2006-317: ANALYZING COMMUNICATIONS ACTIVITIES IN STUDENT SOFTWARE PROJECTS

Frank Tsui, Southern Polytechnic State University

Frank Tsui received his PhD in computer and information science from Georgia Tech and has worked more than 30 years in the software industry. He is currently an associate professor of software engineering at Southern Polytechnic State University.

Orlando Karam, Southern Polytechnic State University

Orlando Karam received his PhD in computer science from Tulane University and is currently an assistant professor in computer science at Southern Polytechnic State University.

Analyzing Communications Activities in Student Software Projects

Introduction

It is well recognized that communications among the team members play an important part in the success of team projects[1]. This paper examines and characterizes the amount of communications that take place in the different activities and phases of software development projects. An important set of activities, project management, is also included in our study.

Previous research[2] has shown that team communications and team performance has a curvilinear relationship. Several studies[7,8] have shown that effective communication is related to success in information technology projects. Some preliminary data in Tsui's paper[3] has shown that the software project team with the most technical problems and least amount of leadership displayed the lowest amount of e-mail messages and volume of communications per team member. In Dutoit and Bruegge's paper[4] it is shown that communications artifacts generated by the software team can provide further insight into software development process and methodologies. The forms of communication have also been studied, and there is strong belief that the most effective form of communications for software development is face-to-face [5, 6].

We studied the communication activities among student team members who were given a software project to manage and complete. Nine student teams were studied. Each team was composed of four team members. These were relatively small teams where one might wonder if communications is an important factor as in large software project teams. While each team developed slightly different solutions, the project problem was the same. In other words, they were given the same requirements. The tools and process utilized by these teams were also very similar. Each team essentially performed requirements analysis, detailed design and code, and unit and functional testing. They all performed three major activities related to direct software development. In addition, each team prepared a project plan, presented a weekly status report, and a final project report. This set is considered indirect activities.

Three basic forms of communications were utilized by all the project teams.

- face-to-face meetings
- telephone
- e-mails

All communications are recorded in terms of person-minutes. Thus if three team members met for 20 minutes, the amount of communications is recorded as 60 person-minutes. Two people talking over the telephone for 5 minutes is recorded as 10 person-minutes. In the case of e-mails, only the construction and sending of the e-mail time by the author is recorded. For this study, no consideration was given to how many people were copied on the e-mail. The communications information were collected weekly and presented at the weekly project status presentations. All necessary clarifications or reclassifications of data were resolved during these weekly status presentations.

Each software project team was graded on the basis of the following criteria.

- meeting the functional requirements
- meeting the schedule (both intermediate and final)
- monitoring the project effectively

The teams may earn similar letter grade such as B, but they were also given numerical grades to retain a finer level of granularity. Project team success is defined in terms of the project team grade, and the numerical grade served as the measuring scale for success.

The following are the specific questions that we will address in this paper.

1. Does the amount of communications affect small software team project success?
2. What are the patterns of communication in software development?

Amount of Communications

Earlier research has shown relationship between the amount of communications and the success of projects. It was believed that frequent communications will increase team information exchange and thus increase team performance. In Patrashkova-Volzdoska and et al's paper[2] it is shown that over-communicating can actually be detrimental; so is under-communication. In our study, we tabulated the amount of communications among the team members for each of the nine teams. First, let's look at the spread of the teams by success and amount of communications in Figure 1.

