Team Assistance in a Software Engineering Team: A Field Study

Pierre N. Robillard, Sébastien Cherry,
Department of Computer and Software Engineering
Polytechnique Montréal
Montréal, Canada
pierre.robillard@polymtl.ca
sebastien.cherry@polymtl.ca

François Chiocchio, Carolyne Hass
Department of Psychology
Université de Montréal
Montréal, Canada
chiocchio@telfer.uottawa.ca
Carolyne.hass@umontreal.ca

Abstract— **Physically collocated teammates often interact spontaneously while working solo on their assigned tasks. These ad hoc interactions could be perceived as counterproductive when they are seen as interruptions or they could be perceived as productive when they are seen as ad hoc team assistances, which contribute to the team awareness, trust amongst team members, and improved shared mental model. This paper reports on a field study performed in a professional environment. Team activities have been continuously video recorded over a period of two months. More than 400 ad hoc interactions have been analyzed. Ad hoc interactions required up to 30% of the team total time. These ad hoc interactions involve all the team members and as such may contribute to team awareness and improvement of shared mental model. Ad hoc team assistances can be categorized according to two purposes: the application domain or the development environment. This study shed light on the team dynamics of collocated teams and can provide insight into the challenges faced by the distributed software development teams. Suggestions are formulated for the management of team assistance activities.**

Keywords- Team process, team assistance, field study, ad hoc interactions, collocated team.

I. INTRODUCTION

Software development paradigms involved harnessing teammate interactions, which are the core of any team activities. Previous studies present overviews of the types of knowledge exchanges occurring during team interactions [1] and explore the roles of ad hoc interactions and the social side of software engineering [2]. Whatever the approach, members of ongoing collocated team engage in teamwork and taskwork. While teamwork refers to how team members work to combine their thoughts, actions, and feelings to coordinate and adapt, and to reach a common goal, taskwork refers to how team members interact individually with tasks, tools, machines, and systems [3][4]. Teamwork is often performed synchronously during scheduled or planned meetings, where team members interact in a shared activity. Typical examples are brain-storming sessions and design reviews [5]. Taskworks occur when teammates work solo on their assigned tasks. In a collocated software development environment, the task is often related to programming, debugging, or testing activities. During these solo activities teammates will nevertheless interact on an ad hoc basis.

Organizational psychologists have long been interested in the dynamics of team interactions and it may be wise for software engineers to capitalize on their expertise to better understand the dynamics of software development team. There is 50-year long tradition of studying helping behaviors in the industrial and organizational psychology literature [6][7]. These helping behaviors refer to a larger class of behaviors called Team Assistances, which could be defined as "individual behavior that is discretionary, not directly or explicitly recognized by the formal reward system, and that in the aggregate promotes the effective functioning of the organization" [8].

"By discretionary, we mean that the behavior is not an enforceable requirement of the role or the job description, that is, the clearly specifiable terms of the person's employment contract with the organization; the behavior is rather a matter of personal choice, such that its omission is not generally understood as punishable." [8]. According to Borman's model [9] of organizational citizenship performance these behaviors include "helping others by offering suggestions, teaching them useful knowledge or skills, directly performing some of their tasks to help out, and providing emotional support for their personal problems; cooperating with others by accepting suggestions, informing them of events they should know about, and putting team objectives ahead of personal interests; taking the initiative to do all that is necessary to accomplish objectives even if not normally a part of own duties, and finding additional productive work to perform when own duties are completed" [9, p 239].

The goal of the teammates is first to perform their tasks (i.e., taskwork, task performance) and to communicate on an as-needed basis within the context of an open space office with cubicles [10]. This communication will generate an interruption vis-à-vis the recipient, and all neighboring team members are likely to be aware of the interaction. In the context, these interruptions, which provide Team Assistances, are transgressions of an organizational norm and as such are not specifically prescribed. However, it is necessary a process ancillary to taskwork performed in a team room.

On the one hand, software engineering scientific and practitioner literatures tend to characterize ad hoc disruptions of taskwork as counterproductive interruptions and have not relied on pertinent evidence-based or theoretical models to explain or use them [11][12][13]. On the other hand, while organizational psychology has identified Team Assistances as performance-related interactions, studies in real work settings are rare, particularly regarding engineering software teams. Consequently, this study aims at gaining a better understanding of the nature, pattern, and content of Team Assistances that occur during taskwork time in software engineering teams. Second, and more specifically, we aim at examining the reasons why software developers need Team Assistance during taskwork time and what kind of knowledge is transferred during these interactions. A better understanding of these issues would provide a foundation for the study of software engineering team needs for ad hoc interactions and the speculative consequences of virtual or even absence of such ad hoc interactions on distributed software development teams. Such a study can also shed light on the appropriateness of specific practices such as occasional pair-programming [14], which can be seen as a special form of team assistance and the relevance of the use of collaborative tools for distributed team works.

In the next sections, we draw on literatures from Team Assistance to position our views in a broader context. We next describe the methods we used to analyze video recordings of software developers interactions in professional work settings. We then present our results regarding Team Assistance modality, purposes and content. These data are presented as a function of teammate roles (novice, leader, expert, developer). We conclude on the salient features of Team Assistance.

