Software Team Processes: A Taxonomy

Pierre N. Robillard, Mathieu Lavallée
Software Engineering Research Lab
Ecole Polytechnique de Montréal
Montréal, Canada
{pierre.robillard, mathieu.lavallee}@polymtl.ca

***Abstract*—A software development team must integrate many process perspectives imposed by the client, the organization, team management, and team dynamics. All these perspectives intermingle at the team process level. We propose a taxonomy to define the various episodes observed at this level and a useful vocabulary for reporting the observations made. Our taxonomy is based on a previous literature review performed in the business management field and adapted in this paper for the software engineering field. It was applied on existing data obtained from a real project carried out with an industrial partner. The vocabulary defined can accurately present team interactions, and may help diagnose communication breakdowns leading to project failures. The benefit of this approach is that it enables team process activities to be measured without reference to specific concepts related to perspectives other than that of the team process.**

***Keywords*-team process; teamwork; task work; software development process; developer; communication; taxonomy**

I. Introduction

A software development team must integrate many process perspectives imposed by the client, the organization, management, and team dynamics. The client brings the vision and the added value to the software product. The organization is concerned with the capability and maturity levels of the product, as well as other certification requirements. The project manager prefers to rely on proven management approaches like Scrum or a computer-mediated work breakdown structure. The development team needs to share knowledge and information, and is likely to adopt a well defined software design methodology and development environment.

Each perspective – see Figure 1 – imposes its own process and definition of the way software development should be performed, based on the priorities and agenda of that perspective. This paper focuses on the processes imposed by the team. However, to model and measure these "team" processes, we first need a process model that is specific to team activities and unrelated to the other perspectives. For example, measuring team activities in terms of Scrum management practices is biased toward the project and organizational perspectives. The resulting measures might be difficult to generalize to teams outside the Scrum context. Our objective is therefore to define a taxonomy that is capable of defining team processes independently of processes imposed by the client, the organization, or the project.

This does not mean that team processes are developed independently from management, organization or client perspectives. For example, an organization might impose outsourced programmers to the development team, coercing the team to adjust its communication channels to the context. To reuse the vocabulary of the psychology field, these "contextual contingencies" create "situational demands" to the task at hand [1]. Team processes must be adjusted to these demands in order to provide "team effectiveness".

Interest in team process mainly derives from the fact that most software development work is performed by teams, which are composed of "*people working together to achieve something beyond the capabilities of individuals working alone*" [2].

Also of interest is the fact that team processes are sandwiched between the needs of individuals and the needs of the organization. A gap is reported to exist between the two [3]. How team members share information and knowledge must therefore involve reconciling management's directives imposed from above and the developers' initiatives, also called empowered actions [4], coming from below.

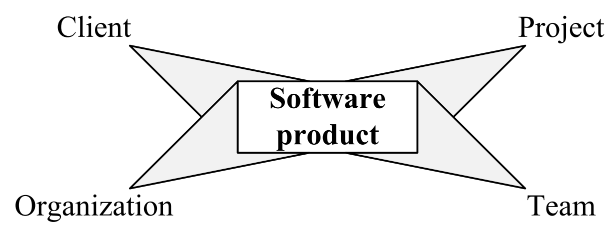

Figure 1. The software product as seen from the various process perspectives.

A. The Process Perspectives in Software Engineering

Every process perspective comes with particular objectives. Table I shows the various perspectives along with their needs and objectives.

A business model reflects the needs of the client and takes into account the context in which the client enterprise conducts its business. This context can be military, avionics, business, gaming, medical, government, etc. Their objectives involve ensuring the integration of the software product being developed into the target context, and complying with product certification requirements and market timelines. These considerations impose a number of practices on software development projects. Product certification, traceability, testability, and norm conformity are examples of business model requirements.

The organizational perspective describes the enforcement of software development organization's practices. The organization's objective is to ensure that the capability and maturity levels of software development projects enable it to maintain its competitiveness in the industry. This includes obtaining software development process certification based on reference models such as CMMI or ISO/IEC 15504, or the appropriate ISO/IEC 9000 standards.

The project management perspective is concerned with the day-to-day management of the software project and official team building activities. Their objective is to ensure optimal use of the assigned human resources, within budget and respecting planned schedules. Project managers are responsible for supporting good team practices, which describe the characteristics of an effective team. The objective of these practices is to help team leaders determine what their team should do, and to evaluate the quality of their current practices. The Team Software Process (TSP) constitutes an example of the qualities of an effective team in software engineering (SE) [5].

Team processes describe how the individuals within a team are organized. The objective of the individuals in a team setting is to define effective communication channels to ensure appropriate sharing of knowledge and information among team members. These channels depend on the knowledge and information needs of each team member, and can therefore be very different from the official channels defined by management. While there is some research in the SE field on this perspective, there is no widely recognized standard model of communication channels. However, the concept of the team process is well developed in the psychology and business management fields [2] and can be adapted to SE teams.