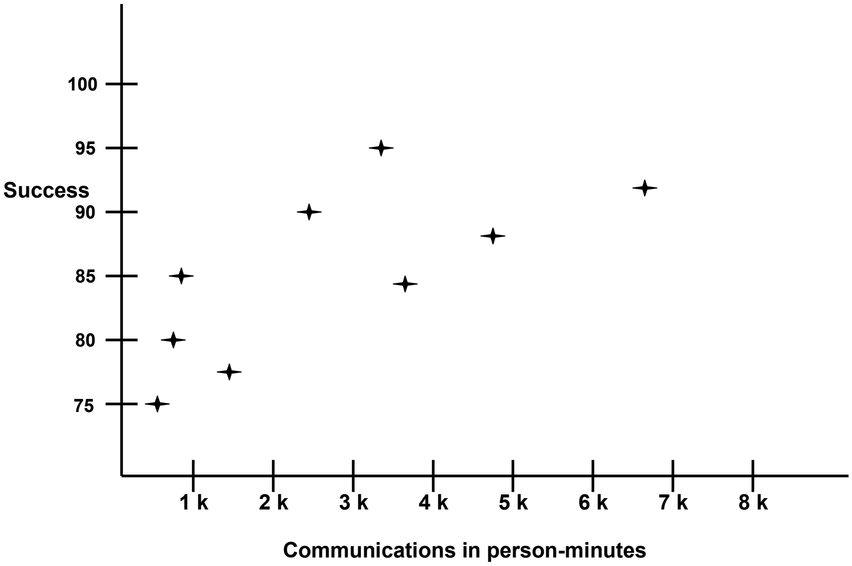

Figure 1 : Project Teams by Success and Amount of Communications

The spread of the data seems to indicate that the project teams with relatively higher amount of communications are more successful. It also shows that the one team with close to 7k person-minutes of communication may have the over-communications situation where it may not be "worth" spending all that communication time as the earlier work[2] has shown. However, the student software project teams normally do not spend much time communicating on topics outside of the immediate needs of the project. Thus the problem of over communicating and "wasting" time does not exist as much here as in other types of project environment. This point of differentiating IT and knowledge work, where communications are an important aspect of the work, from other types of project is explored in Sarker and et al's paper[9], and that may partially explain why our data does not show a curvilinear relationship. In addition, student projects are different from professional work-place projects in that student team members interact less on the non-project related items. They perform less social interactions with their team members unless the team members happen to also be his or her friends. The student teams are mostly virtual teams which are created just for the purpose of the class projects, and they are disbanded at the end of the class projects.

Table 1 shows the amount of communications listed in the order of project team's success, where success is defined as the team's project grade. In column three of the table, we show the ranking of the team by the total amount of team communications. Column 4 shows the difference in the two rankings.

Project Team Success Rank	Amount of Communications in person-minutes	Project Team Rank by Amount of Communications	Ranking Difference (diff)
1	3,202	4	3
2	6,750	1	1
3	2,529	5	2
4	4,895	2	2
5	770	7	2
6	3,785	3	3
7	736	8	1
8	1,570	6	2
9	675	9	0

Table 1: Team Success and Amount of Communications

By examining Table 1, one is inclined to observe that the successful teams tend to show more person-hours in communications. More formally, we tested our hypothesis that project team success and amount of team communications are independent. First, we use the Spearman Rank Order correlation coefficient as follows.

$$\begin{aligned} Rs &= 1 - \{ [6 * \ \Sigma(\text{diff}^2)] / [n(n^2-1)] \} \\ &= 1 - \{ (6 * 36)/(9*80) \} \\ &= + .70 \end{aligned}$$

The +.70 value indicates a relatively high positive correlation between team success and amount of team communications. Then the Z value is computed as follows.

$$Z = Rs * \sqrt{n - 1} \ = \ 1.97$$

Using the Z–table, we can reject, with approximately 95% confidence, that project team success and amount of team communications are independent. Thus our visual observation of the table that more successful software project teams tend to be the ones that communicate more seem to be correct.

We further subdivided the nine project teams in Table 1 into three subgroups of top three most successful teams, middle three teams and the bottom three project teams. Table 2 shows the mean amount of communications for each of these groups.

Project Groups	Mean Amount of communications in person-minutes
A: Top Three	4,160
B: Middle Three	3,150
C: Bottom Three	994

Table 2: Each Group's Mean Amount of Communications

We applied the student t-tests to {3! / [2!* (3-2)!] } pairs or three pairs of the groups: (A,B), (A,C), and (B,C). The pair (A,C) shows the biggest difference and has a 2.36 t-value. That indicates a high probability that the top performing group is communicating a lot more than the lowest performing group. The pair (A,B) has the smallest t-value of .56 and is mostly likely not that different in the amount of communications. The pair (B,C) has a t-value of 1.71. We believe that this indicates that while there is a difference in the amount of communications, the difference is not that large. Using student t-tests to compare pairs of the three groups also suggest that there is a positive relationship among the groups' successful performance and the amount of communications.

Patterns of Communications

The software development activities are divided, at the macro level, into four categories as follows.

- requirements analysis and specifications
- design and coding
- functional testing
- project control and management

It is important to point out two areas that the information gathered in the weekly status reports for this research may differ from a more traditional, large commercial software project. The first is that the requirements solicitation and gathering activities were relatively simple. Since these were class projects, there were limited problems in resolving conflicting requirements or difficulty in scheduling meetings with users and clients. The other is that the nature of the project did not require a large number of people and did not require extensive amount of testing beyond functional testing. There was no need for multiple platform systems test or regression tests. The pattern of communications described here is for small teams of four people.

