysis.

	Group	N of Students	N of Files	N of File Mods
2011 FALL	M1	16 (5,11)	4007	5333
	M2	14 (5,9)	4007	6142
	T1	13 (4,9)	474	1433
	T2	13(4,9)	1740	3173
	W1	13 (5,8)	1603	2856
	W2	13 (5,8)	1412	3288
	W3	14 (6,8)	1994	3845
	W4	14 (6,8)	2082	3357
	W5	13 (5,8)	2156	3873
2011 SPRING	M1	11 (6,5)	2919	5885
	M2	10 (5,5)	3332	5276
	M3	11 (5,6)	1737	3243
	M4	9 (6,3)	2992	4279
	T1	10 (5,5)	1770	3370
	T2	12 (5,7)	1301	2871
	T3	10 (4,6)	1096	1842
	W1	12 (7,5)	5711	7287
	W2	9 (5,4)	1186	2184
	W3	11 (6,5)	2444	4137

Group project grades, other student performance (student exam and coursework grades), and SVN activity was used to analyze collaboration. The project grades of a group were computed by averaging the grades of its group members. CS200 had three assignments and two exams while CS201 had no assignments and two exams. SVN activity was measured by three variables that represent the degree of participation and collaboration in file co-

editing: 1) The number of files she/he modified or added each day, 2) The number of lines of code she/he modified or added each day, and 3) The number of interactions she/he made with each of the other group members in her/his group (if two students modify the same file, there is one interaction between them. If two students modify more than one file, then each "co-modified" file counts as an interaction.

3. DATA ANALYSIS AND RESULTS

This section uses graph theory and entropy analysis to explore participation patterns and their relation to project performance.

3.1 Relationship between influential student performance and group performance

Next we looked at the relationship between influential students and group performance to determine whether the performance of some members has an impact on group performance. Social Network Analysis (SNA) [4] has for decades been analyzing social networks to identify and categorize important people and has defined a variety of "centrality" metrics to identify different types of importance. We used three of these metrics for analysis: degree centrality, closeness centrality and betweenness centrality.

Degree centrality: The degree centrality metric counts the number of relations a person has (figure 1a). The more relations, the more important that person is because she/he talks to more people. A student who co-modifies code with many others might positively impact the project grade and is assigned a high degree.

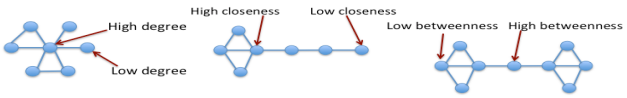

Figure 1. High/low (a) degree, (b) closeness & (c) betweenness

Closeness centrality: This metric identifies people based on how close they are, on average, to everybody else (figure 1b). To compute closeness, we define the *shortest distance, d(v,t)* to be the fewest number of edges needed to traverse from node *v* to node *t*. A student who is closest to everyone might have a lot of influence in the group.

Betweenness centrality: The last metric we consider in this study is the betweenness centrality (figure 1c). People with high betweenness are also known as "bridges" or "brokers" in that they sit between groups that otherwise do not have a lot of interaction. Such a student might be the one to help integrate the front- and back-ends and might positively impact the success of the project.

To compute this metric, we must first compute to what extent a students is a bridge. This is done by computing a shortest path for each pair of vertices. The *betweenness centrality* of node *v* is the number of shortest paths that go through a particular node. We generated one graph per group. Each vertex is a student and an edge indicates one or more interactions between them. Almost every pair of group members had a few interactions, so we only drew an edge between students with more than 10 interactions. After generating the graph, we computed the three metrics described above for each student. If stronger students are more central then we expect their projects to do better.

Two group graphs are shown in figure 2. The larger a vertex is, the larger the corresponding degree centrality. Nodes in red are cs200 students. Nodes in green (with an extra circle), yellow (square) and blue (diamond) are cs201 students. The yellow node (square) is the cs201 student with the best exam1 score. The blue node (diamond) is the cs201 student with the best exam2 score.

The left group was one of the better performers (project grades were 92.31 for cs200 and 94.75 for cs201). We have color-coded the nodes based on cs200/cs201 breakdown as well as the two best cs201 students (best exam scores). We find that cs201 students are more central than cs200 students and see closer interaction between the cs201 students. Finally, note that the two best cs201 performers are also the two most central nodes.

Figure 2. High (left) and low (right) performing groups.

In comparison, consider the graph on the right, which represents a group that performed less well (project scores were 88.63 for cs200 and 87.6 for cs201). We see a dramatic difference in the graph. First, although 3 of 4 cs201 students are quite central, the best cs201 student is not at all engaged (blue node to the far right). All cs200 students are engaged quite well. The blue node to the far right is the cs201 student with the best scores for both exams. These two group graphs seem to indicate that group structure, and in particular, the location of the better students, might significantly impact project grades.

3.2 Relationship between SVN activity and group performance

Finally, we look at the relationship between SVN coding activity and group project grades. We hypothesized that groups that work consistently will have a better grade than groups that do most of their work right before a deadline. To test this hypothesis, we turned to the information theoretic function of *entropy*. Entropy measures the amount of uncertainty in a system, or in our case, how much the activity of a group is spread throughout the project timeline Groups that are consistently active throughout their project will have high entropy, whereas groups that have a spike in activity towards the deadline will have low entropy. The entropy of a group's activity was based on the number of modifications. For each submission, the correlation and p-values between entropy and project grade (cs200+cs201) were computed. The correlations (p-values) between entropy and project grades were v0: 0.24 (0.33), v1: 0.59 (0.007), and v2: -0.37 (0.12). For v1, there is a significant positive correlation between entropy (working continuously) and group project grade; however, for v0 and v2, the p-values are large, which means that entropy and group performance are likely to be independent.

4. ACKNOWLEDGEMENT

The authors thank USC CS faculty Drs. David Wilczynski and Michael Crowley for their assistance. The research was supported by a grant from the National Science Foundation (#0941950).

5. REFERENCES

[1] Michaelsen, L.K., Sweet, M. (2008), The Essential Elements of Team-Based Learning, New Directions for Teaching and Learning, n116 p7-27 Win 2008.

[2] Strijbos, J. W. (2004). The effect of roles on computer supported collaborative learning, Open Universiteit Nederland, Heerlen, The Netherlands (Chapters 3 and 4).

[3] Ganapathy, C., Shaw, E. & Kim, J. (2011) Assessing Collaborative Undergraduate Student Wikis and SVN with Technology-based Instrumentation: Relating Participation Patterns to Learning, Proc. of the American Society of Engineering Education Conference, 2011.

[4] Wasserman, S. & Faust., K. (1994). Social Network Analysis. Cambridge: Cambridge University Press, 1994.