B. Task Performance and Contextual Performance

Teamwork studies in the psychology and business management fields outline a difference between task performance and contextual performance [6, 7].

Task performance is related to teamwork activities typically sanctioned by the organization, called *role-prescribed* activities [7]. Task performance is related to skills usually evaluated by performance reviews.

Contextual performance, also called *organizational citizenship behavior*, is related to teamwork skills outside the abilities prescribed by the role [7]. This includes the inclination to help a colleague with a problem, to answer questions by coworkers from another department, to help a teammate suffering from a personal problem, etc. These altruistic qualities are rarely evaluated in performance reviews.

Studies have however shown that both task performance and contextual performance are important in order to measure team efficiency [6, 7]. The measure of both factors is however difficult, because they are not directly observed by managers. However, by recording team process episodes, we can find both task-performance related and contextual-performance related communications. The study of team processes can therefore assist managers with differentiating good team players from bad ones.

C. Structure of this Paper

This paper focuses on team processes, also called team dynamics [8]. Our goal is to define the main elements of these processes that could be applied independently of the various perspectives. The concept of the team process has been referred to previously in the SE literature [9], but, as far as we could determine, it has never been clearly defined.

The team process taxonomy described in this paper is an adaptation to the SE domain of the generic team process taxonomy of Marks et al. [2]. The generic elements relevant to our taxonomy are described in section II. Section III explains the reasons why this generic taxonomy should be adapted for the SE domain. Section IV describes our SE team process taxonomy. Section V presents real examples of the application of the taxonomy to a team process observed in the field. Section VI presents our conclusions and considerations for future work.

II. The Generic Team Process Taxonomy

Team processes are made up of the interactions between teammates. Analysis of these interactions can be helpful in understanding project failures, which are identified in the literature as problems of cohesion and conflict in teams [10], [11]. One of the key qualities of developers is their ability to communicate their knowledge to their teammates [12]. What team processes demonstrate is that communication abilities are worthless without communication channels to convey these abilities. For example, team process barriers like distance or language can impede communications

TABLE I. PROCESS PERSPECTIVES IN THE SOFTWARE DEVELOPMENT ORGANIZATION

Perspective	Description	Objectives	Application examples
Client business model process	Business goals, product certification, budget, and schedule	Product integration into an existing environment What criteria must the product meet?	Military, avionics, medical, business
Organizational process	Enforcement of practices in accordance with certification	Capability and maturity levels	CMMI, ISO/IEC 15540, ISO/IEC 9000 series
Project Management process	Management of team activities for successful project completion	Optimal utilization of resources, within budget and on schedule	Scrum, Lean, RUP, TSP
Team process	Interaction among team members	Effective communication channels for optimal sharing of knowledge and information	Psychological and business management models

sufficiently to make normally successful teams suddenly experience project failure [13].

Observing team processes makes it possible to evaluate the interactions of individuals independently of management efforts. For example, a project management process, like Scrum, defines activities for managers, but none for developers [14].

To observe these processes effectively, two elements are necessary: (1) we must know what to observe; and (2) we must have a defined vocabulary in which to report our observations. This is our objective with this taxonomy.

Interest in team processes stems from the following fact:

"Success is not only a function of team members' talents and the available resources but also the processes team members use to interact with each other to accomplish the work." [2]

This taxonomy defines a common vocabulary for modeling time-based processes between individuals in a team setting. The Marks et al. taxonomy [2], which serves as the main model for our work, is based on an extensive literature review of the existing models in the psychology and business management fields.

A. Phases and Episodes

Like most models in the domain, Marks et al. separate task work, i.e. "*what it is that teams are doing,*" from teamwork, i.e. "*how they are doing it with each other*" [2].

The Input-Process-Outcome (I-P-O) framework [15–17] is the basic component of the team process, where input artifacts are transformed into output artifacts, through steps defined by a specific process. From a given set of inputs, one or more processes can be applied before the outcomes are produced. A complete I-P_1-P_2-...-P_n-O chain of events is called an *I-P-O cycle*.

The element of work in a team process is called an *episode*. Each episode is defined as a single occurrence of an I-P-O cycle during a specific period of time. Episodes must be associated with one or more evaluable goals. Their length can vary, with a project manager's task potentially containing many episodes of different lengths. Episodes form chains: the outcome of one episode becomes the input for the following one.

Marks et al. describe three different groups of teamwork episode, called *phases,* which form the first level of their taxonomy: *transition phase* episodes, *action phase* episodes, and *interpersonal phase* episodes.

Transition phase episodes are defined as "*periods of time when teams focus primarily on evaluation and/or planning activities to guide their accomplishment of a team goal or objective*." This includes evaluating past performance and forecasting the goals of future episodes.

Action episodes are defined as "*periods of time when teams are engaged in acts that contribute directly to goal accomplishment*." The goals of action episodes are therefore closely related to the objectives of the task work. However, action episodes describe how the information and knowledge are transferred between team members (teamwork), and not how the individual team members perform the task (taskwork). The transition and action phases typically alternate: transition episodes plan the subsequent action episodes, whose outcomes are analyzed in another transition phase which plans the next action phase, and so on.