Dutoit and Bruegge[4] found that the amount of communications during the different major phases of software development show a double camel hump with the first during requirements phase and the second during integration testing phase. Another study of commercial software development organizations by Kaushik [5] found that the amount of communications showed a steady increase from requirements activities through design and code to testing activities. In both of these cases, an important set of activities, project management, was not included. Our study showed that, among the four major software development activities, the highest percentage of the communications is spent in project management activities.

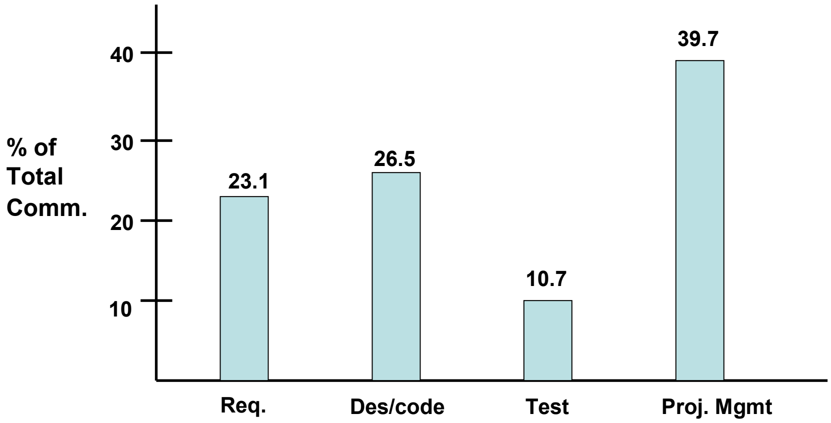

Figure 2: Distribution of Communications

Figure 2 shows that the activities related to project management accounted for 39.7 % of all the communications. This is not a surprise in that project management is a heavy information exchange activity and a heavy people interaction activity. This information does serve as a clue to the relative amount of communications needed to perform project management. What is a surprise is the low percentage of communications, only 10.7%, during the testing activity. It is possible that the nature of the problems found during testing of small team projects is relatively simple that not much discussion is required among the team members. However, during the final presentations of these student projects, it became clear that the amount of testing and the emphasis on testing was low. The students all appreciated the importance of quality product, but most placed schedule ahead of everything. The percentage of total communications during design and coding is just slightly higher than that of requirement activity, as stated earlier, due to the nature of the project where requirements gathering in these students' projects did not face some of the more difficult problems resulting from users and clients. Some of the teams did encounter the traditional requirements closure problem, though.

The previously partitioned three groups of A(top three), B(middle three), and C(bottom three) teams show an interesting difference in the distribution pattern.

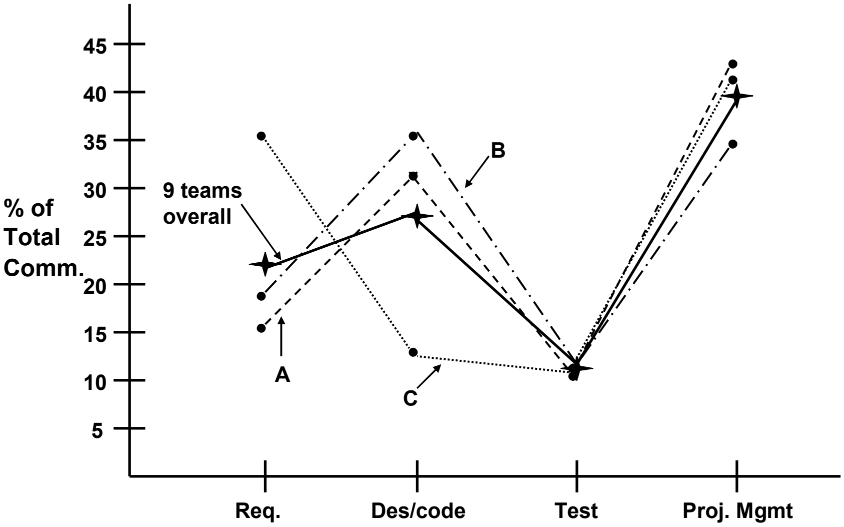

Figure 3: Distribution of Communications by Groups

Note that the nine teams overall pattern is shown with solid line in Figure 3. Both the most successful group, A, and the middle group, B, of project teams show a similar pattern as the overall pattern. All three groups had low percentage of communications in testing activities and high percentage of communications in project management activities. The least successful group, C, had a high percentage of communications during requirements activities and a very low percentage of communications during the design and coding phase of activities. This is interesting in that the pattern for these two phases of activities actually matches the earlier mentioned reference[4]. However, in this case, there is another explanation for group C's communication pattern. These teams in group C did not perform any more requirements gathering or solicitation than the teams in group A or group B. Even though the relative percentage of time spent on requirements for Group C was higher, the actual amount of time spent is estimated at 35% (Figure 3) of

