

Titre: Mesure et analyse des activités collaboratives AD HOC retrouvées
Title: au cours d'un processus de génie logiciel

Auteurs: Sébastien Cherry, & Pierre N. Robillard
Authors:

Date: 2004

Type: Rapport / Report

Référence: Cherry, S., & Robillard, P. N. (2004). Mesure et analyse des activités collaboratives AD HOC retrouvées au cours d'un processus de génie logiciel. (Rapport technique n° EPM-RT-2004-03). <https://publications.polymtl.ca/2615/>

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2615/>
PolyPublie URL:

Version: Version officielle de l'éditeur / Published version

Conditions d'utilisation: Tous droits réservés
Terms of Use:

Document publié chez l'éditeur officiel

Document issued by the official publisher

Institution: École Polytechnique de Montréal

Numéro de rapport: EPM-RT-2004-03
Report number:

URL officiel:
Official URL:

Mention légale:
Legal notice:



EPM-RT-2004-03

**MESURE ET ANALYSE DES ACTIVITÉS
COLLABORATIVES *AD HOC* RETROUVÉES AU COURS
D'UN PROCESSUS DE GÉNIE LOGICIEL**

Sébastien Cherry, Pierre N. Robillard
Département de Génie informatique
Laboratoire de recherche en génie logiciel
École Polytechnique de Montréal

Mars 2004

Poly

EPM-RT-2004-03

Mesure et analyse des activités collaboratives *ad hoc*
retrouvées au cours d'un processus de génie logiciel

Sébastien Cherry, Étudiant M.Sc.A.
Pierre N. Robillard, Ph.D., ing. Directeur de recherche
Département de génie informatique
École Polytechnique de Montréal
Laboratoire de recherche en génie logiciel

Mars 2004

©2004

Sébastien Cherry, Pierre N. Robillard
Tous droits réservés

Dépôt légal :

Bibliothèque nationale du Québec, 2003
Bibliothèque nationale du Canada, 2003

EPM-RT-2004-03

Mesure et analyse des activités collaboratives ad hoc retrouvées au cours d'un processus de génie logiciel

par : Sébastien Cherry, Pierre N. Robillard
Département de génie informatique
École Polytechnique de Montréal

Toute reproduction de ce document à des fins d'étude personnelle ou de recherche est autorisée à la condition que la citation ci-dessus y soit mentionnée.

Tout autre usage doit faire l'objet d'une autorisation écrite des auteurs. Les demandes peuvent être adressées directement aux auteurs (consulter le bottin sur le site <http://www.polymtl.ca/>) ou par l'entremise de la Bibliothèque :

École Polytechnique de Montréal
Bibliothèque – Service de fourniture de documents
Case postale 6079, Succursale «Centre-Ville»
Montréal (Québec)
Canada H3C 3A7

Téléphone : (514) 340-4846
Télécopie : (514) 340-4026
Courrier électronique : biblio.sfd@courriel.polymtl.ca

Ce rapport technique peut-être repéré par auteur et par titre dans le catalogue de la Bibliothèque : <http://www.polymtl.ca/biblio/catalogue.htm>

SOMMAIRE EXÉCUTIF

Les processus de génie logiciel sont de nos jours de plus en plus objets de préoccupations dans plusieurs organisations qui développent du logiciel. Le but du processus étant d'accroître la prédictibilité des échéanciers, des coûts ainsi que de la qualité du logiciel développé, les entreprises investissent désormais une plus grande quantité d'argent et d'effort afin d'améliorer continuellement leur processus de développement de logiciels.

Par ailleurs, beaucoup de chercheurs s'entendent pour dire que la réussite de projets informatiques ne dépend pas seulement de facteurs techniques, mais que certains aspects humains auraient un impact tout aussi significatif, sinon plus. Parmi ces derniers, le travail collaboratif au sein d'une équipe de développement de logiciels a été identifié par plusieurs comme ayant un impact majeur sur les chances de succès d'un projet logiciel permettant à une équipe de partager toute l'information nécessaire en vu de la *cristalliser* en un logiciel de qualité qui correspondra aux attentes de l'usager tout en respectant les contraintes de temps et d'argent. De plus, même si une organisation possède un processus de génie logiciel formel et bien définit, des recherches empiriques ont démontré l'importance indéniable que revêtent les activités collaboratives ad hoc au cours du développement d'un logiciel, importance qui se fait également gravement ressentir dans un contexte de développement distribué. Toutefois, ces dernières recherches ont omis de décrire la teneur de ces activités collaboratives informelles et spontanées laissant de ce fait un vaste champ de recherche en génie logiciel toujours inexploré.

Ce document présente une recherche empirique, entreprise pour l'obtention d'une maîtrise en génie informatique, qui a présentement cours dans le cadre d'une étude de cas en profondeur dans l'industrie et qui explore l'aspect du travail collaboratif en génie logiciel, plus spécifiquement les activités collaboratives ad hoc retrouvées lors d'un processus de développement de logiciels.

Cette recherche visera donc dans un premier temps à modéliser et caractériser les activités collaboratives ad hoc retrouvées au cours d'un processus de génie logiciel pour en connaître l'importance, le contenu, de même qu'en distinguer les activités cognitives sous-jacentes impliquées. Cela constituera dans un deuxième temps de préambule afin de proposer des améliorations au processus observé dans le cadre de cette étude.

Pour ce faire, la stratégie de recherche proposée dans ce document s'appuie sur une collecte de données provenant de diverses sources d'information disponibles rendant ainsi possible une triangulation de celles-ci pour en augmenter la validité. Finalement, une seconde phase à cette recherche pourrait être menée afin de confirmer les hypothèses

issues des théories induites des données empiriques recueillies ou encore, afin d'implanter au sein d'une équipe pilote de nouvelles pratiques au processus observé et visant à y accroître l'efficacité sur le plan du travail collaboratif. Une seconde mesure pourrait enfin avoir lieu afin de ressentir l'impact des nouvelles pratiques et si les résultats s'avèrent fructueux, elles pourraient par la suite être appliquées à plus grande échelle.

Plus de détails sur nos motivations seront trouvés dans ce document suivis d'une discussion sur la méthodologie de recherche employée, incluant nos méthodes de collecte de données et plaçant également une certaine emphase sur les techniques d'analyse utilisées pour déchiffrer l'importante quantité de données amassées. Finalement, quelques résultats d'une analyse préliminaire seront révélés, résultats qui confirment tant la pertinence des données que l'importance du phénomène observé.

TABLE DES MATIÈRES

1. INTRODUCTION.....	1
2. MOTIVATIONS	3
2.1 POURQUOI L'AMÉLIORATION CONTINUE DE PROCESSUS?	3
2.2 POURQUOI LE TRAVAIL COLLABORATIF AD HOC?	4
2.3 POURQUOI UNE RECHERCHE EMPIRIQUE?	6
3. PROTOCOLE DE RECHERCHE	7
3.1 ÉNONCÉ DU PROBLÈME	7
3.2 STRATÉGIE DE RECHERCHE	8
4. RÉSULTATS INITIAUX OBTENUS	15
5. CONCLUSIONS.....	21
6. REMERCIEMENTS	22
7. RÉFÉRENCES.....	23

1. INTRODUCTION

Le génie logiciel, plus jeune discipline du génie, est apparu il y a quelques dizaines d'années en réaction à la crise du logiciel, expression désignée pour la première fois en 1968 [16] et désormais communément utilisée dans la littérature pour faire état du nombre très élevé de ratées dans la conduite de projets logiciels.