Interpersonal phase episodes are concerned with the control of emergent states, which are described in the next section. The objective in these episodes is to keep individual motivations and emotions, as well as interpersonal conflict, under control.

B. Emergent States

Emergent states are not episodes, as they cannot be defined with goals and time constraints. Rather, they represent the dynamic properties of the team, like motivation and values. The relationship between emergent states and team processes is described as follows:

"*Teams with low cohesion (an emergent state) may be less willing to manage existing conflict (the process), which, in turn, may create additional conflict that lowers cohesion levels even further. [...] Emergent states do not represent team interaction or team actions that lead toward outcomes. Rather, they are products of team experiences and become new inputs to subsequent processes and outcomes.*" [2]

By analogy, an emergent state in the telecommunications field constitutes noise affecting the inputs of I-P-O episodes. If the emergent state is under control, its effect on the outcomes of the episodes will be negligible. If the emergent state is out of control, it can significantly garble the inputs and thus interfere with the quality of the outcomes.

The emergent states can also be beneficial, as seen with transactional memory [1]. Transactional memory refers to the degree in which team members are aware of the knowledge and capabilities of their teammates. Interpersonal episodes can occur to enhance this transactional memory, but it represents a state of the team instead of an event.

Therefore, the goals of interpersonal episodes are to enhance positive emergent states like transactional memory, as well as to dampen negative emergent states, like interpersonal conflicts.

C. Relevance of I-P-O Cycles

I-P-O cycles have come under fire recently [18], because they do not represent the dynamic aspect of team processes, the feedback loop between outcome and input, and the presence of broken cycles.

Team processes are dynamic because they change throughout the project, because of the evolution of the task, but also because of the evolution of the team [18]. Some communication channels can appear or disappear during the project as team members are more accustomed to each other's abilities and personalities. The team process itself can be seen as an emergent state, which changes constantly throughout the project to follow the mold of the team.

The outcomes of the episodes can also influence the emergent states [18]. For example, a series of poor outcomes produced by a team can result in a drop in morale, an emergent state. This drop in morale will affect the inputs of the next episodes, for example by sowing suspicion between teammates. This creates a feedback loop between outcomes and inputs which can have a significant effect on the episodes' success.

I-P-O events do not always form a complete cycle [18]. For example, an information retrieval process can be broken down in multiple episodes, sharing the same input and producing the same outcome. The retrieval can fail, which would result in no outcome being produced. The cycles can therefore be incomplete, or can only be complete across multiple observations.

Despite these critiques, we believe that I-P-O cycles are still relevant in the study of team processes, as long as they are not viewed as static structures. I-P-O episodes must therefore be observed in their context in order to draw conclusions: A planning episode performed by a newly formed team at the beginning of a new project will be different from the one performed by the same team at the end of the project.

III. Adapation to the Software Engineering Domain

The team process taxonomy developed by Marks et al. is intended to be generic enough to be applied to any type of team: emergency response, marketing, brainstorming, etc. By briefly looking at some characteristics of these teams, we can see the need to adapt the taxonomy to the specific SE context [19].

Emergency response teams, for example, are concerned with brief interactions occurring during a short period of time, usually among members who have jelled as a unit. Personal conflicts are typically set aside and resolved once the emergency is over. Communication channels and team interactions are defined before the event, and can be difficult to redefine during an intervention. Team composition is also typically static during the emergency.

Marketing team projects are typically homogeneous. They are also more oriented toward design than functionality, and their requirements are typically short, but vague. Brainstorming teams share the same limitations, but, unlike marketing teams, they generally exist solely for the duration of a meeting.

Software development teams, by contrast, are often built at the beginning of a project that can span months, or even years. Team interaction and communication channels can change as needs arise, and the team may recruit new members as the project progresses. Members may also leave the team as the product's life cycle progresses and their expertise is no longer required. Software development teams are dynamic and heterogeneous, and their activities, which include design as well as construction, must meet specific requirements.

The original taxonomy, which is based on the psychology and business management fields, was designed to describe team processes only through the observation of teamwork, without considering taskwork at all. A theoretical view that excludes taskwork is too restrictive for the SE domain, because the causes of many teamwork episodes usually stem from difficulties encountered during taskwork activities. For example, a developer having difficulty removing a software bug may request help from a teammate. There is a need, therefore, to understand the relationship between taskwork and teamwork in SE.

All the elements of our adapted taxonomy should therefore be measurable in an SE project. The last section of this paper presents a case study that illustrates the measurement of episodes.

IV. Software Engineering Taxonomy of Team Processes

Figure 2 presents our taxonomy of SE team processes, and each of the subsections below describes a phase of the team process and its corresponding episodes. Each episode type is illustrated with examples observed in capstone projects.

We have enhanced the original taxonomy by including one phase defined by two episodes for taskwork. The three teamwork phases are part of the original model but the seven episodes have been adapted to the software engineering field.

