ELSEVIER

The Journal of Systems and Software 53 (2000) 219–224

The Journal of Systems and Software

www.elsevier.com/locate/jss

Types of collaborative work in software engineering

Pierre N. Robillard [a,*], Martin P. Robillard [b]

[a] École Polytechnique de Montréal, Department of Electrical and Computer Engineering, Laboratoire de Recherche en génie logiciel, C.P. 6079, Suc. Centre-ville, Montréal, Que., Canada H3C 3A7

[b] The Department of Computer Science, University of British Columbia, 201-2366 Main Mall, Vancouver, BC, Canada V6T 1Z4

Received 3 September 1999; received in revised form 8 September 1999; accepted 22 October 1999

Abstract

This paper is an account on work distribution analyzed from the collaboration point of view. It presents a new classification of the collaborative work in software engineering project. Four types of collaborative work are defined derived from empirical measurements of activities. Mandatory collaborative works are formal scheduled meetings. Called collaborative work is defined when team members call a meeting to solve a problem, which is most often technical. Ad hoc collaborative work is defined when team members work on the same task at the same time and individual work occurs when a team member works on its own on a task related to the project. Data are extracted from the logbook filled out by four team members working on an industrial project that lasts 19 weeks. The characteristics of each type of collaborative activity are described and a quantitative breakdown of how people spend their time in collaboration within a single project is presented. © 2000 Elsevier Science Inc. All rights reserved.

Keywords: Software project; Collaborative work; Team work; Measurements and software engineering

1. Introduction

An understanding of the work patterns is needed for eventually supporting that work through computers, internet and collaborative systems. Software engineering is knowledge intensive (Robillard, 1999) and a large project is mostly done by a team of people. Collaborative work is often a key issue in software engineering. Various aspects of teamwork that have been studied by the social science community among many subjects are the group stress created by members working on multiple projects (Ancona and Caldwell, 1990), the problem of measuring and evaluating teamwork (Christ and Muckler, 1993), and the aspect of technology and time in group communication (McGrath, 1990).

Collaborative work and team structure are also studied in software engineering from various viewpoints. Researchers in software engineering, showed interest in team internal relationships and for the interactions between teams and their environment. For example, (Bendifallah and Scacchi, 1989) demonstrated that work structures map to task and conclude that in software activities, the most critical variable for team structuring concerns the underlying work structure. (Burke et al., 1995) investigated the development of various group characteristics in both face-to-face meetings and in distributed meetings over time. Other researchers focused their attention on formal review meetings. For example, (Johnson and Tjahjono, 1997; Adam and Johnson,1997) investigated the cost/benefit ratios of meeting-based reviews versus non-meeting-based ones. (Porter et al., 1997) studied the effects of developer activities on Inspection intervals, and (Seaman and Basili, 1997, 1998) studied communication in code inspections. Although the previous studies in empirical software engineering were able to provide insight into the group dynamics involved in the development of software, most of them only focused on a particular activity, namely, software reviews. The first contribution of this paper is thus to present and interpret data concerning all the team activities involved in a software development process and to contrast them with the more individual activities.

This paper presents data based on a defined software process. Few case studies in empirical software engineering also provide information about software processes. (Bradac et al., 1994; Perry et al., 1994) presented the relative time spent by software developers in the various tasks they execute. However, these studies

* Corresponding author. Tel.: +1-514-340-4238; fax: +1-514-340-3240.
E-mail address: pierre-n.robillard@polymtl.ca (P.N. Robillard).

illustrate the proportion of time the developers invest in activities such as writing specifications, coding, and the like, without concern about what time is spent working individually versus what time is spent doing teamwork. Teamwork activities in software engineering course environment have been studied under the aspect of process, activities and relationship to professional environment (Robillard, 1995, 1996; Robillard and Robillard, 1998). A second contribution of this paper is to present data illustrating what tasks in software development involve more teamwork as compared to tasks where more individual work is needed.