II. Team Assistance

Team Assistances are defined as "the discretionary provision of resources and task-related effort to another member of one's team that is intended to help that team member obtain the goals as defined by his or her role …" [15]. Essentially, Team Assistances can be seen as helping one's fellow teammates perform their role.

Team Assistances are central to the concept of adaptive team performance [17]. When one team member's task requires greater capacities than possessed, another team member can step in and compensate – the team is therefore adjusting on the spot and performing in a way not anticipated during the planning phases. These are complementary behaviors that arise either out of a specific request or merely from awareness on the part of one of the team members [17]. Unsurprisingly then, Team Assistances are a crucial form of interactions that allow a team to function as more than the sum of its individual members [18][19].

A. *Team Task Characteristics*

There is one caveat however. Team Assistances will arise out of a legitimate need for assistance resulting from issues with task assignment or task distribution problems [16]. The legitimacy of a need means that team members are experiencing true task difficulties beyond their capacity rather than a lack of effort [15][20]. Help provided because of social loafing or an unwarranted dependency need (when workload is in fact normal or low) is considered an illegitimate need for help and causes process loss and frictions [15][21][22]. Legitimacy of need is therefore the key situational factor that can affect the amount of Team Assistance requested or provided.

A study by Porter et al. [15] explored the personality traits of both Team Assistance recipients and providers in order to determine team composition characteristics related to the most effective use of Team Assistances. Their results showed that team members high on conscientiousness will receive more Team Assistance only when there is a legitimate need for it. These members are discriminate enough in their requests for assistance when it comes to the legitimacy of their need for it. Team members high on extraversion secured the most Team Assistances relative to members low on extraversion. There was a similar interaction effect between extraversion and legitimacy of need in terms of amount of Team Assistance received.

Porter et al. [15] also explored the personality traits most likely to lead to team members providing Team Assistances. They found that team members high on conscientiousness and emotional stability provided more Team Assistances to fellow team members, regardless of the legitimacy of need, compared to members low on these traits. Moreover, team members high on emotional stability provided even more Team Assistances if legitimacy of need was high, showing an interaction effect that the authors feel is critical to team composition. When team members are low on emotional stability, they are likely too self-focused to concentrate on the problems of fellow members, and will leave them to fend for themselves regardless of legitimacy of need.

B. *Shared Mental Models*

Team Assistance first and foremost requires that team members possess accurate knowledge of each other's responsibilities. Shared mental models in teams form the grounds on which team members know when to step in and provide Team Assistance, which team member should provide it, and what kind of Team Assistance is needed [16]. A team that possesses a shared mental model can anticipate and predict the needs of fellow members through a common understanding of team goals and expectations of performance. Shared models create a basic framework that promotes common understanding, as well as common action -- that is, a team that is headed toward the same goals [16]. They are particularly important in cases where a need for assistance is not initiated by a help request from the Team Assistance recipient – the need is anticipated by the Team Assistance provider, because of the shared mental model that allows predicting needs that may not be expressed [15].

Members must be willing and able to back up their fellow members – that is, they must be first aware that there is a task problem, but must also be competent in the areas of other members in order to be able to recognize when a member has problem with his/her task. The team member will have the knowledge and ability to step in and provide compensatory behavior when a fellow member finds himself with task problems [16].

C. Team Assistance Impact on Teamwork and Team Performance

Team Assistance has direct positive effects on task performance in a team context [15], particularly in a high legitimacy condition, where the task that is causing a member to need assistance is critical to the team's performance. However, if this is not the case, Team Assistance may in fact hinder adaptive team performance by providing a behavior that is redundant [20].

The relationship between team assistances and team performance is said to be mediated by the team's ability to adapt to changes internal and external to the team (the change in environment that would lead to a workload distribution problem) [16]. Team Assistances are essential to the planning phase of teamwork, since they are demonstrative of a team's ability to adapt or revise their coordination processes if needed. This flexibility when executing teamwork plans greatly facilitates adaptive team performance in unpredictable or ever-changing contexts [17]. Teamwork is characterized by dynamic, adaptive and flexible interrelated behaviors and actions. That is, members must be able to adjust the timing of their actions and their strategy quickly in order to meet the demands of other members. This leads to – or explains the need for – coordinated and synchronized collective team action [4][19].

The preceding discussion leads to the following conclusions and research questions. Interactions during taskworks can be seen as counterproductive interruptions [23] or, as we have shown using models and empirical findings from organizational psychology, can be construed as productive Team Assistances. To explore this possibility in ongoing software development teams, we will seek to answer the following questions:

1. Are Team Assistances naturally present in software engineering teams?
2. What are the modalities of Team Assistance?
3. What are the reasons for Team Assistance?
4. What kind of knowledge is transfer during Team Assistance?

III. Field Study and Methodology

The purpose of this field study is to characterize the Team Assistance activities within a software maintenance team from an organization providing general business applications.