- **Teamwork**
  - Transition phase
    - Backward evaluation episode (BE)
    - Forward visioning episode (FV)
    - Planning and prioritization episode (PP)
  - Action phase
    - Resource and progress monitoring episode (RM)
    - Team assistance episode (TA)
    - Information gathering episode (IG)
  - Interpersonal phase
    - Emergent state management episode (ES)
- **Taskwork**
  - Work phase
    - Individual activity episode (IA)
    - Group activity episode (GA)

Figure 2. The SE taxonomy of team processes.

A. Transition Phase Episodes

The transition phase includes episodes whose goals are oriented toward post-mortem analysis, forecasting, and planning.

1) Backward Evaluation Episode

A backward evaluation (BE) episode is an I-P-O cycle whose inputs are implicit or explicit artifacts that need to be evaluated. The outcomes are the results of these evaluations. An example of this process could be a review meeting involving all teammates.

Another example of a BE episode would be a documentation review carried out by the team. The quality of the documentation is evaluated based on predetermined criteria and either approvals or recommendations for adjustments are produced. The episode can be performed in a plenary session, where the results are distributed synchronously (face-to-face) to the team members. It can also be performed by a single team member, in which case the results are distributed asynchronously (through e-mail) to the team members.

Failure at this level can occur at each stage of the I-P-O cycle: Poor predetermined criteria (Input) can lead the team to believe that their documentation is better than it actually is. Insufficient time spent on the evaluation (Process) can produce unjustified approval of the document, leading the team to a mistaken belief in its quality. The results (Outcome) can also be discarded without being read, leaving warnings unheeded.

2) Forward Visioning Episode

A forward visioning episode (FV) is an I-P-O cycle dedicated to anticipating future events. It is not concerned with how work will be performed, but with what may happen in the future. It includes risk management evaluations and long term event planning. Previous research has shown that the backward and forward capabilities of the team are closely related [20], but, since these two episodes are typically distinct in SE, they have been separated in our taxonomy.

In SE, forward visioning can be based either on internal team information or on predictions made by experts externally and published on the Web.

For example, forward visioning occurs when the team members brainstorm on the content to include in the context diagram. This diagram is eventually transformed into use cases and sequence diagrams, which will eventually lead to packaging and class diagrams. An ill-defined context diagram can be the root of many problems down the road, and brainstorming sessions can find potential problems and risks early on that have been overlooked in the planning activities.

Again, failures can occur at any stage of the I-P-O cycle. For example, a purpose must be sufficiently defined (Input) for the brainstorming sessions, and these sessions must be moderated to ensure that the purpose is not lost (Process). Results of the sessions (Outcome) must actually be used later in the project.

3) Planning and Prioritization Episode

A planning and prioritization (PP) episode involves preparation of the teamwork and task work elements required to achieve the mission objectives.

"*This involves decision making about how team members will go about achieving their missions, discussion of expectations, relay of task-related information, prioritization, role assignment, and the communication of plans to all team members.*" [2]

PP episodes organize the taskwork and teamwork episodes required to realize the taskwork elements.

These episodes can take place face to face among team members, or they can be executed virtually through Web tools (e-mail, instant messaging, chat, etc.). For example, planning is under way when team members discuss the episodes to perform next in order to complete a task. These episodes distribute the work among team members, and the plan produced constitutes the team process to be applied for the next few episodes.

For a planning episode to be useful, team members must have access to accurate data on the project (Input), and individual team members must be assigned to particular episodes according to their abilities and expertise (Process). Finally, the resulting plan must be followed by the team members (Outcome).

B. Action Phase Episodes

The action phase involves episodes whose goals are oriented toward monitoring, knowledge exchange, and knowledge acquisition.

1) Resource and Progress Monitoring Episode

An RM episode is an I-P-O cycle dedicated to determining and distributing the current state of the task work:

"This involves providing feedback to the team on its goal accomplishment status so that members can determine their progress and their likelihood of success within a given period of time." [2]

These episodes pertain to the diffusion of information that is internal to the team. It can be distributed through face-to-face interaction, but also virtually, through a Web tool like Bugzilla or Basecamp.

For example, RM episodes can be contained within a weekly progress meeting involving all team members. The individuals report their current progress to the team, the work completed, and the problems solved. They can also occur in the daily "stand up" meetings characteristic of the Scrum management approach.

Reporting can fail if the team members do not have an accurate view of their progress because data have not been accurately recorded (Input). Poor results can be produced if the importance or state of advancement of a problem have been inappropriately evaluated (Process). Also, developers can hide data and present a rosier picture of their progress than the reality (Outcome).

2) Team Assistance Episode

In a team assistance (TA) episode, a teammate provides his expertise to a colleague upon request. This can involve face-to-face knowledge transfer, or transfer with the help of Web tools (e-mail, instant messenger, etc.). These episodes pertain to the diffusion of knowledge internal to the team.