Depuis, la discipline a grandement évolué par l'introduction de processus de gestion de projets, de diverses pratiques et techniques en développement de logiciels, ainsi que de diverses métriques et mesures pour tenter d'assurer un meilleur contrôle de qualité des logiciels développés. C'est dans ce même courant qu'on vit apparaître une panoplie de processus de génie logiciel ou de méthodologies, suggérant plutôt de concentrer des efforts non pas uniquement sur le produit à concevoir et développer, mais plutôt sur le processus qui produira le produit logiciel. Ainsi, on dit que les processus de génie logiciel augmentent les chances de succès de projets logiciels en améliorant la prédictibilité des échéanciers et des coûts en plus d'assurer la production d'un logiciel de grande qualité [21].

Toutefois, encore aujourd'hui, malgré les énormes progrès réalisés et les innombrables processus et méthodologies qui existent, certains problèmes persistent toujours. Plusieurs auteurs en arrivent à ce constat tels que Brooks (1987) dans son article intitulé « No Silver Bullet » [3], Gibbs (1994) dans « Software's Chronic Crisis » [8] et Pressman (1996) en parlant de « software's chronic affliction » [17]. Les entreprises productrices de logiciels pour usage interne ou pour fins commerciales sont de plus en plus conscientes de l'utilité des processus de génie logiciel, certaines mettant même sur pied des départements consacrés uniquement à l'instauration et à l'amélioration continue de leur processus de développement de logiciels. Chacun y allant donc de théories et méthodes différentes afin d'implanter et de faire de l'amélioration continue de processus, c'est dans cet élan que s'inscrit la recherche proposée dans le présent document.

Par ailleurs, il est de plus en plus supporté dans la littérature que multiples problèmes rencontrés au cours de processus de développement de logiciels ne sont pas nécessairement imputables à un manque de technologie, mais peuvent être également attribuables à des facteurs humains du génie logiciel [4 ; 5 ; 9 ; 10 : 12 ; 18 ; 20 ; 21 ; 22]. Alors que certains aspects tels que la « communication » [12 : 22], la « coordination » [9 ; 10], ainsi que la « collaboration » [4 ; 5 ; 20], gagnent en intérêt parmi la communauté de chercheurs, de nouveaux défis méthodologiques émergent. Effectivement, les facteurs humains ont été longuement outrepassés auparavant pour plusieurs raisons, mais principalement pour la difficulté à mesurer quantitativement ces facettes [18]. Néanmoins,

les études empiriques en génie logiciel gagnent en popularité et commencent à se raffiner de plus en plus pour étudier ce nouveau sujet d'intérêt, l'homme, empruntant certaines méthodes et techniques jadis propres aux sciences humaines telles que la psychologie et la sociologie. Comme plusieurs chercheurs, nous croyons que ce domaine offrira plusieurs opportunités de recherches pour encore de nombreuses années.

Ce document présente une recherche empirique qui a présentement cours dans le cadre d'une étude de cas en profondeur dans l'industrie et qui explore l'aspect du travail collaboratif en génie logiciel, mais plus spécifiquement les activités collaboratives *ad hoc* retrouvées lors d'un processus de développement de logiciels. Par activités collaboratives *ad hoc*, nous entendons les activités qui ne sont pas prescrites formellement, qui ont lieu entre deux ou plusieurs concepteurs travaillant ensemble sur une tâche particulière du projet et qui surviennent de façon informelle et spontanée. Elles peuvent prendre diverses formes telles qu'en conversations de pair à pair, en échanges par courrier électronique, en appels téléphoniques, etc...

Aucune recherche connue à ce jour n'a tenté de décrire le contenu de ce type d'activités et nous croyons qu'une meilleure compréhension de celles-ci, de leur contenu et des mécanismes cognitifs sous-jacents nous aidera par la suite à proposer certaines améliorations aux processus de développement de logiciels, améliorations qui seront plus adaptées à la réalité humaine et à la réalité empirique du développement de logiciels. Ce but prend tout son sens puisque certaines études antérieures ont démontré l'importance indéniable de ces activités [18 ; 20]. Aussi, une autre raison qui nous porte à réaliser cette recherche se situe dans le domaine du développement de logiciels collaboratif, également connu comme développement de logiciels distribué, où il a été découvert que la distance dans un tel contexte lève des barrières aux communications informelles, ce qui résulte en divers problèmes de coordination [10]. Cela est vraiment d'actualité, puisque pour un certain nombre de raisons d'affaires, le développement de logiciels distribué est de plus en plus pratiqué ces jours-ci dans l'industrie [11].

Plus de détails sur nos motivations seront trouvés dans la prochaine section de ce document suivis d'une discussion sur la méthodologie de recherche employée, incluant nos méthodes de collecte de données et plaçant également une certaine emphase sur les techniques d'analyse utilisées pour déchiffrer l'importante quantité de données amassées. Finalement, quelques résultats d'une analyse préliminaire seront révélés, résultats qui confirment tant la pertinence des données collectées que l'importance du phénomène observé.

2. MOTIVATIONS

Conscient que le sujet d'étude de cette recherche et les méthodes employées pour l'investiguer ne sont pas très orthodoxes dans le domaine du génie logiciel, les raisons qui motivent ces choix sont néanmoins nombreuses et pertinentes. Quelques-unes d'entre-elles vous seront donc présentées au cours de cette section.

« To improve the rigor of measurement in software engineering, we need not restrict the type or range of measurements we can make. Indeed, measuring the unmeasurable should improve our understanding of particular entities and attributes, making software engineering as powerful as other engineering disciplines. Even when it is not clear how we might measure an attribute, the act of proposing such measures will open a debate that leads to greater understanding. » [6]

2.1 POURQUOI L'AMÉLIORATION CONTINUE DE PROCESSUS?

Comme nous l'avons mentionné plus haut, les processus de génie logiciel augmentent les chances de succès de projets logiciels en améliorant la prédictibilité des échéanciers et des coûts en plus d'assurer la production d'un logiciel de plus grande qualité [21]. Les entreprises productrices de logiciels sont donc de plus en plus conscientes que l'amélioration du produit logiciel passe inévitablement par l'amélioration du processus qui produira ledit logiciel.

Ainsi, dans leur article intitulé *ANSI An Inductive Method for Software Process Improvement : Concrete Steps and Guidelines*, Briand, El Emam et Melo (1995) [2] exposent une approche différente pour faire de l'amélioration de processus. Ceux-ci proposent qu'au lieu de faire du *process improvement* par une approche descendante, c'est-à-dire en imposant l'instauration de processus ou méthodologies « génériques » ou commerciales, il est plus efficace d'adopter une approche ascendante en partant de la compréhension du processus déjà en place, en déterminant les causes des problèmes rencontrés et en adoptant des mesures pour y palier. Le problème majeur rencontré avec les processus génériques ou commerciaux, poursuivent les auteurs, est qu'ils sont basés sur de nombreuses assomptions sur ce qui constitue de soit disant « bonnes pratiques », qui ne sont pas vérifiées empiriquement et d'ailleurs pas toujours applicables dans certains contextes. De plus, ils soulignent que beaucoup de problèmes rencontrés en pratique ne sont pas adressés par ces processus. L'approche ascendante de l'amélioration du processus consiste donc, selon ces mêmes auteurs, à planter une équipe d'amélioration du processus, modéliser le processus existant dans l'entreprise concernée, mener une analyse

qualitative afin d'identifier les sources de problèmes, définir et documenter un plan d'action ainsi qu'un programme de mesures, mener un projet pilote pour finalement implanter le processus à l'ensemble de l'organisation.