A. Field Study Description

This international organization has several thousand developers in many countries. In spite of the size of the organization, the setting has the attributes of smaller organizations, as development is shared among several small teams of up to 15 members each, often located at a single site. The small team observed in this field study is a stable team, whose members are used to work together and who are familiar with their tasks. There is no known conflict between the teammates and they have a respectful attitude. The four observed participants, who are all males, were part of a larger team composed of 12 individuals (1 project manager and 11 software developers) ranging widely in age, with varying levels of schooling (from a Bachelor's degree to a Ph.D. in the computer sciences and engineering), and individual experience ranging from 2 to 16 years in the field and from 9 months to 5 years of service in the company. They used a companywide software development process that is largely inspired from the waterfall model.

Physically, the participants occupied individual adjacent cubicles separated by semi-transparent walls a meter and half high. From their desks they can see whether or not their neighbors are present. Monadic (F1F) interactions occurred when participants communicate while seated at their desks. Dyadic (F2F, i.e., Face-to-Face) interactions involved two participants and they occurred when there is a movement of one of the teammates (the recipient or the provider) toward the cubicle of the other most often to gain access to an artifact. Polyadic (FnF) interactions involved more than two (n) participants and they are mostly built up from dyadic interactions. Most of the time, someone who is aware of a dyadic interaction will join his teammates to add his comment to the ongoing interaction. Dyadic and polyadic interactions required that at least one of the participants physically moved from his cubicle to another location, which was one of the other participant's cubicle most of the time.

The observed participants are described in terms of the role that each of them occupies within the team. Based on previous studies on social aspects of software engineering with this data, the roles of the four participants are the following [2][25]:

- Leader: the project manager who occupies the formal leadership position.
- Expert: the individual who is responsible for configuration management of the software built by the team, his informal leadership being rooted mostly in his knowledge and expertise.
- Developer: an individual who has no specific role on the team, formal or informal, who can be seen as the embodiment of an average developer.
- Novice: the recruit software developer who has been with this team for six months.

All procedures for these observations were approved of by independent ethics committees of both the participating organization and our University prior to the study and by each of the team members who agreed to participate in this study on a voluntary basis.

B. Recording Set-up

Video equipment was installed in the ceiling over the work area and microphones set up on various places within the working environment. Data were taken from continuous video recording during the working hours excluding lunch time. A recording session begins either in the morning or in the afternoon, and lasts half a day, with a typical duration of 2 to 3 consecutive hours. A regular session is defined as a session where all teammates are present and where there are no special events, such as meetings, visitors, etc., which could disturb the usual task work. We retained 12 regular half-day sessions from the 23 recorded sessions. These selected sessions are evenly distributed over the two months of the recording time and account for 35 hours of video recording.

One of the researchers was a participant on the team. He was hired as a full time software developer six months prior to the study. He is identified as the Novice in this study. The purpose of being involved as a team member was to acquire the knowledge and the jargon used by the teammates in order to be able to subsequently analyze the data collected. A second researcher, who was not involved as a participant, but who had lead software development projects and has experience in video analyses, could objectively validate the observations made. Coding of the team interactions from the 12 sessions resulted in 404 Team Assistance occurrences. There was almost no e-mail exchanged between the team members, except for forwarding artifacts.

C. *Team Assistance Purposes*

We distinguish two general purposes for Team Assistance: one is to provide help to a teammate to perform his/her task and the other is to share the task with that teammate. The two purposes of Team Assistance are categorized according to the following definitions:

(1) Cooperation purpose [26] [27]: is providing feedback and coaching to increase performance. It categorizes sequences that take place when individuals provide help, but not necessarily for mutual benefit. It is characterized by informal relationships that exist without a common mission, structure, or effort. Information is shared as needed. For example, typical cooperation activities are: informal code checking, helping a teammate to set up his environment, or with a debugging task.

(2) Collaboration purpose [28]: is sharing task with a teammate. It categorizes sequences that take place when two teammates work together at an intersection of common goals, and do so by sharing knowledge, by learning, and by building consensus. This form of Team Assistance is usually an on-demand activity performed by two team members who want to work together on a specific task. Examples of collaboration are: a shared design session, and brain-storming sessions. All the collaborators have a genuine interest in the activity. We categorized only unscheduled Team Assistance collaboration sequences

These two purposes for Team Assistance can occur on various types of content, which could be related to the application or the development environment. The content of the sequences has been thoroughly studied to determine a categorization scheme for the various topics discussed. During the recording period, the team worked on 7 specific issues. To identify each of these interactions from one of these issues would make the characterization idiosyncratic and irrelevant outside this very specific field study. It was found that a more useful approach would be to define generic topics that are likely to be relevant in any software development studies. A thorough analysis of the team's project and team' interactions yielded two topic categories that were later validated successfully by the three coders.

1. Application domain related topics are associated with specific aspects or features of the software product; for example, functionality, a software component, etc. The content of the Team Assistance is based on some understanding of the application to be developed.