For example, a TA episode occurs when a team member assists his colleague during the integration of the core of the system with the Graphical User Interface (GUI). He provides information on how the GUI works based on his experience in developing the component.

Failure of a TA episode can occur if the assisting team member does not have accurate information or expertise in the subject at hand (Input). The communication channel must also be appropriate for the information transfer (Process): for example, it is difficult to transfer requirements by video-conferencing alone or by e-mail alone [21]. As well, the assisted colleague must be listening to the information provided (Outcome).

3) Information Gathering Episode

Information gathering (IG) refers to the acquisition of new information previously unknown to the team, i.e.

external information. It can be gathered in a face-to-face manner with other members of the organization, or virtually from the Web.

In SE, an example of an IG episode would be to read up on a specific concept on the Internet in order to inform the team by means of a wiki, or other knowledge management tool. The information obtained is then distributed to the other team members, through e-mail, for example.

An IG episode can fail if the information researched is inaccurate, out of date, or not found (Input). Poorly documented external libraries can result in many IG episodes having no outcomes at all. If relevant information is found, it must be translated into a format that can be understood by its intended audience (Process): for example, if the information source is a book, the content distributed to the team members must be sufficiently clear to anyone who has not read the book. Finally, the information distributed must be received, read, and understood by its intended audience (Outcome).

C. Interpersonal Phase Episode

The interpersonal phase includes one episode type, whose goal is oriented toward mitigating the effect of emergent states.

1) Emergent State Management Episode

An emergent state management (ES) episode pertains to preemptive and reactive conflict management, emotion management, and motivation management. Its goals are to diminish the effect of ES on the outcomes of episodes. These episodes can take the form of informal pep talks, or be planned motivation boosting events.

For example, consider a conflict related to the proper use of a configuration management tool. One team member launches a BE episode on the configuration environment and finds that some team members are not following the agreed rules on its usage, and numerous issue tickets are being filled out improperly. This creates a conflict between the team members filling out the tickets, who are devoted to development, and the team members reading the tickets in order to track progress, who are devoted to management. In these ES episodes, the rules to be followed for the configuration environment in question must be reiterated, in order to prevent an escalation of the conflict to the point where some team members might refuse to talk to others because they perceive them as unreliable. Reliability is one of the major perceived [12] and observed [13] qualities of the developer.

For such an episode to be successful, the grievance submitted (Input) must be justified. Some tact is required in order to prevent aggravation of the problem, which would manifest as a breakdown in communication between some of the developers (Process). The aggrieved parties must agree to the changes and respect the commitment of the other developers to implement them, in order to promote reliability and restore the communication channels (Outcome).

D. Taskwork Phase Episodes

The original model does not have a taskwork phase, and so does not define work phase episodes either. This is because models in psychology and business management present a clear distinction between taskwork and teamwork [2]. However, in SE teams, the concept of taskwork needs to be taken into account because of its impact on teamwork episodes. Focusing on teamwork alone ignores the complex interactions between teamwork and taskwork. For the taxonomy to be useful in SE research, it must be able to record both types of episodes, so that the ways in which they influence each other can be observed. Taskwork is therefore added as a mode at the same level as the teamwork mode. Taskwork embodies a single phase, the work phase, which describes the approach taken to perform the work.

1) Individual Activity Episodes

An individual activity (IA) episode pertains to the realization of solo work. Its goal is the realization of a specific task, given the information received through previous teamwork mode episodes.

For example, programmers can code, test, and debug specific functionalities in software components. The choice of the functionality to implement has been determined in a previous meeting (a PP episode). The design to follow has been created by other teammates, who previously explained their vision of the system (a TA episode). The programmers read up on the coding language through tutorials available on the Web (an IG episode).

But, when the programmers sit down to actually write the code, they are involved in IA episodes. The intention behind the code written is not to communicate it to the team, but to build a product. Although the code will be distributed to the team, its intention is not communication, but functionality.

The IA can fail if insufficient or inaccurate information is provided (Input). Inadequate expertise and insufficient resources (time, material) can yield poor results (Process). As well, bad unit testing can result in the work being unreliable (Outcome).

Note that an individual working in an office by himself/herself can also be involved in teamwork episodes. Web tools can be used to communicate with remotely located team members. Similarly, collocated teams are not constantly involved in teamwork episodes: A considerable amount of the time spent by collocated teams involves working alone focusing on a task [22].

2) Group Activity Episodes

A group activity (GA) episode pertains to the realization of a task by one or more teammates collaboratively. Its goal is similar to that of the IA episode, which is the realization of a specific task by one person. It differs in the approach used, however, which involves a group session. Typical GA episodes include plenary work sessions, Pair Programming, etc.

GAs, like Pair Programming, are in fact, small IA episodes mixed with immediate TA episodes. Generally, one individual writes down his/her ideas, which are immediately approved or turned down by the others. To follow the theoretical model accurately, we should therefore split GA episodes into the appropriate teamwork and taskwork sub episodes. This is very difficult to achieve in practice, however, as these exchanges can be either very brief or totally implicit. Observing GA episodes as a whole is much more feasible, and can also provide useful information on which tasks are performed in this fashion and why.