2.2 POURQUOI LE TRAVAIL COLLABORATIF *AD HOC*?

Comme mentionné auparavant, un nombre croissant de chercheurs soutiennent que de nombreux problèmes survenant au cours du développement de logiciels peuvent être imputables aux facteurs humains. Entre autres, Perry, Staudenmayer et Votta (1994) [18] croient que trop d'attentions sont portées sur des aspects technologiques. L'une des raisons fréquemment citées disent-ils, est la difficulté de mesurer quantitativement ces facteurs humains. Ces mêmes propos sont par ailleurs soutenu par Seaman (1999) [23].

Ainsi, pour étudier l'aspect humain du génie logiciel, plusieurs approches ont été envisagées. Certains chercheurs se sont penchés sur la communication au cours du développement de logiciels [12 : 22], alors que d'autres ont plutôt étudié l'aspect de la coordination [9 : 10] et finalement, certains autres quant à eux se sont intéressés au travail collaboratif en génie logiciel [4 ; 5 ; 20]. C'est d'ailleurs ce dernier sujet qui nous intéresse dans le cadre de la recherche présentée dans ce document.

Par rapport à l'aspect du travail collaboratif, Robillard et Robillard (2000) [20], ont identifié empiriquement quatre types d'activités collaboratives au cours d'un processus de génie logiciel. Ils y définissent les activités collaboratives obligatoires (mandatory) comme étant des réunions formelles planifiées sur une base régulière et les activités appelées (called) comme étant des réunions planifiées par des membres d'une équipe pour résoudre un problème ponctuel et souvent technique. Les activités collaboratives *ad hoc* sont quant à elles définies comme étant le travail que les membres d'une équipe effectuent ensemble au même moment incluant le partage d'information et de connaissances sur le travail qu'ils font et finalement, les activités individuelles sont celles où un individu travaille seul sur une tâche du projet. Dans l'étude de cas réalisée par Robillard et Robillard (2000), il a été recensé que 4% du temps était occupé sous forme d'activités obligatoires, 14% du temps sous forme d'activités appelées et les 82% restant étaient répartis équitablement entre les activités collaboratives *ad hoc* et les activités individuelles. Une conclusion apportée par cette recherche est que les activités collaboratives *ad hoc* peuvent jouer un rôle majeur dans la dynamique des communications d'équipes. En effet, comme mentionné précédemment, ce type d'activité occupe une portion importante du temps total d'un projet, soit de l'ordre de 41% lors de l'étude de cas. De plus, ces activités semblent avoir un impact important sur les

activités individuelles puisqu'elles sont souvent réalisées par deux équipiers et précédent souvent de longues sessions de travail individuel.

Par ailleurs, Perry, Staudenmayer et Votta (1994) [18] trouvent au cours d'une autre étude de cas que les communications informelles occupent en moyenne approximativement 75 minutes par jour du temps des développeurs logiciels.

Seaman (1996) [22] soutient également que ce type de communication est un élément non négligeable entrant en ligne de compte au cours d'un processus de développement de logiciels et qu'il est essentiel pour que les développeurs puissent accomplir leurs tâches adéquatement.

Monopolisant donc une portion du temps relativement considérable d'un projet logiciel et constituant un élément important tel que soutenu par les auteurs susmentionnés, une recherche exploratoire plus approfondie est conséquemment essentielle afin de connaître le contenu de ces activités collaboratives *ad hoc*, les communications qui en découlent et d'en mesurer l'impact autant positif que négatif sur le reste du processus de développement.

D'autre part, le développement de logiciels collaboratif, traduit de l'anglais « collaborative software development » ou aussi connu sous l'expression « développement de logiciels distribué », est un domaine de recherche de plus en plus à la vogue ces jours-ci. Ces deux expressions réfèrent au développement de logiciels distribué dans le temps et sur de grandes distances, chose que l'on retrouve de plus en plus répandue de nos jours pour de nombreuses raisons d'affaires appuyées par les adeptes de cette pratique.

Toutefois, des recherches récentes dans ce domaine [10] rapportent que la distance entre les membres des équipes virtuelles élève certaines barrières aux communications informelles ce qui résulte en divers problèmes de coordination. Rappelons que la coordination ou le « task ordering » est définie par Klein (2001) comme étant la tentative par plusieurs entités d'agir en concert pour en atteindre un but commun par l'exécution d'un plan que tous comprennent.

Cela constitue donc une autre raison importante pour laquelle une recherche dans ce champ doit être entreprise. À l'issue des résultats obtenus par la présente étude, différentes approches pourront être envisagées. D'une part, en tentant de créer ce que certains appellent le « virtual 30 meters » [12] en facilitant par divers moyens de communications les échanges informels et spontanés. D'une autre part, en promulguant des pratiques plus formelles d'échanges d'informations et de connaissances ou telles que celles qui visent par

exemple, à modulariser le produit, le processus ou l'expertise en parallèle avec l'organisation diminuant ainsi les dépendances entre les groupes distants [9].

2.3 POURQUOI UNE RECHERCHE EMPIRIQUE?

La recherche empirique qui prend son fondement de la méthode expérimentale est employée depuis belle lurette dans les divers domaines reliés aux sciences humaines tels que la psychologie et la sociologie. Elle est d'ailleurs très souvent considérée comme étant la seule méthode scientifique valable et acceptée dans ces domaines. Quant au génie logiciel, la recherche empirique y est utilisée à beaucoup plus petite échelle, mais tout de même, y est depuis bon nombre d'années. Ce n'est que tout récemment que la recherche empirique a vu sa popularité croître en raison, entre autres, de l'intérêt également croissant des aspects humains du génie logiciel [23].

Par ailleurs, à ceux qui contesteraient encore cette nouvelle tendance, Tichy (1998) [24] énonce de nombreux arguments en sa faveur par le biais de son article intitulé « Should Computer Scientists Experiment More? ». Toutefois, les tenants de cette pratique en génie logiciel croient que maintenant que la quantité de recherches empiriques augmente, la qualité de ce type de recherches doit augmenter à son tour [15 ; 25] et conséquemment, le protocole de recherche suggéré et décrit dans le présent document fut largement influencé par les recommandations tirées des nombreuses publications à ce sujet.

3. PROTOCOLE DE RECHERCHE

3.1 ÉNONCÉ DU PROBLÈME

3.1.1 Objectif général de la recherche. Tel que discuté auparavant, l'importance et la nécessité du travail collaboratif *ad hoc* et des communications qui en découlent au sein du processus de développement sont largement supportées par plusieurs auteurs [4 ; 9 ; 10 ; 11 ; 12 ; 18 ; 20 ; 22]. Bien que certains chercheurs aient quantifié l'importance du phénomène, aucune recherche connue à ce jour n'a tenté d'en déterminer la teneur, d'en décrire le contenu. Il est par ailleurs intéressant de constater que malgré les nombreux processus formels prescrits que l'on peut retrouver de nos jours, qui sont parfois inapplicables dans certains contextes, basés sur de nombreuses assumptions ou encore qui ne répondent pas à l'ensemble des problèmes qui peuvent survenir lors du développement d'un logiciel [2] et malgré les nombreuses formes d'activités collaboratives formelles que ces processus prescrivent, les activités collaboratives *ad hoc* occupent une portion relativement importante du temps alloué par les concepteurs à un projet logiciel [10; 18; 20; 22]. Par conséquent, ces considérations nous amènent donc à définir les principaux objectifs de recherche suivants :

- Observer le travail collaboratif dans le cadre d'une étude de cas en profondeur dans l'industrie pour essayer d'en dresser un modèle conceptuel théorique et d'en distinguer des patrons;
- Caractériser les activités collaboratives *ad hoc* recensées, les communications qui en découlent, en identifier décrire le contenu;
- Générer une série d'hypothèses issues des résultats de cette recherche qui pourront être par la suite validées par diverses recherches confirmatoires.