This paper reports on a project that has the following five characteristics:
1. It is a case study rather than a controlled experiment.
2. The data are extracted from the logbook of the participants. The logbook has been required in this organization for many years. The participants are not filling out the logbook for the purpose of this study and the data from the logbook have been validated.
3. All participants are professional software engineers with some years of experience, which is in contrast to the studies done with students or juniors.
4. Participants are not aware that this kind of study is performed so there are no artificial behaviors.
5. Many kinds of teamworks are studied. This is different from studies that are focused on a specific type of meeting, for example, inspection.

Section 2 describes the major characteristics of the project. Then, the processes used by the team are described; the measurement tools are presented and data are presented. We are aware that the data presented in this paper are based on specific projects and that a word of caution is required before generalizing the conclusions. However, we believe that the new classification scheme derived from the data analysis is of general interest and is not specific to this project.

2. Project description

The goal of the project was to design and implement a simulator of Petri Net (Marsan, 1990). This simulator is integrated into a modeler, which is distributed worldwide. Requirements are well defined and the task is to design and implement the simulator. The project was developed in a Windows environment in C++. An object-oriented methodology was used. The project was successful, all requirements were implemented and tested, the documentation was appropriate and the schedule and the budget were respected.

The project has well-defined requirements for the design and implementation of computational algorithms and there is no input or output interface. All data are transmitted to and from files. The software engineering process is defined. All team members are required to fill out a daily time log. They have been informed that the daily time log is for research purpose only. The data from the logbook have been validated. The data will not be available to the human resources or the management people to evaluate performance or productivity. However, they do not know the subjects of the research conducted. Some meetings have been videotaped and are the subject of research on specific collaborative activities (Robillard et al., 1998; d'Astous et al., 1998).

Table 1
List of standards used to define the process

IEEE std 829	Software Test Documentation
IEEE std 830	Software Requirement Specifications
IEEE std 1008	Standards for Software Unit Testing
IEEE std 1016	Recommended Practice for Software Design Description
IEEE std 1058	Software Project Management Plan
IEEE std 1074	Developing Software Life cycle Processes

The teams used a defined software development approach. The main characteristics of software development processes are:
- A systematic approach to software development.
- The use of appropriate software engineering standards and CASE tools.
- The use of reviews for each step of the software development process (Freedman and Weinberg, 1990).
- Preparation of the appropriate documentation for software development.

The software process used is inspired from the Capability Maturity Model (CMM) key practices and is based on the (IEEE Standards Collection, 1994) list in Table 1.

The first phase of the industrial project, which corresponds to the first release, lasted 19 weeks and required four full-time software engineers. Two software engineers had a degree in computer engineering and one a degree in computer science. The coordinator had a Master's degree in software engineering. Their professional experience ranged from 2 to 7 years. They used a democratic team approach, with one of the team members acting as the coordinator.

The team members share a large room subdivided into cubicles. A nearby small meeting room was available on request. All team members were dedicated to the project on full time basis. Except for a 1 h mandatory meeting once a week the team members were free to schedule their own meeting activities. The project was typical of the projects conducted in this organization for many years with the same high level management.

3. Definition of the measures

This section describes the approach used to gather information on the team activities. The time spent by an individual on each activity is captured by filling out, on

Table 2
Definition of the field of the records for the logbook

TIME	Clock time. It is captured automatically
ID	Identification of the team member
DURATION	Time spent on the given activity (multiples of 15 min)
NB	Number of team members involved in the given activity
PHASE	Name of the phase of the process
ACTIVITY	Name of the activity being measured
TASK	Name of the task as defined in the planning of the project
COMMENTS	Any comments that are relevant to the activity record

a daily basis, an activity record once the activity has ended. All the records are stored in the same database (Access). An activity record is made up of eight fields, which are defined in Table 2.

Each individual was responsible for filling out his own logbook. One team member was responsible for collecting the completed logbooks of teammates every week and integrating them into the team database. All the data presented in this paper were obtained through analysis of the logbooks.

This measuring process seems straightforward and easy. However, it took us many projects to develop reliable measuring mechanisms. The major difficulty is in defining unambiguously the phases, activities and tasks. To do that requires a definite process and good project planning.