2. Integrated Development Environment (IDE) related topics are associated with specific aspects or features of the development environment and tools, which do not relate to the application domain, for example, programming concepts, development environment features, configuration management issues, etc.

IV. RESULTS

The results from the analyses of the 404 interaction sequences found in the 12 recorded sessions have shown that ad hoc team assistances are naturally present in collocated software engineering teams. The four observed participants spent more than a quarter (28%) of their time on cumulative Team Assistance, which in this study accounts for a total of almost 2 hours and 20 minutes per 8-hour workday per participant. The rest of the time (72%) was spent mostly on taskwork performed solo. This data support that Team Assistances are naturally present in software engineering teams. The following presents the answered resulting from this field study for each of the following three questions.

2. What are the modalities of Team Assistance?
3. What are the reasons for Team Assistance?
4. What kind of knowledge is transfer during Team Assistance?

A. *Modalities of the Team Assistances*

Fig. 1 illustrates the three modes of interaction observed during Team Assistance. All recorded Team Assistances are face to face (FtF) verbal interactions. Monadic (F1F) interaction, which we recalled, occurred when one participant communicates while seated at his desk, account for 12% of the total Team assistance occurrences. Dyadic (F2F) interactions account for 82% of all the Team Assistance recorded. Polyadic (FnF) interactions account for only 6% of all face-to-face interaction. Each of these three modes of Team Assistance behavior filled up different objectives.

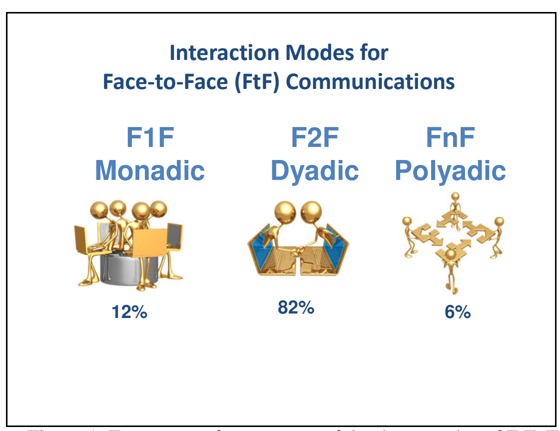

Figure 1. Frequency of occurrence of the three modes of FtF Team Assistance: Monadic (F1F), Dyadic (F2F), and Polyadic (FnF).

It has been observed in this field study that monadic interactions have short duration and contribute mostly to team awareness. For example, a teammate will state loud that he has completed the test procedure. Dyadic interactions occurred spontaneously during task work and they are truly a form of team assistance. We found that polyadic interactions last longer and are most often followed up of dyadic interactions.

Fig. 2 presents the relative frequency of the involvement of each of the four roles into each of the three modes of interaction. For example, the leader was involved in 25% of all the observed dyadic interactions (first column of the F2F mode). The Expert was involved in more than one-third of all the dyadic interactions. The total participation frequencies do not add up to 100%, because there is more than one participant for all of the interactions, except for monadic interactions (F1F).

Dyadic interaction (F2F) is the preferred mode of Team Assistance. We observe that monadic F1F interaction frequency increases when participants 'cubicle' are closer to one another. In this set up, the Developer had a central situation, he was sitting closer to the Expert than to the Leader, and the Novice was the furthest away.

B. *Reasons for Team Assistances*

Who are the initiators of interactions?

Does everyone initiate them occasionally, or only a few individuals do so?

Fig. 3 shows the relative frequency of interaction initiations for the four teammates who have been observed on a full-time basis.

The novice (27%) and the expert (34%) are the more frequent initiators of interactions but for different reasons. The novice was recruited on the team to add resources on various tasks but also because he had good knowledge on networks and server environments. More than 60% of his involvement in Team Assistance was initiated by him to obtain help on the understanding of the component functionalities while in 40% of his Team Assistance involvement he was as a provider of help on servers and network topics. The novice initiated interactions because he needed help for completing his task.

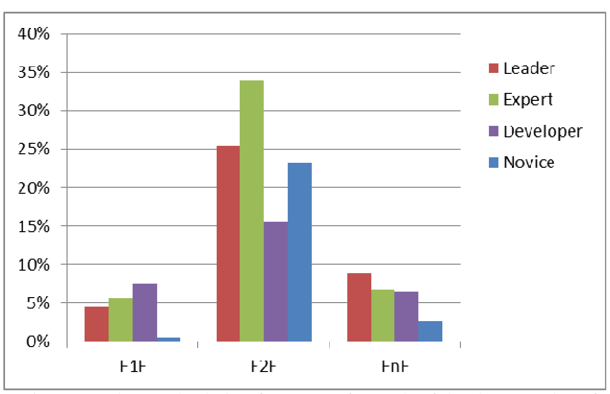

Figure 2. Observed relative frequency for each of the three modes of interaction for each role.