Cette recherche se voudra donc exploratoire dans un premier temps, étant donné que le domaine à l'étude est toujours inexploré ou très peu et afin d'induire, à partir des données collectées, la représentation théorique du travail collaboratif *ad hoc* retrouvé au cours d'un processus de génie logiciel donné. Dans cette même optique, il sera très important de garder un esprit d'ouverture quant à l'issue de cette recherche et à toute(s) découverte(s) pouvant y découler. Cette recherche se voudra par la suite plus descriptive pour dresser un portrait plus précis de ces échanges informels à l'aide du modèle conceptuel formé des données empiriques recueillies. Finalement cette recherche pourrait potentiellement devenir plus explicative en cherchant les raisons issues, entre autres, des sciences cognitives et qui sont sous-jacentes aux résultats obtenus.

3.1.2 Pertinence théorique de la recherche. L'absence ou sinon le peu de recherches empiriques sur les activités collaboratives *ad hoc* font que cette recherche est tout à fait pertinente en soit d'un côté théorique. Elle permettra d'établir un modèle de ces dernières, de mieux saisir les aspects cognitifs impliqués, à partir desquels nous pourrons générer une série d'hypothèses créant ainsi une base théorique dans ce domaine toujours inexploré du génie logiciel.

3.1.3 Pertinence pratique de la recherche. Partant du fait qu'une bonne collaboration est une condition *sine qua non* pour que le travail d'une équipe de développement de logiciels se fasse efficacement afin d'obtenir un produit de qualité, qui correspond réellement aux besoins de l'usager, dans les temps et les coûts prévus, cette recherche est pertinente d'un côté pratique puisqu'elle permettra potentiellement de proposer des améliorations aux processus de génie logiciel.

Tel que souligné par Robillard (1996) [19];

« Il est possible qu'une connaissance des processus mentaux propres à une activité spécifique puisse aider à concevoir des règles de l'art qui soient appropriées et efficaces pour cette activité. Il n'existe peut-être pas de démarche universelle applicable à toutes les situations mais quelques approches adaptées seraient déjà un gain significatif. »

3.2 STRATÉGIE DE RECHERCHE

Cette section tentera de décrire la démarche de recherche privilégiée afin de répondre à chacun des objectifs de recherche susmentionnés.

3.2.1 Approche générale. Le travail de recherche entrepris sera accompli par le biais d'une observation participante réalisée dans le cadre d'une étude de cas en profondeur en milieu industriel. Plusieurs raisons motivent ce choix :

- Observation des évènements et du phénomène à l'étude en temps réel;
- Données recueillies dans leur contexte [23] ;
- L'observation participante permet d'étudier les comportements et relations interpersonnelles des participants dans leur environnement naturel ;
- L'observation participante est appropriée pour les recherches exploratoires et descriptives [1, 13] ;

- Les échanges à observer ne sont pas visibles du point de vue de quelqu'un hors du terrain d'observation, du public [23] ;
- Le phénomène à observer n'est observable que dans la vie de tous les jours et difficilement reproductible dans un contexte expérimental ;
- Le phénomène est assez limité en étendu pour être observé sous forme d'étude de cas.

De plus, ce type d'approche est tout à fait approprié dans notre cas puisque cette recherche est exploratoire, comme nous l'avons mentionné précédemment. De plus, comme le souligne Jorgensen (1989) [13] et Babbie (2001) [1], l'étude sur le terrain et l'observation participante sont appropriées lorsqu'il ne s'agit pas de vérifier empiriquement des hypothèses formulées d'avances, mais plutôt pour générer inductivement des théories à partir d'observations et de données empiriques recueillies. De cette façon, des observations initiales peuvent mener à certaines conclusions, lesquelles peuvent guider certaines observations subséquentes, vérifiant les conclusions précédemment faites, impliquant peut-être une révision de ces dernières et ainsi de suite.

3.2.2 Population cible. La population cible est constituée de toutes équipes de développement de logiciels d'au moins deux personnes et plus, dans toutes entreprises produisant du logiciel pour usage interne ou pour fins commerciales.

3.2.3 Échantillon. L'échantillon ou l'équipe de développement de logiciels choisie oeuvre dans une grande entreprise* productrice de logiciels pour fins commerciales, entreprise ayant pignon sur rue depuis de nombreuses années, dans laquelle il existe un processus de développement clairement défini et que l'on pourrait qualifier de mature. Quoique la recherche se déroule au sein d'une grande entreprise, on peut dire que celle-ci comporte également certains attributs de petites ou moyennes organisations puisque le travail y est subdivisé en équipe de petite taille.

Ce groupe de concepteurs a donc été sélectionné tout simplement sur la base de l'opportunité de recherche qui s'est offerte au sein de celui-ci, opportunité assez rarissime en milieu organisationnel pour réaliser ce type de recherche nous forçant à saisir l'occasion sans pouvoir faire de choix sur une base plus scientifique ou statistique. Nonobstant cet échantillonnage qui s'est imposé quelque peu de lui-même, on peut dire d'un jugement fondé sur le sens commun (face validity) [1], que l'équipe choisie constituée de 8 personnes

* En vertu d'une entente de confidentialité conclue avec l'entreprise participante, il nous est interdit d'identifier cette dernière.

peut être tout à fait représentative de la plupart des équipes de développement que l'on peut retrouver dans l'industrie avec une échelle très variable d'âges, de degrés de scolarisation, d'années d'expérience en développement de logiciels ainsi que d'ancienneté dans la compagnie même.

Toutefois, puisque cette recherche constitue une étude de cas et que par définition à partir d'une étude de cas il n'est pas possible d'en généraliser les résultats à l'ensemble de la population, le choix de cette équipe n'implique aucun biais notable.

3.2.4 Méthode de collecte de données. La méthode de collecte qui suit fut élaborée à la suite d'une période préalable d'ethnographie au sein du groupe choisi et qui dura plusieurs mois.

Une première phase de cette collecte qui eu lieu à l'automne 2003 est maintenant terminée et s'est échelonnée sur une durée de 8 semaines. C'est d'ailleurs à partir de ces premières données que furent produits les résultats présentés plus loin. Le but de cette collecte était de recueillir le maximum d'informations du début jusqu'à la fin du développement d'une mise à jour (patch) d'une version donnée du logiciel produit. L'observation se faisait donc lors d'un cycle de maintenance d'un logiciel, plutôt qu'en développement d'une nouvelle version du logiciel.

Les données recueillies lors de cette première phase incluent donc :

- Plus de 185 heures d'enregistrements audio-videos de sessions de travail réalisées par les membres de l'équipe et s'échelonnant sur 37 jours de travail ;
- La capture automatique d'un total de 2496 courriels reçus et envoyés par les 8 membres de l'équipe ;
- Une sauvegarde journalière du code source ;
- Une sauvegarde journalière de tous les documents et artefacts internes se trouvant sur le serveur de fichiers de l'équipe.

Nous avons opté pour l'enregistrement audio-videos versus la prise de notes sur le terrain parce que les enregistrements nous donnaient l'énorme avantage de pouvoir être visionnés à répétition. Cet avantage est très précieux dans une situation comme la nôtre où ce que nous cherchons n'est pas défini d'avance étant donné le caractère exploratoire de notre recherche. Bien entendu, lors de l'enregistrement, des précautions particulières ont été prises afin que la caméra et les microphones soient les moins visibles que possible. Aussi, les 37 jours d'enregistrements vidéo comprennent 15 jours de films enregistrés avant le début du développement de la « patch ». Cela a donc permis aux développeurs de

s'habituer à la présence de la caméra et des microphones diminuant ainsi les biais qui peuvent y être associés. À cet effet, certaines personnes ont même avoué qu'elles oublaient le fait qu'elles étaient filmées, ce qui ne se serait pas produit disaient-elles si un observateur avait été constamment à leur côté pour écouter leurs conversations.