994 person-minutes (Table 2) or a total of approximately 348 person-minutes. The actual amount of time spent on requirements by Group A is estimated at 15% of 4160 or approximately 624 person-minutes, and for Group B the estimate is 17% of 3150 or approximately 535 person-minutes. In addition, we observed from the weekly status presentations that the least successful group, C, of teams had more problems with requirements analysis and coming to consensus among themselves. To overcome the team consensus problem, the teams in Group C had to communicate, relative to their other phases of software development, much more during the requirements phase.

Concluding Remarks

In this paper we have studied nine student software project teams performing a similar project within a same duration period. The teams' communications volume and communications patterns were studied within the context of software development projects. Several important observations are made for these student projects.

- The volume or amount of communications and the success of software project team performance are positively related. The relationship was not found to be curvilinear as in some earlier study [2].
- The pattern of communications in software development projects indicate that the activity which had most of communications efforts was project management related activities.
- The amount of communications for testing related activities was the least.
- The least successful group spent more percentage of time communicating on requirements than the other two groups. However, the actual time spent on requirements was still far less than the more successful groups.

Two observations stand out from our work. The first is that the relatively high amount of communications expended in the project management activities. It was a surprise to most of the students. This is important information for future software project planning purpose. The second is that spending a high percentage of time communicating during requirements phase, compared to other phases of development, may indicate that there is a problem with requirements rather than the traditional belief that more time spent in the front-end would result in a better project. If the high percentage of communications effort spent represents instability of requirements and difficulty in reaching agreement, then the team should be alerted for potential downstream problem in software development where these open requirements still need to be resolved during design, coding or even testing phases. The point on coming to consensus on requirements is a very real problem even in the industry. Students usually do not experience this problem because of the artificial environment where the requirements are often handed to them by the instructor.

Our findings are limited to relatively small, one-semester-long, student software development projects. In the future, we intend to broaden the base and the complexity of projects to require more extensive requirements activities and more systems builds with regression tests. Secondly, we intend to investigate the relationship between the patterns of communications and the patterns of actual performance efforts.

References

1. Smith, K.A. (2000) Project Management and Teamwork, McGraw-Hill, 2000.
2. Patrashkova-Volzdoska, R.R., McComb, S.A., Green, S.G. (2003) "Examining a Curvilinear Relationship Between Communication Frequency and Team Performance in Cross-Functional Project Teams," IEEE Transactions on Engineering Management, vol. 50, No 3, august, 2003, pp 262-269.
3. Tsui, F. (2004) "Information Communications in Software Projects," Proceedings of the 2004 ASEE SE Annual Conference, Auburn University, Alabama, 2004.
4. Dutoit, A.H. and Bruegge, B. (1998) "Communication Metrics for Software Development," IEEE Transactions on Software Engineering, vol. 24 no. 8, August 1998, pp 615-628.
5. Kaushik, S. (2001) "A Study of Attributes of Communications as They Relate to Software Development," Master's Research Project, Southern Polytechnic State University, 2001.
6. Cockburn, A. (2000) "Selecting A Project's Methodology," IEEE Software, July/August 2000, pp 64-71.
7. Roberts, T.L., Cheney, P.H., Sweeney, P.D. (2002), "Project Characteristics and Group Communication: An Investigation," IEEE Transactions on Professional Communication, vol.45, no.2, June, 2002, pp 84 – 96.
8. Hirschheim, R. and Newman, M. (1991) "Symbolism and Information Systems Development: Myth, Metaphor, and Magic," Information Systems Research vol.2, no.1, 1991, pp29-62.
9. Sarker, S., et al, (2005) "Knowledge Transfer in Virtual Systems Development Teams: An Exploratory Study of Four Key Enablers," IEEE Transactions on Professional Communication, Vol. 48, No.2 June 2005, pp 201-218.