4. Four types of collaborative activities

Teamwork is made of a variety of activities where the collaborative nature of the work may vary from one activity to the other. We found that the team activities could be divided into four types of collaborative activities. Mandatory, Called, Ad hoc, and Individual:

- *Mandatory collaborative activities* are formal meetings scheduled on a regular basis. They are usually planned long in advance and participants are required to attend.
- *Called collaborative activities* occur when two or more team members decide to get together to do some technical work. Usually these meetings are not scheduled long in advance and only participating members will attend.
- *Ad hoc collaborative activities* occur when two or more team members work on the same subject at the same time and share on ad hoc basis comments, or information on what they are doing.
- *Individual activities* occur when a team member is working on a task that is not shared at the same time by other team members and is then unlikely to interact on this subject with other team members.

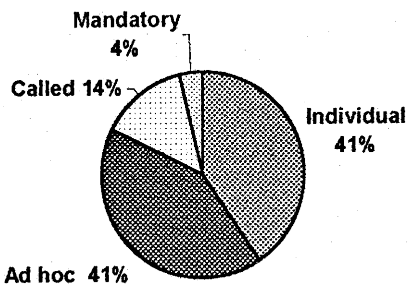

Fig. 1. Relative time spent in the collaborative activities.

Fig. 1 shows the relative time spent in each type of collaborative activity for the duration of the whole project. The mandatory activities are composed of the short meetings required by the project leader to synchronize the various tasks. It is interesting to note that even if it is named a team work almost 41% of the time is spent in Individual activities. The same amount of time (41%) is spent on Ad hoc collaborative activities. Finally only 14% of the total time is spent on formal Called technical meetings.

Fig. 2 shows the distribution of the collaborative activities for each week of the 19 weeks of the project duration. It is observed that the Individual activities form cycles, which are preceded by Ad hoc collaborative activities. It seems that the team members are working together to define various tasks and then they work on an individual basis to complete the task. There is a straight decrease in Mandatory activities as the project progresses. Called collaborative activities are more intense in the middle of the project.

Fig. 3 shows the distribution of duration for each collaborative activity. Most of the Mandatory activities (80%) last less than 1 h. The Called collaborative activities are mostly (70%) in the 1–2 h range. Ad hoc collaborative activities are clearly preferred for tasks of long duration (3–5 h) while Individual activities are kept for smaller tasks. More studies are required to confirm the individual behavior. Intuitively, we had expected longer individual tasks. This could be a side effect of the team cohesion or it could be related to the small size of the team. However, we suspect that such behavior is likely to improve the quality of the product.

Fig. 4 shows the number of participants for each type of collaborative activity. The Called and Ad hoc collaborative activities are often assumed (60% of the time) by two of the teammates. The Mandatory collaborative activities are usually conducted with the full team as expected. Since there are few Mandatory activities, see Fig. 1, the 3–4 meetings that were not fully attempted account on a relative basis for 40% of the meetings.

Fig. 5 shows that all the participants (P1–P4) have almost an equivalent involvement in each of the collaborative activities. In that sense this project was truly a

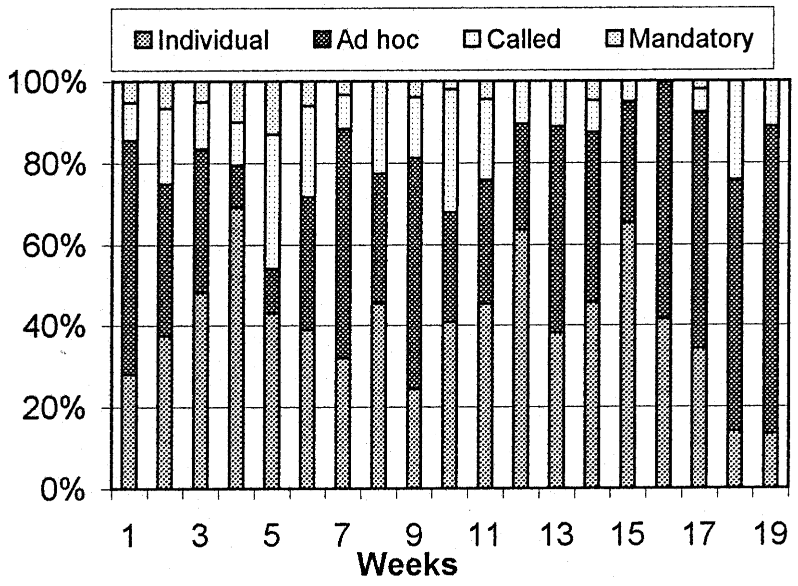

Fig. 2. Weekly distribution of the collaborative activities.