Comme mentionné plutôt, la capture des courriels se faisait de façon automatique et ce, à l'aide de déclencheurs définis dans le logiciel de messagerie utilisé dans cette compagnie. Cette capture incluait les courriels reçus et envoyés par chaque membre de l'équipe. Ainsi, cela permet de faire une validation croisée des courriels capturés, puisqu'un courriel qu'un émetteur a envoyé doit nécessairement avoir été reçu par son récepteur et ce qu'un récepteur a reçu doit forcément avoir été envoyé par son émetteur.

La sauvegarde journalière du code source et des divers documents et artefacts internes pourront quant à eux servir potentiellement à une analyse de contenu subséquente.

Il est à noter que beaucoup d'informations pour les besoins ont été récoltées lors de cette phase de collecte. Toutefois, comme mentionné auparavant, le but était d'amasser le maximum de données dans l'intervalle de collecte choisie et de faire un échantillonnage sur les données collectées plutôt que d'établir le type de données à ramasser, de ne recueillir que le strict minimum en risquant de s'apercevoir lors de l'analyse que tel ou tel élément manque pour faire telle ou telle analyse. Cette dernière approche est la façon de procéder typique dans le cas de recherche confirmatoire où l'on opérationnalise des concepts connus à l'avance. Cependant, dans le cas de recherche exploratoire comme la nôtre, nous jugeons plus opportun de rester prudent et de nous laisser toute la latitude de faire ce choix au moment de l'analyse et la flexibilité de changer de variables ou de modèle de données si plusieurs itérations d'analyse s'avèreraient nécessaires. Ces itérations sont souvent un mal nécessaire dans le cas de recherches exploratoires puisque comme nous l'avons déjà mentionné, ce que l'on recherche n'est pas clairement connu d'avance.

3.2.5 Analyse des données. La section qui suit décrira certaines méthodes qui seront mises de l'avant pour l'analyse des données, car celle-ci sera sans aucun doute la partie la plus longue et cruciale de la recherche. En effet, elle consistera grossièrement à appliquer différentes techniques d'analyse sur la vaste quantité de données récoltées sur le terrain afin d'obtenir un modèle conceptuel qui illustre le plus fidèlement possible la réalité empirique de ces données.

Pour ce faire, la technique d'analyse préconisée est l'analyse exploratoire et séquentielle des données ou mieux connue sous le nom anglais de *Exploratory Sequential*

Data Analysis (ESDA) [7]. Cette technique est appropriée pour les recherches exploratoires dont le but est de trouver des réponses à des questions de recherches empiriques ou encore de trouver des patrons parmi les données recueillies et de les décrire en utilisant, par exemple, de simples représentations statistiques. Ensuite, l'ESDA permet aux chercheurs d'énoncer, à partir de ces descriptions, certaines hypothèses qui sont par la suite vérifiées par le biais de recherches confirmatoires utilisant des méthodes d'inférences statistiques. Toutefois, la particularité de l'ESDA est qu'elle s'applique plus spécialement dans le cas de recherches où l'intégrité séquentielle des données doit être conservée. Ainsi, l'analyse de conversations, d'interactions, l'analyse du verbal et du non verbal, des processus cognitifs ou de discours sont tous des exemples de domaines de recherches où l'information séquentielle des données est essentielle pour être menés à bien.

L'utilisation de la méthodologie de l'ESDA repose sur l'établissement et la définition de concepts formels issus de trois traditions ou influences et prenant sens également parmi les données qui seront collectées ultérieurement [7]. Sommairement, la tradition bémoriste s'intéresse principalement au comportement humain dans son environnement naturel, la tradition cognitiviste s'intéresse essentiellement aux performances cognitives humaines dans l'accomplissement de certaines tâches alors que la tradition sociale se penche plutôt sur l'interaction et la communication de l'individu avec son milieu soit entre êtres humains ainsi qu'entre humains et ordinateurs [7]. Dans le cadre de cette recherche, les deux points de vues sous lesquels nous nous plaçons sont les traditions bémoriste et cognitiviste.

Afin de faire émerger un patron, une structure ou un modèle des données empiriques recueillies, l'ESDA propose huit sortes d'opérations ou de manipulations de données (Fisher et Sanderson 1996) [7]. La première appelée *chunks* ou segmentation, consiste à grouper les données adjacentes qui démontrent ensemble une certaine cohérence. Ainsi, les données se retrouvent toutes regroupées en séquences qui sont ensuite contenues dans de séquences plus larges et ainsi de suite, afin former un ordre hiérarchique dans les données. La deuxième, *comments* ou commenter, suggère simplement d'ajouter aux éléments de données des notes informelles ou formelles pour ajouter des précisions additionnelles ou certains éléments d'information pertinents supplémentaires aux données manipulées. Ensuite, *codes* ou codage, est sans doute la manipulation la plus importante, propose de d'étiqueter chacun des éléments de données à l'aide d'un code formé d'une syntaxe particulière et contenu dans une liste exhaustive, exclusive et relativement restreinte de catégories et ce, afin de diminuer la variabilité des données de

même que d'en faciliter la manipulation subséquente. C'est via cet encodage qu'il est possible de transformer les données qualitatives en données quantitatives rendant possible l'analyse statistique de celles-ci [7 ; 23]. La quatrième manipulation nommée *connections* ou connexions consiste à établir des relations entre des éléments de données qui ne sont pas nécessairement adjacents ou du même type. *Comparisons* ou la comparaison, comme son nom l'indique est la comparaison de différents éléments entre eux afin de comparer par exemple l'encodage des mêmes données brutes par différents analystes ou encore dans le but de comparer les données provenant de différents sujets, contextes ou conditions entre un modèle de prédiction et les données empiriques réelles. La sixième technique, *constraints* est l'établissement de contraintes ou de bornes aux données afin d'en isoler une partie pour appliquer des transformations subséquentes. *Conversions* ou la conversion, suggère la transformation des données permettant ainsi l'émergence de nouveaux patrons et finalement *computations*, propose de calculer des représentations de données sommaires telles qu'un simple décompte, une moyenne ou encore un test de signification statistique.

Le processus d'analyse avec l'ESDA tel que proposé par Fisher et Sanderson (1996) [7], est un processus itératif. Il s'agit sommairement de définir une série de concepts issus des questions de recherche qui nous intéressent et basés sur les traditions de l'ESDA qui nous préoccupent, lesquels guideront par la suite ce qu'il devra être observé parmi les données brutes collectées et quelles manipulations devront être effectuées afin d'obtenir des données dérivées sur lesquelles il est possible de s'appuyer pour générer de la théorie ou définir des hypothèses. Ce processus est itératif disons-nous, puisqu'il est souvent nécessaire de faire plusieurs retour arrières par exemple, pour venir ajouter, enlever ou redéfinir des concepts ou catégories qui servent également à l'encodage des données. Déjà, un certain nombre d'itérations a été effectué afin de peaufiner notre liste de catégories d'activités collaboratives.