GA episodes are subject to the same problems as IA episodes. Similarly, a collocated setting does not imply constant GAs, as much of the work is still performed alone [22]. At the same time, it is possible to perform group work while sitting alone in a cubicle, with the use of whiteboard software, for example [24].

V. An Observed Team Process

To be practical and useful, the episodes in the team software development process need to be measurable. We have conducted many case studies with capstone projects to evaluate their measurability. This section illustrates episode measurement from case study P08. The team was composed of five SE students in their senior year, who were cherry-picked for this study. The client, who provided the high-level requirements, was a collaborating industry partner.

The objective was to realize an Optical Character Recognition (OCR) tool for the extraction of character strings in videos. The resulting software product satisfied the industrial partner, but did not implement all the requirements. Also, some quality problems were identified in a previous article [22], some of which can be linked to communication breakdowns. Overall, the project was considered a partial success.

The project was carried out according to the following process perspectives: The client was from the avionics industry. The client's process perspective concerned the integration of the software product into their software framework and the reuse of some of their proprietary libraries. The organization process perspective was based on the requirements for this capstone project at the engineering school. It required nine hours of collocated team activities per week, the recording of the episodes by each teammate, and the use of various practices based on the Technical

TABLE II. Examples of Episodes Recorded from Project P08.

TN	ID	Date	Duration	Teammates	Input	Output	Episode	Note
5	YD	8 10:00	90	GB, ED, YD, VG, VM	Vision document	Meeting minutes	FV	Meeting with the client. Capture of the first requirements.
30	VM	15 12:00	30	GB, ED, YD, VG, VM	Knowledge	Meeting minutes	TA	Meeting during which ED distributes the information he found on the OCR.
45	GB	16 15:30	30	GB, ED, YD, VG, VM	Knowledge	Meeting minutes	PP	Discussions on the iteration plan.
46	GB	16 16:00	80	GB	Meeting minutes	Iteration plan	IA	Writing of part 4.1 of the iteration plan and creation of the template for iterations.
65	YD	17 17:00	35	GB, ED, YD, VG, VM	Knowledge	Virtual diagram	FV	Discussion on the context diagram.
84	YD	21 09:00	60	ED, YD, VG, VM	SRS	SRS	BE	Review of the SRS.
131	VM	24 16:45	30	VM, OG	Project artifacts	Knowledge	ES	Meeting with OG on the project progress and team dynamics.
156	YD	28 10:00	30	GB, ED, YD, VG, VM	Use cases	Personal notes	BE	Discussion on use cases that must be modified.
171	ED	29 10:00	90	GB, ED, YD, VG, VM, CC	Knowledge	Personal notes	IG	Presentation by the client (CC) of the Replica software.
197	YD	31 17:00	15	GB, ED, YD, VM	Knowledge	Personal notes	TA	ED explains how to report the tasks with the project monitoring tool.
207	GB	33 14:45	75	GB	Knowledge	Prototype	IA	Realization of a simple prototype importing a WPF control in a Windows format.
232	YD	35 11:30	30	GB, ED, YD, VG, VM	Use cases	Sketches	GA	Design of classes in a plenary session.
588	ED	76 13:20	140	ED	Web	Knowledge	IG	Research on a solution to the flickering image problem and on improvement of the code.
596	ED	77 09:10	20	ED, VG, VM	Iteration plan	Personal notes	RM	Review of the tasks performed during the final week.
640	VG	79 14:30	30	GB, ED, YD, VG, VM, PR	Knowledge	Personal notes	RM	Discussion with PR on project progress.
704	GB	85 13:45	15	GB, ED, YD, VG, VM, OG	Emotions	Knowledge	ES	Round table with OG to motivate team members, in spite of all the work left to do.
708	GB	85 14:15	50	GB, YD	Software product	Software product	GA	Work with YD on debugging the WPF code.
909	VM	104 08:30	20	ED, VM	Iteration plan	Personal notes	PP	Planning the day's work.

Solution (TS) process area of CMMI level 3. The project process perspective is based on project management practices recommended by the Unified Process for Education (UPEDU) [23].

The data collection process was validated by the researchers throughout the project, and feedback was provided to the team when errors were made.

The activity tokens obtained provide effort information on completion of an episode. Each token details the participants involved, their role in the work performed, the activity and discipline pertaining to the work, the main input and output artifacts, and a short description of the work performed.

This collection returned a total of 942 validated tokens, each of which represents either an I-P-O teamwork episode or an I-P-O taskwork episode. Tokens with multiple participants usually refer to teamwork episodes. Single participant tokens can either refer to teamwork, such as reading on an external library in order to distribute the information within the team, or to taskwork, such as a creative activity like writing software code or documentation.