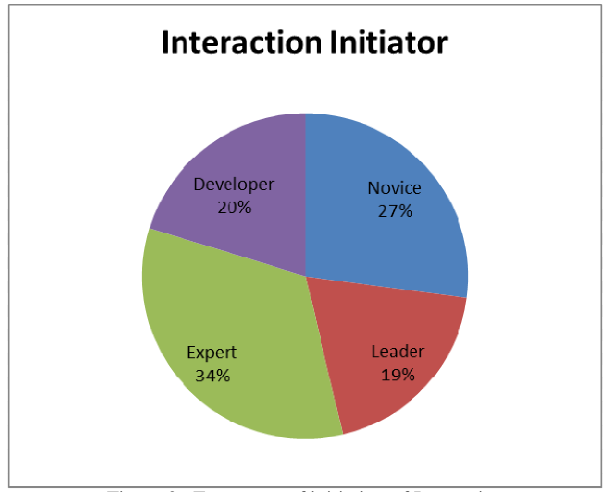

Figure 3. Frequency of initiation of Interactions.

In this context, it may sound surprising that the Expert initiated most of the requests for Team Assistance. The initiator of the interaction is not necessarily the candidate that needs Team Assistance as one could expect. A detailed analysis of the Expert interaction initiations revealed that the Expert was initiating some of the interactions for the purpose of following up on previous requests. Two cases were frequently observed. On case occurred when the Expert initiates Team Assistance interaction to provide the help that a teammate had requested earlier when the Expert cannot interrupt his work. A second case occurred when the Expert initiates interactions to follow up on previous team assistances help that were provided. He wants to make sure that the help was useful and that the recipient can proceed with his task and if needed provides additional information. That behavior was reported by the researchers in team process, as described in the previous section on shared mental model, that the teammates that have a have a high level of shared mental model can anticipate the needs for Team Assistance [15].

The Leader initiated 50% of the Team Assistance in which he was involved. The leader initiated interaction most of the time to provide information that will help the recipient. Typical cases were changes in configuration management, shared information on requested modifications to the software components. In some cases, he initiated interactions because he needs help to understand a component or the state of progress on a task. The leader was also an experienced member of the development team and he was the provider of information in most of the interactions, which he did not initiate.

The Developer initiated only 30% of the Team Assistances in which he was involved and he was exclusively a recipient concerned by technical subject related to his task. A quarter (25%) of the Team Assistances in which he was involved, as provider, had been initiated by the Expert as followed up.

C. *Kind of Knowledge in Team Assistances*

Fig. 4 shows the cumulative relative duration for each category of topics for collaboration and cooperation purposes. For example, it shows that more than 60% of the time spent was on collaborative Team Assistances (see left column in Fig. 4). Most of the collaborative Team Assistance (46%) is required to solve problems related to the application domain. Since the two teammates are working toward the same goal, these Team Assistance activities contribute to shared mental model. It is observed that cooperative Team Assistances, which account in this study for almost 40% of all Team Assistance activities (see right column in Fig. 4), are mainly required to solve IDE problems and very little, less than 10% of cooperative Team Assistance activities, are undertaken to solve application related problems. We recall that all the Team Assistance activities account for almost 30% of the total time the team spent in the team room and all participants are involved at almost the same level (see Fig. 3) but in different ways as explained in the previous section.

V. Discussion

This section discusses the outcomes of this observational study and shows how these outcomes can help understand the mechanics of team assistance. The threats to validity and reliability of such study are also discussed.

A. *Summary of Results*

A first observation is that all team members are almost equally involved in all three interaction modes. We found that ad hoc Team Assistance is a natural phenomenon that required almost the third of the time spent by the team members during their solo taskwork. These behaviors consist mostly of dyadic face-to-face interactions where one of the team members will visit a teammate cubicle.

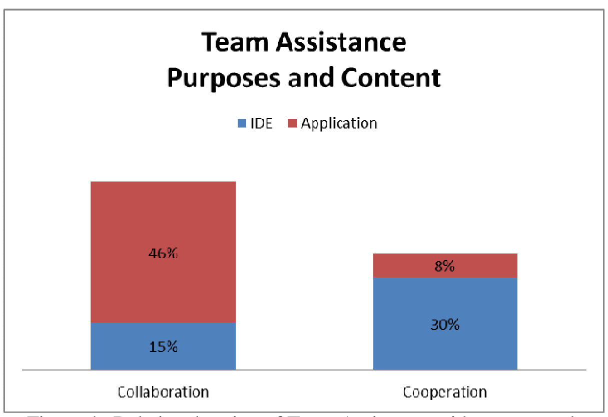

Figure 4. Relative duration of Team Assistance with respect to the purposes and the topics.

The expert role initiates almost the third of the Team Assistance interactions followed by the novice role. Although their reasons for initiating Team Assistance are different. The Novice initiated Team Assistance to obtain help on various tasks while the Expert initiates Team Assistance to follow up on requested assistance by teammates. It is noteworthy that the Leader is the one who initiated the less Team Assistance interactions.

Team Assistances are initiated for two purposes: collaboration or cooperation. Collaborative Team Assistances involve teammates sharing the same objective in assuming their tasks. Collaboration occurred mostly for increasing understanding of the application for the two teammates involved. Collaboration occurred in IDE context when developers worked together to install a server feature, for example.