3.2.6 Validité et fiabilité de la recherche. La validité des données réfère à la propriété selon laquelle la mesure empirique traduit fidèlement la réalité empirique du phénomène mesuré [1]. Dans la stratégie de recherche proposée, certaines mesures devront être misent en place afin d'éliminer le plus de biais que possible et d'assurer une certaine validité des résultats. Outre les quelques mesures expliquées dans la section sur la collecte de données, il est possible de dire que la capture sur bandes audio-vidéos et la saisie des transmissions des courriels nous assure d'une certaine complétude des données concernant le phénomène qui nous intéresse. En effet, suite à la période d'ethnographie qui

a précédé la phase de collecte de données, il a été noté que peu d'échanges pertinents s'effectuaient en dehors des espaces de bureaux, c'est-à-dire en dehors du champ de la caméra qui fut placée par la suite pendant la collecte.

Pour accroître la validité de la recherche, une attention particulière devra aussi être portée sur la définition des concepts ou catégories choisies afin d'encoder les données. Ces définitions de concepts devant être issus des traditions de l'ESDA qui nous préoccupent ainsi que du contexte du terrain sur lequel s'effectue la recherche, cela nous assure une certaine représentativité du phénomène à l'étude par une validité référée dans la littérature comme relevant du sens commun (face validity) [1 ; 13].

Les concepts ou catégories sous lesquels seront encodées les données de même que le nombre de catégories choisies seront également très importants en ce qui a trait à la validité de contenu (content validity) [1]. Cet aspect de la validité réfère plutôt à la couverture des sens couverts par les concepts. Par exemple, regrouper tous les échanges collaboratifs sous une seule catégorie serait faire preuve de réductionnisme et trahirait tous les concepts sous-jacents servant à cerner la complexité de l'ensemble des activités collaboratives. De plus, la validité des connexions ou des relations (construct validity) [1] devra être assurée entre les concepts formant le modèle théorique émergeant des données. Cela pourra être fait par le biais de certaines mesures de corrélations ou d'associations statistiques, par exemple en formant une matrice de transitions ou de changements d'états entre les différents concepts pour démontrer la fréquence du changement d'un état ou d'un concept à un autre.

Finalement, une triangulation de données sera effectuée entre données qualitatives et quantitatives de même que de données provenant de différentes sources [23 ; 25]. Sur ce dernier point, d'autres phases de collecte de données sous formes de questionnaires ou d'entrevues sont à prévoir.

La fiabilité de la méthode ou d'une mesure réfère quant à elle, à la capacité à reproduire les mêmes résultats dans les mêmes conditions. Dans ce cas-ci, les techniques de collectes de données utilisées sont fortement adaptées au contexte du terrain sur lequel s'effectue la recherche ce qui rend donc la reproductibilité de la technique de mesure plus difficile. Toutefois, il sera très important que les concepts définissants les activités collaboratives *ad hoc* soient transférables d'un contexte ou d'un terrain à un autre pour que le modèle théorique tienne la route en dehors du cas spécifique étudié. Comme nous l'avons mentionné plus tôt, la définition des concepts utilisés sera donc une étape cruciale de cette recherche.

4. RÉSULTATS INITIAUX OBTENUS

Les résultats présentés dans la présente section sont le fruit d'une analyse préliminaire concernant quatre des huit développeurs de l'équipe observée sur une période de 8 heures. Le choix des personnes ne s'est pas fait à l'aide d'une méthode d'échantillonnage quelconque. Les personnes choisies l'ont été simplement parce qu'à partir d'observations directes sur le terrain, elles avaient été identifiées comme étant susceptibles de faire plus de travail collaboratif que les autres. Ce choix est justifié puisque ce qui est recherché par le biais de cette recherche n'est pas de trouver un nombre magique indiquant le temps passé à faire du travail collaboratif *ad hoc*, mais plutôt d'investiguer le contenu de ces activités collaboratives. Ainsi, ces personnes pouvaient potentiellement nous donner plus de matière à étudier. Il est également à noter que les résultats ci-dessous ne tiennent pas comptes des interactions par courriels, mais seulement que des conversations de pair à pair et téléphoniques qui ont été captées par la caméra.

Voici donc les quelques résultats initiaux obtenus.

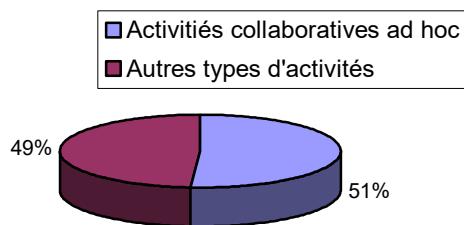


Figure 1. Distribution du temps occupé aux activités collaboratives *ad hoc* par rapport aux autres types d'activités

Comme il est possible de le constater par la Figure 1, 51% du temps est occupé au travail collaboratif *ad hoc* chez les quatre sujets observés par rapport aux autres types d'activités. Cela tend vers les observations directes faites sur le terrain qui suggéraient l'importance du phénomène.

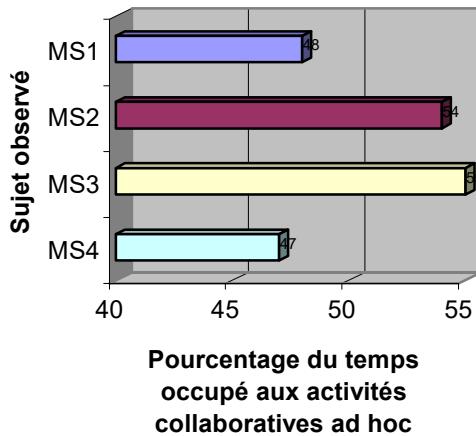


Figure 2. Pourcentage du temps occupé aux activités collaboratives *ad hoc* par sujet observé

La Figure 2 indique le pourcentage du temps occupé aux activités collaboratives *ad hoc* par sujet observé. Comme il a été constaté sur le terrain, les sujets MS2 et MS3 semblent passer une grande partie de leur temps à collaborer et communiquer informellement et de façon spontanée avec leurs collègues. Cette tendance se maintient lorsque nous analysons le nombre moyen d'interactions à l'heure par sujet observé tel qu'illustré à la Figure 3.

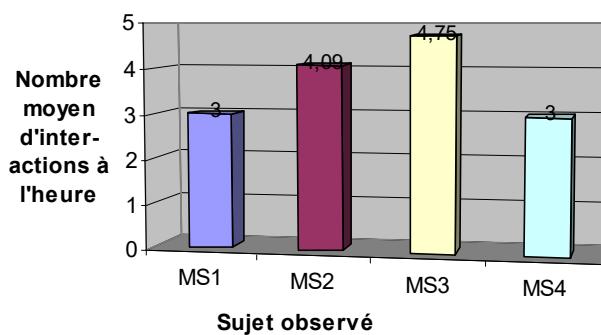


Figure 3. Nombre moyen d'interaction à l'heure par sujet observé

Cette observation peut s'expliquer par la nature du travail réalisé par les sujets MS2 et MS3. En effet MS2 tient lieu du gestionnaire de projet au sein de l'équipe et l'un de ses rôles est de faire circuler les informations pertinentes au travail des développeurs de son équipe. Lorsque l'on examine plus attentivement les interactions dans lesquelles MS2 est impliqués, on peut s'apercevoir que dans 78.13% du temps, ce sont ses collègues qui

initient les interactions. Cela peut hypothétiquement suggérer que MS2 constitue une source d'informations nécessaires à ses collègues. Toutefois, il a été noté sur le terrain que beaucoup d'informations transmises par MS2 à ses collègues se font par courriels. Il serait donc intéressant d'investiguer de ce côté. MS3 quant à lui est l'un des responsables de l'infrastructure du logiciel bâtit et souvent la personne consultée pour résoudre certains problèmes. Il mène par ailleurs plusieurs tâches de front ce qui l'amène à devoir communiquer plus souvent avec plusieurs de ses collègues.