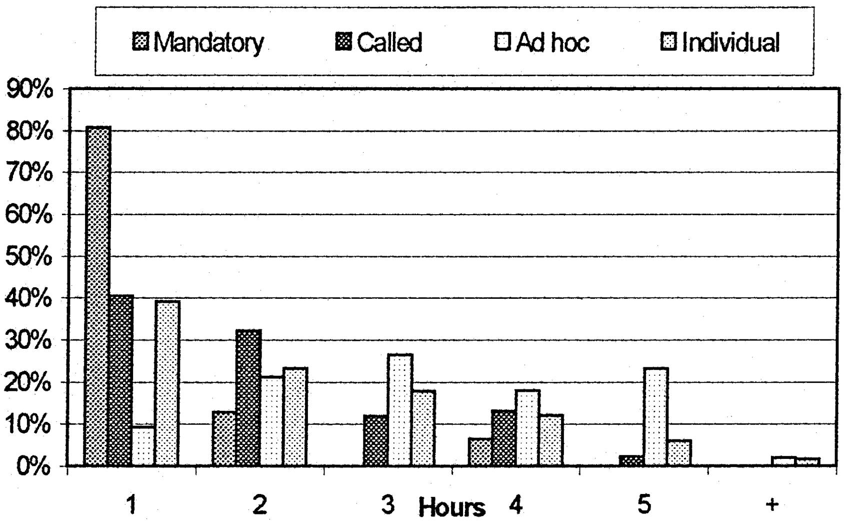

Fig. 3. Collaborative activities duration.

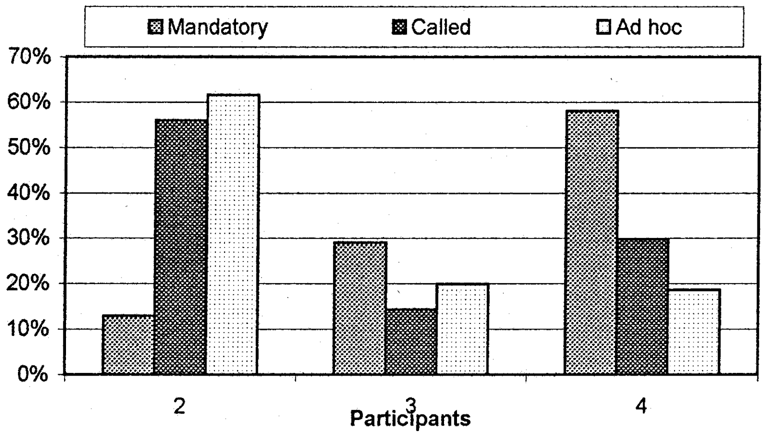

Fig. 4. Number of participants for each type of collaborative activity.

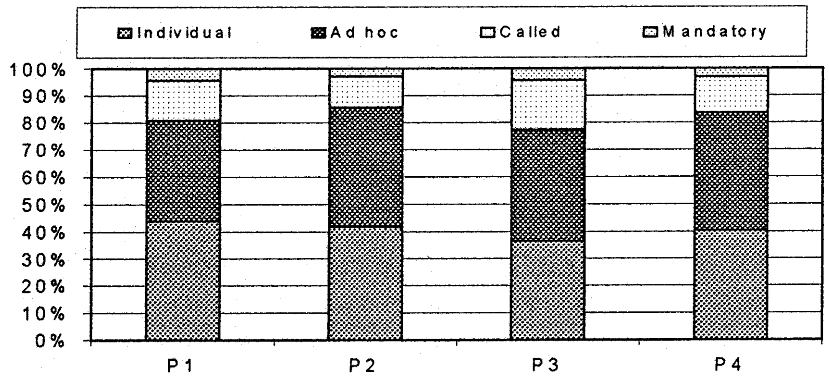

Fig. 5. Involvement of each participant in each type of collaborative activity.

team project even if 80% of the activities were not what is usually identified as team activities since Individual and Ad hoc collaborative activities are usually not considered as team activities. We do not observe any significant different involvement pattern for the team coordinator (P3). This may be due to the small size of the team and the democratic team structure.