Cooperative Team Assistances require that the provider teammate helps the recipient on subjects that are not immediately in line with the provider interest or task. Cooperation occurred mostly in the IDE context when teammates needed help with the configuration management systems or the debugger, for example.

B. *The Mechanics of Team Assistances*

Our observations point out to Team Assistance as an opportunistic behavior used by all participants in a co-located team. We found that each of the communication modes has a distinct purpose.

The monadic mode (F1F) contributes to maintaining team awareness. Team awareness involves knowing what activities teammates are working on and how they relate to individuals' own tasks. It allows teammates to informally communicate and coordinate their work. Burke et al. [17] explain that teams adapt to the extent that they assess the situation, formulate a plan, execute the plan, and learn from this process. In line with media richness theory [29] co-location affords teammates more opportunity for cue recognition and higher quality meaning ascription. In distributed, as well as in collocated teams, the monadic mode (F1F) can be easily computer-mediated by providing a kind of instant messaging system, where each teammate can post information judged to be valuable to maintain team awareness. The advantages of computer-mediated F1F are to avoid the interruptions caused by someone talking aloud and probably more important is the possibility of keeping track of all the messages sent.

The dyadic mode (F2F), which occurs when one teammate moves from his cubicle to communicate verbally with another, may contribute to team efficiency via what Borman [9] described as helping behaviors (i.e., citizenship behaviors). It is an opportunistic, just-in-time interaction initiated by a recipient teammate who needs information to continue his task or by a provider teammate who wants to validate help that was provided before as in the case of the Expert in this study. The degree of team efficiency, where one individual receives help and the other, who is the provider, is being interrupted, depends on the impact of the interruption on the provider. In a team room, a physical or a numerical device, such as a flag, can be raised to indicate that someone does not want to be interrupted momentarily.

The technical e-forum is a kind of asynchronous virtual F2F. A developer asks a question on the Forum, expecting that someone will answer it. The efficiency of the team room derives from the fact that the ad hoc communication is synchronous (the answer is immediate), and it involves trusted and aware co-workers. It has been observed in this study that teammates will always prefer F2F to e-mail, within a team room.

There is a great deal of research on the difficulties involved in computer-mediating F2F communications. A verbal dialog not only allows participants to assess their understanding, but also to develop a sense of community with teammates. Most studies comparing F2F and computer-mediated communications are related to the educational environment (for tutoring) or the planned meetings. These findings cannot be readily applies to opportunistic ad hoc F2F interactions. These interactions are usually very short, and based on team awareness and the role that each teammate plays in the project. More observational and experimental studies are needed to evaluate the effectiveness of computer-mediating Team Assistance on the form of ad hoc F2F interactions, which is still the major feature of the collocated team.

The polyadic mode FnF, which occurs when an ad hoc Team Assistance involves many teammates, seems to contribute to the solution of environmental or application problems. This mode is often initiated from the dyadic mode when some issues cannot be readily resolved. When this happens, other team members may become involved and take the ad hoc Team Assistance into polyadic mode (FnF). We believe that when polyadic Team Assistance mode occurred the participants should schedule a meeting in a closed room, with only those participating who can contribute to the solution.

It is observed from this study that Team Assistance can be categorized from two purposes: cooperation or collaboration, which has been identified in independent studies [16]. Cooperation is characterized by providing help to the recipient for his own benefit, while collaboration is sharing the problem-solving task for mutual benefit. To increase the generalizability of these observations we consider the content in terms of information related to the application (like business rules) or to the development environment (IDE). It is observed that most of the collaboration occurred to increase mutual benefits of application understanding and most of the cooperation occurred to help teammates with their environment development. In terms of duration there are almost as much time spent on application understanding as on help on using the development environment.

C. *Threats to Validity*

Reliability and validity of the coding were assessed based on observations made on the 404 Team Assistance sequences extracted from the 12 recorded working sessions. The first step involved an intra-coder agreement, where a number of encoded data sequences were re-encoded a month later by the same coder. The second step involved an inter-coder agreement, where another coder who was able to understand the context and the jargon employed by the participants performed the same operation. Finally, the third step involved an extra-coder agreement, where an experienced coder who was not familiar with the team's work performed the same operation. An index proposed by Perreault and Leigh [24] was used to measure reliability. The inter-coder agreement indices obtained show a value of 0.89 between the two coders familiar with the team dynamics, enabling us to deduce a strong agreement. The indices obtained with the extra-coder agreement show a value of 0.72. These values suggest acceptable reliability of the coding and validity of the coding scheme. To avoid capturing behaviors that might be affected by workers' reaction to the recording equipment, interactions occurring in the first 4 weeks of the equipment's installation were not coded. Furthermore, interactions that were outside the range of cameras or microphones were deleted from the data set (n; 137).

VI. CONCLUSION

These observations confirm that Team Assistance is a core activity within a team dynamic, which may contribute to jell the team by increasing awareness, shared mental model and exchanges between the teammates. The nature of Team Assistance practices is complex and depends on various factors such as the role of the participants, the character of the individual, the physical location within the team set up, the taskwork, and the purpose and the content of the needed help.