D'autre part, la durée moyenne des interactions analysées des quatre développeurs est de 6 minutes et 31 secondes et ces interactions impliquent en moyenne 2.3 intervenants. Rappelons qu'une interaction est défini comme étant une unité communicative qui présente une évidente continuité interne, alors qu'elle rompt avec ce qui la précède et la suit [14]. Par ailleurs, ces résultats ont été basés sur un total de 82 interactions observées.

La Figure 4 et la Figure 5 plus bas donnent un premier aperçu d'une distribution en pourcentage par rapport au nombre d'occurrences et au temps occupé dans les diverses catégories d'activités collaboratives *ad hoc* identifiées. Ces catégories ont été définies comme ceci :

La « synchronisation cognitive » est lorsque deux ou plusieurs développeurs échangent de l'information afin d'être sûr qu'ils détiennent la même connaissance ou partagent la même représentation de l'objet dont il est question.

La « résolution de problème » est lorsque deux ou plusieurs développeurs connaissent l'existence d'un problème et tentent par divers moyens de régler le problème ou de pallier à celui-ci.

Le « développement » est lorsque deux ou plusieurs développeurs s'adonnent au développement d'une nouvelle fonctionnalité, particularité ou composante du logiciel.

La « gestion » est lorsque deux ou plusieurs développeurs discutent afin de coordonner et planifier des activités telles que des réunions, des séances de travail communes ou encore pour fixer des échéanciers.

La « résolution de conflit » est lorsque deux ou plusieurs développeurs discutent afin de régler un différent ou une divergence d'opinion.

Les activités collaboratives *ad hoc* sous la catégorie « non-pertinent » sont celles dont il n'est pas question du projet ou encore du logiciel construit.

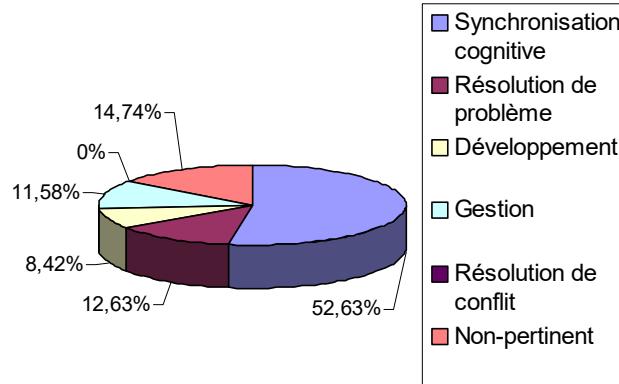


Figure 4. Distribution en pourcentage par rapport au nombre de d'occurrences des activités collaboratives *ad hoc* identifiées

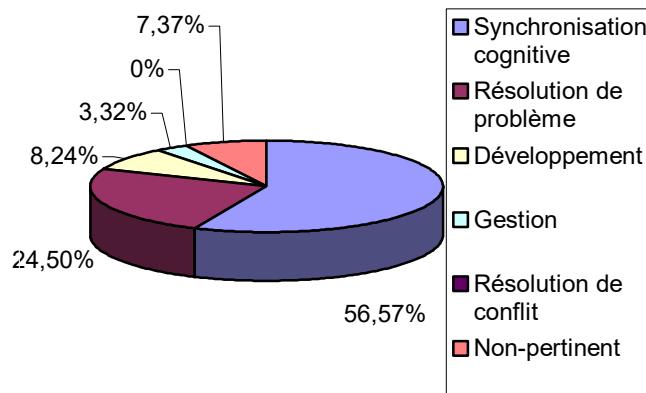


Figure 5. Distribution en pourcentage par rapport au temps occupé dans les activités collaboratives *ad hoc* identifiées

Comme la Figure 4 le suggère, plus de la moitié, soit 52.63% des activités collaboratives *ad hoc* qui sont survenues sont des formes de synchronisation cognitive. Cela va également dans le même sens que les observations directes faites sur le terrain. Ce chiffre n'est pas surprenant quand on considère que l'échange d'informations et de connaissances constitue un élément essentiel au développement de logiciel afin de *cristalliser* [21] toutes les informations nécessaires en un logiciel de qualité et qui répond

aux besoins de l'usager. Aussi, la synchronisation cognitive occupe un pourcentage de 56.57% du temps occupé au travail collaboratif ce qui soutient également la dernière explication.

Autre fait notable, la résolution de problème ne semble guère importante en nombre d'occurrences à la lecture de la Figure 4. Toutefois, la Figure 5 suggère que cette activité accapare presque le quart du temps occupé au travail collaboratif des quatre sujets observés. Cela démontre peut-être que les activités de résolution de problème ne sont pas si nombreuses, mais monopolisent un temps assez appréciable lorsqu'elles surviennent. Cela est d'ailleurs démontré par la Figure 6.

Les résultats relatifs aux activités de gestion comportent également une particularité intéressante. À l'inverse des activités de résolution de problème, elles sont nombreuses par rapport au temps relativement court qu'elles occupent. Cela tend peut-être à corroborer les théories de certains auteurs [10] qui maintiennent que les communications informelles sont nécessaires pour que les membres d'une équipe puissent coordonner efficacement leurs activités.

Quant à elles, aucune activité de résolution de conflit n'a été recensée. Toutefois, nous croyons que cette catégorie est tout à fait pertinente puisque malheureusement, des situations de résolution de conflit sont une réalité dans bien des équipes de développement de logiciel. Aussi, la liste de catégories devant être exhaustive et aussi complète que possible, il nous est impossible d'outrepasser cette réalité. Quant au 0% de résolution de conflit obtenu dans le cas qui nous préoccupent, peut-être pouvons nous faire la supposition avec les observations directes sur le terrain que l'équipe de développement observée a tous les attributs qui correspondent à la définition d'un « jelled team » selon Robillard, Kruchten et D'Astous [21].

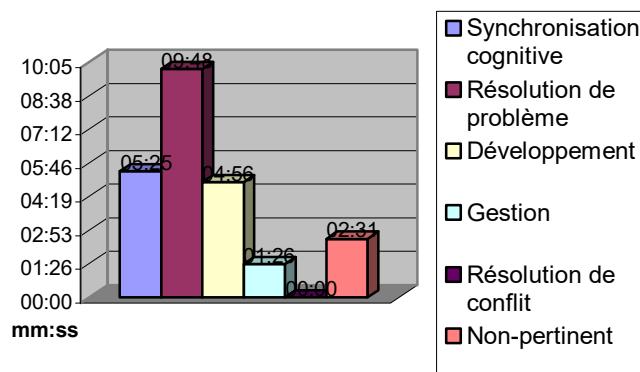


Figure 6. Temps moyen occupé par séquence de chaque type d'activités collaboratives *ad hoc* identifiées

Enfin, il est important de réitérer que les données ont été récoltées dans un contexte de maintenance et que nous aurions peut-être obtenu des résultats différents dans une situation de développement. Toutefois, comme les contextes de maintenance sont très courants dans l'industrie, cela n'affecte en rien la pertinence de cette recherche.

5. CONCLUSIONS

Il est clair et largement soutenu qu'une bonne collaboration et communication sont des conditions essentielles pour que le travail d'une équipe de développement de logiciels puisse se faire rondement afin de livrer un produit de qualité, qui correspond aux besoins de l'usager dans les temps et les coûts prévus.

Il a été trouvé par quelques recherches antérieures que les activités collaboratives *ad hoc* et les communications informelles occupent une portion fort appréciable du temps qu'un développeur consacre à un projet logiciel. Cependant, aucune recherche n'a tenté de décrire le contenu de ces activités laissant un vaste champ de recherche encore inexploré.