5. Conclusion

This analysis of team members' logbooks showed that the team project could be seen as made of four types of collaborative activities. The Mandatory and Called collaborative activities are easy to identify and their meaning is straightforward. It is interesting to observe that these formal meetings account together for less than 20% of the time spent by the team members. More studies, which could be focused on the content of the meeting, are required to determine if these face-to-face meetings can be held through electronic communications. The mandatory meetings can be seen as a mechanism to synchronize the team members on the project. It is observed that when synchronization is no more required in the last weeks of the project, for example, mandatory meetings are reduced to the minimum. This study suggests that the frequency of mandatory meetings should be adjusted to the needs for team synchronization.

Called collaborative activities meetings are in fact working meetings. In these meetings team members discuss analysis, validate design or do problem-solving activities. These meetings are usually scheduled but with a very short notice (few hours to one day or two).

On-going researches, based on protocol analysis and methods used in cognitive psychology, are trying to understand the detail activities of these meetings (d'Astous et al., 1998). Preliminary data indicated that there are five types of exchanges: cognitive synchronization, which occurs when participants make sure they share a common representation of a given subject, evaluation, which occurs when participants judge the value or give their opinion on a particular subject and their acceptance, elaboration of alternative solutions, which occurs when a participant proposes a new solution to the subject under study, conflict resolution, which occurs when participants have an argument about a given subject, and management, which occurs when participants are planning the ongoing working session or the forthcoming meetings.

We found these meetings important for the quality of the product based on the topics that have been discussed. Most of these meetings were technical review meetings where team members validate a design component or a piece of code. Some of the meeting works could probably be done from remote site and electronic communications. However, preliminary studies of the content of these meetings indicate that the contributions of face-to-face activities are significant.

Ad hoc collaborative work is a new type of activity that has been created to describe the collaborative activities observed when people are working at the same time, in the same room and on related tasks. The exchange is truly informal and sporadic. Most of the time a teammate is interrupting his work to answer a question from a fellow teammate. In that sense, the Ad hoc collaborative activity is different from the Called collaborative activity where the teammates involved agree in advance to a planned meeting. Ad hoc collaborative activities are significant but very difficult to measure. We observe that some people will use electronic mail to ask questions and provide answers even if they are sitting at the next desk. We do not know yet the importance of face-to-face meeting in the Ad hoc collaborative activities. We have little information on the content and the usefulness of these exchanges. Nevertheless, they account for more than one-third of the working time.

Finally the Individual activity is fully autonomous and can be probably done in remote location. We still have to find the extent of the relationship between the Individual activities and the Ad hoc collaborative activities.

A practical conclusion is that people need to communicate in a team project. These communications take various forms and channels. We found that the formal channels expressed by the Mandatory and Called collaborative activities are not the most important in terms of time spent. It seems that the Ad hoc collaborative activities may play a major role in team communications dynamics. First, it is important in terms of time spent in this channel of communications: on project basis it accounts for 41% of the activities and on personal basis it is often the longest working session. Second, it seems to have a significant impact on the individual activities: they are mostly done by two teammates and often precede long individual working session.

This study, based on the observation of teamwork, is one of the first steps in trying to understand the dynamics of teamwork. We believe that such studies are a prerequisite to implementing distributed teamwork based on Internet tools. This study also stresses the point that face-to-face meeting can have various forms and formal meetings are only one component of teamwork.

We all agree that the people are the most important assess of software development project, however we know little about the mechanism and the value of the people interaction.

Acknowledgements

We are grateful to DMR Consulting Group Inc. for providing the industrial project. This work was supported in part by National Sciences and Engineering Research Council of Canada under grant A0141.

References

Adam, A.P., Johnson, P.M., 1997. Assessing software review meetings: results of a comparative analysis of two experimental studies. IEEE Trans. Software Engrg. 23 (3), 129–145.