Spontaneous interactions between collocated software developers may be perceived by practitioners and managers as undesirable interruptions that distract the developers from their tasks. However, we have observed that although it may be perceived as counterproductive interruptions, it is nevertheless a necessary – even naturally occurring – workplace behavior.

The following points are stressed based on our observational study of team dynamic:

- Interactions are legitimate, opportunistic, and of short duration;
- Almost 30% of the total team activity is devoted to spontaneous and just-in-time Team Assistance interactions
- All team members are involved in these interactions as recipients or as providers.
- Team Assistances are mostly collaborative for application domain and cooperative for development environment problems.

Our results show that Team Assistances occur without prescription from the team leader (i.e., they are ad hoc) and are an efficient means of just-in-time learning and adaptation in the workplace. It enables the initiator of the interaction to obtain quick access to information and then proceed with the task at hand. Although we did not frame our observational scheme in terms of longitudinal team development, Kozlowski et al.'s model [30] of team compilation would be an appropriate conceptual footing to examine this aspect in a future study. The compilation model argues that interactions progress from to dyadic to polyadic as people understand their respective tasks and roles.

This study was not an experiment where all the various parameters could be controlled. However, our analyses stem from reliable coding of multiple interactions that occurred over many weeks in a real-world working context. Although this study requires replication, our theoretical background and results are compelling. Our results suggest three practices to facilitate Team Assistance and that are likely to improve team dynamics and the success of the project.

First, developers use Team Assistance for 30% of their time meaning that they do not necessarily need help for the other 70% of the time. Team leaders would be better off ensuring that Team Assistances are encouraged as long as they represent a legitimate need, and work to understand and correct non-legitimate demands.

Second, Team Assistances are sought from a pool of providers, which indicates a choice of the best provider is made. Team leaders should make sure all team members understand where each other's talents rest so that legitimate help is sought efficiently from a competent provider (e.g., application information versus IDEs).

Third, Team Assistances can be collaborative or cooperative. While both purposes foster a shared mental model, they impact different aspects of software development. Team leaders would benefit from ensuring that the correct Team Assistances are used with the appropriate task requirement from the software development life cycle.

Our results show promising avenues for future studies. One avenue would be to document Team Assistances across more teams and more project phases. This would potentially underscore how context changes the nature and frequency of Team Assistances. A second avenue would be to test whether coaching from the team leader can help leverage the impact of Team Assistances. Hackman and Wageman [31] suggest a theory of team coaching that hinges on three components one of which is consultation on team processes. A leader that consults his/her team mid-way within a project phase is likely to identify if and how team members engage in Team Assistances. The theory predicts that such a consultation is likely to foster more efficient team process. A third avenue would be to measure performance such that the efficacy of Team Assistances can be assessed against mainstays of team performance such as proficiency, adaptability and pro-activity [32].

ACKNOWLEDGMENT

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks. This research was supported in part by NSERC grant A-0141.

REFERENCES

[1] P. N. Robillard and S. Cherry, "Types of knowledge exchange during team interactions: A software engineering study," The sixth International Conference on Information, Process and Knowledge Management, eKNOW 2014, pp. 131-136.

[2] S. Cherry and P. N. Robillard, "The social side of software engineering – A real ad hoc collaboration network," Int. J. Human-Computer Studies (IJHCS), Vol. 66, 2008, pp. 495-505, doi:10.1016/j.ihcs.2008.01.02.

[3] M. A. Marks, J. E. Mathieu, and S. J. Zaccaro, "A temporally based framework and taxonomy of team processes," Academy of Management Review, 26, 2001, pp. 356-376.

[4] F. Chiocchio, S. Grenier, T. O'Neill, K. Savaria, and D. J. Willms, "The effects of collaboration on performance: A multilevel validation in project teams," International Journal of Project Organisation and Management, 4, 2012, pp. 1-37.

[5] P. D'Astous and P. N. Robillard, "Empirical study of exchange patterns during software peer review meetings," Information and Software Technology, vol. 44, no. 11, 2002, pp. 639-648.

[6] S. W. J. Kozlowski and D. R. Ilgen, "Enhancing the effectiveness of work groups and teams," Psychological Science in the Public Interest, vol. 7, no. 3, 2006, pp. 77-124.

[7] S. J. Motowidlo, Job performance. In W. C. Borman, D. R. Ilgen, R. Klimoski, and I. B. Weiner (Eds.), Handbook of Psychology: Industrial and Organizational Psychology, vol. 12, pp. 39-53. London: Wiley. 2003.

[8] D. W. Organ, "Organizational citizenship behavior: The good soldier syndrome," Lexington, MA: Lexington Books. 1988.

[9] W. C. Borman, "The concept of organizational citizenship," current directions in Psychological Science, vol. 13, no. 6, 2004, pp. 238-241, doi: 10.1111/j.0963-7214.2004.00316.x

[10] A. J. Ko, R. DeLine, and G. Venolia, "Information needs in collocated software development teams," Proceedings of the 29th International Conference on Software Engineering, 2007, pp. 344-353.