A. The Team Process

Samples of the tokens filled out by the teammates during their project are presented in Table II. This table presents 18 tokens, that is, two tokens to illustrate each of the 9 types of episode. The TN column gives the token number extracted from the data base containing the 942 validated tokens. The ID column contains the identification of the teammates who filled out this token. The Date column is a coded field recording the date and hour of the beginning and end of the episode. The Duration column shows the length of each episode in minutes. The Input and Output columns are the main artifacts used for these episodes. The Episode column records the type of episode, as defined by this team process taxonomy. The Note column describes the activity performed during this episode in more detail.

Interviews with the teammates after project completion give us a measure of token reliability, which was evaluated at around 80%. More than 10 case studies were conducted based on this token recording approach. These case studies show that the episodes are measurable and can be used to create a model of team process or to analyze team activities. The most important feature is the taxonomy presented herein, which enables the team process to be measured without bias by the specific project management approach.

B. Discussion

Many problems reported in the literature can be related to taxonomy elements of the team process model. Layman et al.'s recommendations for collocated client and management roles were made in order to improve the Inputs of Transition I-P-O episodes [24]. Damian et al.'s need to use both synchronous and asynchronous means of communication to convey requirement details is related to the Process of Team Assistance I-P-O episodes [21]. Paasivaara's problem solving practice recommendation is mostly related to the Outcome of Team Assistance I-P-O episodes [25]. Karhu et al.'s report of information not reaching the testers is related to the absence of essential Team Assistance I-P-O episodes [26].

Another advantage of the team process taxonomy is to identify interactive and emotive elements. Research has shown that many practices designed for software development teams can help enhance developer motivation. This taxonomy can help identify the origin of this enhancement, through better control of emergent states (ES episodes), for example, or through an increase in team assistance (TA episodes), which results in fewer conflicts.

The identification of team processes can also help inform developers of the communication channels that exist. It can tell them where to find the information they seek, and how to provide information needed by others. It can also help identify the vulnerable episode types in a given context. For example, a language barrier between stakeholders and developers can be counter-productive in obtaining fast and accurate feedback (BE episode) [13].

Some episode types are harder to observe. TA episodes, for example, can be as short as a yes/no question posed between teammates, or as long as an extended coaching session. The token captures the team process episodes that required measurable effort. Informal exchanges outside the context of software development (such as most ES episodes) are not so easily recorded. Also, the tokens do not show brief exchanges between the participants (e-mail, instant messaging, etc.). The team process obtained from the token here is therefore incomplete. Nevertheless, our goal is to demonstrate that team process recording and analysis can provide useful information for research purposes.

The I-P-O episode approach differs from keeping a logbook, as seen in the project management field. Typical logbooks present the information in a sequential fashion. I-P-O episodes, as the name implies, show the importance of recording input, process, and outcome. The details of input, output, and goal description included in the tokens are invaluable in identifying the appropriate I-P-O episode type.

Note that we observe in this team process the alternating of transition and action phases predicted by Marks et al.

VI. CONCLUSIONS

Many process perspectives are related to the development of a software product, and they are intermingled at the team process level. What we propose is a taxonomy to define the various episodes at this level. The benefit of this approach is to enable measurement of team process activities without referring to specific concepts belonging to perspectives other than that of the team process. Software development is essentially a knowledge sharing team activity, where team members accrue the expertise to build a system through reasoning and research. To be effective, this information and knowledge needs to be appropriately shared among team members, a method described by team processes.

But, to observe and report team processes, we must first know what to observe and how to report it. The taxonomy presented here, made up of nine episode types categorized in three teamwork phases and one taskwork phase, provides a common vocabulary that can explain conclusions previously recorded in the literature. Many case studies have shown that these episodes can be reliably self-recorded by teammates. This taxonomic categorization will help future researchers provide a clearer picture of their team-related problems and recommendations.

Future work is planned to analyze audio-video recordings of software team activities based on this taxonomy and build a model of team activities. Work is also in progress to evaluate the impact of the virtual community on team activities by introducing virtual components to episodes. These research projects should validate the exhaustiveness of the taxonomy, and help organize previous works in the field in order to reach a definitive assessment of the impact of team processes on software development projects.

ACKNOWLEDGMENT

This research would not have been possible without the agreement of the company in which it was conducted, and without the participation of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks. We thank Dr. Olivier Gendreau for his supervision of the cases studies and to Professor François Chiocchio from the Department of Psychology at the University of Montreal for his helpful comments. This research was supported in part by NSERC grant A-0141.

REFERENCES

[1] S. W. J. Kozlowski and D. R. Ilgen, "Enhancing the Effectiveness of Work Groups and Teams," Psychological Science, vol. 7, no. 3, pp. 77-124, 2006.

[2] M. A. Marks, J. E. Mathieu, and S. J. Zaccaro, "A temporally based framework and taxonomy of team processes," Academy of Management Review, vol. 26, no. 3, pp. 356–376, 2001.

[3] H. Donker, and M. Blumberg, "Collaborative Process Management and Virtual Teams," Proc. of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 08), IEEE Computer Society, May 2008, pp. 41-43, doi: 10.1145/1370114.1370125.