La recherche empirique décrite dans le présent document propose d'investiguer ce terrain parce que nous croyons que de comprendre comment les gens communiquent et collaborent nous permettra en bout de ligne de proposer des pratiques rendant la collaboration et la communication plus efficaces au sein d'une équipe de développement améliorant du même coup les processus de développement de logiciels, chose qui sera également bénéfique dans un contexte de développement distribué.

Ce document a présenté sommairement la méthodologie employée pour répondre aux objectifs de recherche définis. La définition de cette méthodologie a été fortement influencée par diverses recherches empiriques antérieures qui se sont également intéressées aux aspects humains du génie logiciel, mais celle-ci fut toutefois adaptée au contexte du terrain sur lequel se déroule cette étude.

Les quelques résultats embryonnaires qui furent présentés en partie dans ce document et qui sont issus d'une infime portion de la quantité appréciable de données qui fut collectée sont très intéressants. Même s'il reste encore beaucoup de travail et d'analyse à effectuer, un modèle de données et des patrons semblent déjà se dessiner ce qui nous permettra par la suite d'élaborer des hypothèses qui pourront être validées par d'autres recherches confirmatoires forgeant ainsi une base de connaissances dans ce domaine encore méconnu du génie logiciel.

Finalement, étant donné le caractère novateur, la contribution scientifique et l'envergure de ce projet de recherche, il serait préférable que celui-ci soit porté immédiatement au niveau du doctorat et ce, afin de saisir l'opportunité de recherche ainsi que de conserver l'originalité et la globalité du projet.

6. REMERCIEMENTS

Cette recherche n'aurait pas été possible sans l'accord de l'entreprise dans laquelle se déroule l'étude et bien entendu, sans la généreuse participation et patience des membres de l'équipe de développement dans laquelle les données sont collectées. À tous ces gens, mille mercis.

7. RÉFÉRENCES

- [1] BABBIE, Earl. *The Practice of Social Research*. 2001. 9^{ème} édition. International Thomson Publishing Company. 448p.
- [2] BRIAND, Lionel., EMAN, Khaled El., MELO, Walcélio L. 1995. « AINSI: An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines ». *Proceedings of the ESI-ISCN'95: Measurement and Training Based Process Improvement*. Vienne, Autriche.
- [3] BROOKS, Frederick P. jr. 1987. « No Silver Bullet: Essence and Accident of software Engineering ». *IEEE Computer*. 20 :4. 10-19.
- [4] D'ASTOUS, Patrick., ROBILLARD, Pierre N. 1997. *Les aspects de l'échange d'information dans un processus de génie logiciel*. Montréal : Éditions de l'École Polytechnique de Montréal. EPM/RT-97-06. 49p.
- [5] D'ASTOUS, Patrick., ROBILLARD, Pierre N. 1997. *La mesure des activités collaboratives retrouvées lors d'une réunion de révision technique du processus de génie logiciel*. Montréal : Éditions de l'École Polytechnique de Montréal. EPM/RT-97/30. 50p.
- [6] FENTON, Norman E., PFLEEGER, Shari Lawrence. 1997. *Software Metrics - A Rigorous & Practical Approach*. PWS Publishing Company. 638p.
- [7] FISHER, Carolanne., SANDERSON, Penelope. 1996. « Exploratory Sequential Data Analysis: Exploring Continuous Observational Data ». *Interactions*. 3:2. 25-34.
- [8] GIBBS, W. Wayt. 1994. « Software's chronic crisis ». *Scientific American (International Edition)*. 72-81.
- [9] GRINTER, Rebecca E., HERBSLEB, James D., PERRY, Dewayne E. 1999. « The geography of coordination: Dealing with distance in R&D work ». *Proceedings of SIGGROUP Conference on Supporting Group Work*. New York : ACM Press. 306-315.
- [10] HERBSLEB, James D., GRINTER, Rebecca E. 1999. « Splitting the Organization and Integrating the Code: Conway's Law Revisited ». *Proceedings of the 21st International Conference on Software Engineering*. Los Angeles, CA : IEEE Computer Society Press. 85-95.
- [11] HERBSLEB, James D., MOITRA, Deependra. 2001. « Guest Editors' Introduction: Global Software Development ». *IEEE Software*. 18:2. 16-20.
- [12] HERBSLEB, James D., MOCKUS, Audris. 2003. « An empirical study of speed and communication in globally-distributed software development ». *IEEE Transactions on Software Engineering*. 29:6. 481-494.
- [13] JORGENSEN, Danny L. 1989. *Participant Observation A Methodology for Human Studies*. New-bury Park, CA : Sage Publications. 136p.
- [14] KERBRAT-ORRECCHIONI, Catherine. 1998. *Les Interactions Verbales*. Armand Colin.

- [15] KITCHENHAM, Barbara A., PFLEEGER, Shari Lawrence., PICKARD, Lesley M., JONES. Peter W., HOAGLIN. David C., EMAN, Khaled El., ROSENBERG. Jarrett. 2002. « Preliminary guidelines for empirical research in software engineering ». *IEEE Transactions on Software Engineering*. 28:8. 721–734.
- [16] NAUR, Peter., RANDELL, BRIAN. 1969. *Software Engineering: Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968*. Brussels, Bel : NATO Scientific Affairs Division. 136p.
- [17] PRESSMAN, Roger S. 1996. *Software Engineering: A Practitioner's Approach*. 4^{ième} édition. New York : McGraw-Hill. 852p.
- [18] PERRY, Dewayne E., STAUDENMAYER, Nancy A., VOTTA, Lawrence G. 1994. « People, Organizations, and Process Improvement ». *IEEE Software*. 11:4. 36-45.
- [19] ROBILLARD, Pierre N. 1996. *Études des aspects cognitifs applicables au génie logiciel*. Montréal : Éditions de l'École Polytechnique de Montréal. EPM/RT-96/18. 49p.
- [20] ROBILLARD, Pierre N., ROBILLARD, Martin P. 2000. Types of Collaborative Work in Software Engineering ». *The Journal of System and Software*. 53. 219-224.
- [21] ROBILLARD, Pierre N., KRUCHTEN, Philippe., D'ASTOUS, Patrick. 2002. *Software Engineering Process with the UPEDU*. Addison Wesley. Pearson Education. 346p.
- [22] SEAMAN, Carolyn. 1996. *Organizational Issues in Software Development: An Empirical Study of Communication*. 123p. Thèse de doctorat en informatique. University of Maryland.
- [23] SEAMAN, Carolyn. 1999. « Qualitative methods in empirical studies of software engineering ». *IEEE Transactions on Software Engineering*. 25:4. 557–572.
- [24] TICHY, Walter F. 1998. « Should computer scientists experiment more? ». *Computer*. 31:5. 32–40.
- [25] WALKER, Robert J., BRIAND, Lionel C., NOTKIN, David., SEAMAN, Carolyn B., TICHY, Walter F. 2003. « Panel: Empirical Validation-What, Why, When, and How ». *Proceedings of the 25th International Conference on Software Engineering*. Washington DC: IEEE Computer Society. 721-722.

L'École Polytechnique se spécialise dans la formation d'ingénieurs et la recherche en ingénierie depuis 1873



École Polytechnique de Montréal

**École affiliée à l'Université
de Montréal**

Campus de l'Université de Montréal
C.P. 6079, succ. Centre-ville
Montréal (Québec)
Canada H3C 3A7

www.polymtl.ca



**ÉCOLE
POLYTECHNIQUE
MONTREAL**