[11] E. Arroya, T. Selker, and A. Stouffs, "Interrruptions as multimodal outputs: Which are the less disruptive?" in Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02), 2002, 4 pages.

[12] T. D. LaToza, G. Venolia, and R. DeLine, "Maintaining mental models: A study of developer work habits," Proceedings of the 28th International Conference on Software Engineering, 2006, pp. 492-501.

[13] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung, "An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks," IEEE Transactions on Software Engineering, vol. 32, no. 12, 2006, pp. 971-987.

[14] M. Ally, F. Darroch, and M. Toleman, "A framework for understanding the factors influencing pair programming success," in 6th International Conference on Extreme Programming and Agile Processes in Software Engineering, 2005, pp. 82–91.

[15] C. O. L. H. Porter, J. R. Hollenbeck, D. R. Ilgen, A. P. J. Ellis, B. J. West, and H. Moon, "Backing up behaviors in teams: The role of personality and legitimacy of need," Journal of Applied Psychology, vol. 88, no. 3, 2003, pp. 391-403, doi: 10.1037/0021-9010.88.3.391

[16] E. Salas, D. E. Sims, and C. S. Burke, "Is there a "Big Five" in teamwork?" Small Group Research, vol. 36, no. 5, 2005, pp. 555-599, doi: 10.1177/1046496405277134.

[17] C. S. Burke, K. C. Stagl, E. Salas, L. Pierce, and D. Kendall, "Understanding team adaptation: A conceptual analysis and model," Journal of Applied Psychology, vol. 91, no. 6, 2006, pp. 1189-1207, doi: 10.1037/0021-9010.91.6.1189

[18] R. M. McIntyre and E. Salas, "Measuring and managing for team performance: Emerging principles from complex environments," In R. A. Guzzo & E. Salas (Eds.), Team effectivness and decision making in organizations, pp. 9-45, San Francisco: Jossey Bass. 1995.

[19] E. Salas, C. S. Burke, and J. A. Cannon-Bowers, "Teamwork: emerging principles," International Journal of Management

Reviews, vol. 2, no. 4, 2000, pp. 339-356. doi: 10.1111/1468-2370.00046

[20] D. R. Ilgen, J. R. Hollenbeck, M. Johnson, and D. Jundt, “Teams in organizations: From Input Process-Output models to IMOI models,” Annual Review of Psychology, vol 56, 2005, pp. 517-543, doi: 10.1146/annurev.psych. 56.091103. 070250.

[21] I. D. Steiner, Group Process and Productivity. New York: Academic Press. 1972.

[22] C. M. Barnes, J. R. Hollenbeck, D. T. Wagner, D. S. DeRue, J. D. Nahrgang, and K. M. Schwind, “Harmful help: The costs of backing-up behavior in teams,” Journal of Applied Psychology, vol. 93, no. 3, 2008, pp. 529-539, doi: 10.1037/0021-9010.93.3.529

[23] Q. R. Jett, and J. M. George, “Work interrupted: A closer look at the role of interruptions in organizational life,” Academy of Management Review, vol. 28, 2003, pp. 494-507.

[24] W. D. Perreault and L. E. Leigh, “Reliability of nominal data based on qualitative judgments,” Journal of Marketing Research, vol. 26 May, 1989, pp. 135-148.

[25] J. Conny, P. A. V. Hall, and M. Coquard, “Talk to Paula and Peter -- They are experienced: The experience engine in a nutshell,” SEKE 2000, 2000, pp. 171-185.

[26] S. M. Hord, “Working together: Cooperation or collaboration,” Austin, TX: Research and Development Center for Teacher Education, University of Texas. (ERIC Document Reproduction Service No. ED 226 450), 1981.

[27] J. Roschelle, and S. D. Teasley, “The construction of shared knowledge in collaborative problem solving,” In Computer Supported Collaborative Learning, C. E. O’Malley, ed. Springer-Verlag, 1995, pp. 69-97.

[28] P. Dillenbourg, “What do you mean by collaborative learning?” In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches, Elsevier, Oxford, 1999, pp. 1-19.

[29] R. L. Daft and R. H. Lengel, “Information richness: A new approach to managerial behavior and organization design,” Research in Organizational Behavior, vol. 6, 1984, pp. 191-233.

[30] S. W. J. Kozlowski and K. J. Klein, “A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes,” In K. K. Klein & S. W. J. Kozlowski (Eds.), Multilevel theory, research, and methods in organizations, pp. 3-90. San Francisco: Jossey-Bass. 2000.

[31] J. R. Hackman and R. Wageman, “Asking the right questions about leadership,” American Psychologist, vol. 62, 2007, pp.43-47.

[32] M. A Griffin, A. Neal, and S. K. Parker, “A new model of work role performance: positive behavior in uncertain and interdependent contexts,” Academy of Management Journal, vol. 50, no. 2, 2007, pp. 327-47.