Ancona, D.G., Caldwell, D.F., 1990. Information technology and work groups: The case of new product teams. In: Kraut, R.E., Galegher, J., Egido, C. (Eds.), Intellectual Teamwork: Social Foundations of Cooperative Work. Lawrence Erlbaum Associates, London.

Bendifallah, S., Scacchi, W., 1989. Work structures and shifts: An empirical analysis of software specification teamwork. In: 11th International Conference on Software Engineering. pp. 260–270.

Bradac, M.G., Perry, D.E., Votta, L.G., 1994. Prototyping a process monitoring experiment. IEEE Trans. Software Engrg. 20 (10), 774–784.

Burke, K., Chidambaram, L., Locke, J., 1995. Evolution of relational factors over time: {A} Study of distributed and non-distributed meetings. In: Proceedings of the 28th Annual Hawaii International Conference on System Sciences. Vol 4: Information Systems – Collaboration Systems and Technology Organizational Systems and Technology. pp. 14–23.

Christ, R.E., Muckler, F.A., The measurement and evaluation of collective unit training and performance. In: Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting, vol. 2, 1993, p. 1170.

d'Astous, P., Détienne, F., Robillard, P.N., Visser, W., 1998. Types of dialogs in evaluation meetings: an analysis of technical-review meetings in software development. In: International Conference on the Design of Cooperative Systems. Cannes. pp. 25–33.

Freedman, D.P., Weinberg, G.M., 1990. Handbook of Walkthroughs, Inspections and Technical Reviews, fourth ed. Little Brown and Company.

Johnson, P.M., Tjahjono, D., 1997. Assessing software review meetings: {A} controlled experimental study using {CSRS}. In: Proceedings of the 1997 International Conference on Software Engineering. pp. 118–127.

Marsan, A., 1990. Stochastic petri nets: An elementary introduction. In: Rozenberg, G. (Ed.), Advances in Petri Nets. Springer, Berlin, pp. 1–29.

McGrath, J.E., 1990. Time matters in groups. In: Kraut, R.E. Galegher, J., Egido, C. (Eds.), Intellectual Teamwork: Social Foundations of Cooperative Work. Lawrence Erlbaum Associates, Hillsdale, NJ.

Perry, D.E., Staudenmeyer, N.A., Votta, L.G., 1994. People organizations and process improvement: two experiments to discover how developers spend their time. IEEE Software 11 (4), 36–45.

Porter, A.A., Siy, H.P., Votta, L.G., Jr., 1997. Understanding the effects of developer activities on inspection interval. In: 19th International Conference on Software Engineering. pp. 128–138.

Robillard, P.N., Robillard, M.P., 1998. Improving academic software engineering projects: a comparative study of academic and industry projects. Ann. Soft. Engrg. 6, 343–363.

Robillard, P.N., d'Astous, P., Détienne, F., Visser, W., 1998. Measuring cognitive activities in software engineering. In: 20th International Conference on Software Engineering. pp. 19–25.

Robillard, P.N., 1995. Experience in teaching team software design. In: Sixth World Conference on Computers in Education. Birmingham, UK, pp. 441–453.

Robillard, P.N., 1996. Teaching software engineering through a project-oriented course. In: Ninth Software Engineering Education. pp. 85–94.

Robillard, P.N., 1999. The role of knowledge in software. Commun. ACM 42 (1), 87–92.

Seaman, C.B., Basili, V.R., 1997. An empirical study of communication in code inspections. In: Proceedings of the 19th International Conference on Software Engineering. pp. 96–106.

Seaman, C.B., Basili, V.R., 1998. Communication and organization: an empirical study of discussion in inspection meetings. IEEE Trans. Software Engrg. 24 (7), 559–572.

Software Engineering, 1994. IEEE Standards Collection, The Institute of Electrical and Electronics Engineers, Inc.

Pierre N. Robillard, Ph.D. is a full-time professor in software engineering in the department of Electrical and Computer Engineering at the École Polytechnique de Montréal in Montréal, Canada. His main research interest lies in software processes and the cognitive aspects in software engineering.

Martin P. Robillard, M.A.Sc. is completing Ph.D. studies in software engineering at the University of British Columbia in Vancouver, Canada.