[4] E. G. Mcguire and A. J. L. Salle, "Reengineering Team Processes in Quality-Focused Software Development Environments," Managing Virtual Enterprises, no. 202, pp. 375-380, 1996.

[5] W. S. Humphrey, "The Team Software Process," Technical Report, Software Engineering Institute, Nov. 2000.

[6] S. J. Motowidlo and J. R. V. Scotter, "Evidence That Task Performance Should Be Distinguished From Contextual Performance," Journal of Applied Psychology, vol. 79, no. 4, pp. 4-9, 1994.

[7] W. C. Borman and S. J. Motowidlo, "Task Performance and Contextual Performance: The Meaning for Personnel Selection Research," Human Performance, vol. 10, no. 2, pp. 99-109, 1997.

[8] S. Basri and R. V. O'Connor, "Towards an Understanding of Software Development Process Knowledge in Very Small Companies," in Systems, Software and Services Process Improvement, Communications in Computer and Information Science, vol. 99, A. Riel, R. O'Connor, S. Tichkiewitch, and R. Messnarz, Éds. Berlin: Berlin Heidelberg, 2010, pp. 153-164.

[9] D. S. Staples and A. F. Cameron, "The Effect of Task Design, Team Characteristics, Organizational Context and Team Processes on the Performance and Attitudes of Virtual Team Members," Proc. of the Annual Hawaii International Conference on System Sciences (HICSS 05), IEEE Computer Society, Jan. 2005, pp. 52-61.

[10] S. Acuna, M. Gomez, and N. Juristo, "How do personality, team processes and task characteristics relate to job satisfaction and software quality?," Information and Software Technology, vol. 51, no. 3, pp. 627-639, Mar. 2009.

[11] C. Cramton and S. Webber, "Relationships among geographic dispersion, team processes, and effectiveness in software development work teams," Journal of Business Research, vol. 58, no. 6, pp. 758-765, Jun. 2005.

[12] K. Siau, X. Tan, and H. Sheng, "Important characteristics of software development team members: an empirical investigation using Repertory Grid," Information Systems Journal, vol. 20, no. 6, pp. 563-580, Nov. 2010.

[13] N. B. Moe and D. Smite, "Understanding a Lack of Trust in Global Software Teams: A Multiple-case," Software Process Improvement and Practice, no. 13, pp. 217-231, Jun. 2008.

[14] Mountain Goat Software, "Introduction to Scrum - An Agile Process," 2011. [Online]. Available: http://www.mountaingoatsoftware.com/topics/scrum [Accessed: 26-Dec-2011].

[15] M. E. Gist, E. A. Locke, and M. S. Taylor, "Organizational behavior: Group structure, process, and effectiveness," Journal of Management, no. 13, pp. 237-257, 1987.

[16] R. A. Guzzo and G. P. Shea, "Group performance and intergroup relations in organizations," in Handbook of industrial and organizational psychology, 2nd ed., M. D. Dunnette and L. M. Hough, Eds. Palo Alto, CA: Consulting Psychologists Press, 1992, pp. 269-313.

[17] J. R. Hackman, "A normative model of work team effectiveness," New Haven, CT, 1983.

[18] 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, pp. 517-43, Jan. 2005.

[19] D. J. Devine, "A review and integration of classification systems relevant to teams in organizations.," Group Dynamics: Theory, Research, and Practice, vol. 6, no. 4, pp. 291-310, 2002.

[20] E. Blickensderfer, J. A. Cannon-Bowers, and E. Salas, "Training teams to self-correct: An empirical investigation," in 12th annual meeting of the Society for Industrial and Organizational Psychology, 1997.

[21] D. Damian, F. Lanubile, and T. Mallardo, "On the Need for Mixed Media in Distributed Requirements Negotiations," IEEE Trans. Softw. Eng., vol. 34, no. 1, pp. 116-132, 2008.

[22] M. Lavallee and P. N. Robillard, "Causes of premature aging during software development: An observational study," Proc. of the 12th International Workshop on Principles on Software Evolution (IWPSE-EVOL 11), Association for Computing Machinery, pp. 61-70, Sep. 2011.

[23] P. N. Robillard, P. Kruchten, and P. D'Astous, "Unified Process for Education (UPEDU)," École Polytechnique de Montréal, 2011. [Online]. Available: www.upedu.org.

[24] L. Layman, L. Williams, D. Damian, and H. Bures, "Essential communication practices for Extreme Programming in a global software development team," Information and Software Technology, vol. 48, no. 9, pp. 781-794, Sep. 2006.

[25] M. Paasivaara and C. Lassenius, "Collaboration practices in global inter-organizational software development projects," Software Process: Improvement and Practice, vol. 8, no. 4, pp. 183-199, Oct. 2003.

[26] K. Karhu, O. Taipale, and K. Smolander, "Investigating the relationship between schedules and knowledge transfer in software testing," Information and Software Technology, vol. 51, no. 3, pp. 663-677, Mar. 2009.