Comparing two Communication Media in Use Case Modeling: Results from a Controlled Experiment

Ugo Erra, Ana Portnova, Giuseppe Scanniello
Dipartimento di Matematica e Informatica
Università della Basilicata
ugo.erra@unibas.it, ana.portnova@gmail.com, giuseppe.scanniello@unibas.it

ABSTRACT

A critical claim in software requirements regards the assertion that the team performances improve when media with higher richness levels are used. To investigate this claim, we have conducted a controlled experiment to compare traditional face-to-face communication, the richest medium, and a leaner medium, namely an advanced chat implementing a distributed version of Think-Pair-Square (i.e., a well known method for collaborative problem solving). The comparison has been performed considering the time needed to model functional requirements through a use case modeling technique. Since the only assessment of time could be meaningless, we have also analyzed the media effect on the quality of the produced use cases. The results indicate a significant difference in terms of time to model software requirements in favor of face-to-face communication with no significant impact on the quality.

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: :Methodologies

General Terms

Experimentation, Measurement

1. INTRODUCTION

Splitting the development of a same software product among globally distributed sites is increasingly becoming a common practice in the software industry. Academy and industry refer to this relevant phenomenon as global, distributed, or multi-site software development [9]. Many advances have been made in the global development since it offers several benefits to software organizations, such as working cost reduction, enhanced availability of skilled development staff, proximity to the market, flexibility and efficiency for in-house staff usage to adapt quickly to volatile business needs.

Also well established and consolidated practices may become a challenge in a distributed software development setting due to the impact of distance and temporal, geographical, and socio-cultural differences. In this context, empirical studies become a useful and powerful tool to assess processes, methods, and tools for global software development. In particular, controlled experiments may be employed since they can be used to effectively test cause-effect relationships [10].

In this paper, we present a controlled experiment to compare two communication media, with different richness levels, in the modeling of functional requirements using use cases. The richest medium is the face-to-face communication, the leaner medium is a text-based chat implementing a well known approach for collaborative problem solving, namely Think-Pair-Square [8].

The remainder of the paper is organized as follows. Section 2 presents some background material and related work. The study definition and details of the adopted experiment design are presented in Section 3, while results are discussed in Section 4. Discussion and final remarks conclude the paper.

ESEM '10, September 16-17, 2010, Bolzano-Bozen, Italy.

2. BACKGROUND AND RELATED WORK

In the following, we present the background useful to better comprehend our contribution as well as research work related to the distributed requirements engineering.

2.1 Use Case Modeling

Use cases represent a popular modeling technique to capture and define software requirements. They are textual descriptions based on a more or less rigorous template to specify a sequence of simple steps to describe the interaction between one or more actors and the system [7]. The template adopted in the experiment is that proposed by Bruegge and Dutoit in [1]. It includes: the name of the use case, the participant actors, the entry conditions, the flow of events, the exit conditions, and the quality requirements.

2.2 Think-Pair-Square

Think-Pair-Square [8] is a method originally conceived for solving problems in the collaborative learning field. Students are grouped in homogeneous or heterogeneous way and are asked to accomplish the following three steps/phases: **Think** (it is individually accomplished to approach a solution for a problem); **Pair** (the students work on the problem in a pair, sharing the possible solutions individually identified in the previous step); **Square**(the pairs are grouped with the other pairs and work on the problem solution. Pairs share the work made in the previous step). Generally, stu-

dents perform the Pair and Square steps in the same physical setting using face-to-face communication. In our study we modified the original definition of the method replacing the face-to-face communication with a text-based structured chat, thus making the method suitable to solve problems in distributed settings. This represents another contribution of the paper.

2.3 CoFFEE

CoFFEE [4] has an extensible architecture designed to mediate the interaction of face-to-face group discussions. Each discussion is defined by a session that is divided in steps. The activities that can be accomplished within each step are defined combining one or more tools. In the experiment, we used the chat and the threaded discussion tools. The first tool is a traditional text-based chat, while the latter enhances a chat tool structuring a discussion flow in threads. We have created here a session composed of three steps (one for each phase of Think-Pair-Square). In the first step, a threaded chat is provided to the subject. This tool has been mainly used to let the subjects specify the entries of the template to specify a use case (see Section 2.1). In the second and third step, the subjects had a traditional text-based chat to support the discussion with the other students. To effectively support Think-Pair-Square, at each step the content of the threaded chat is made to be available to the subsequent one.

2.4 Related Work

Similarly to the traditional software development, global software development requires that software engineers spend a large part of their time to communicate both directly, through meetings and informal conversations, and indirectly, by means of software artifacts. An example, where the communication plays a relevant role, both in the traditional and distributed software engineering, is the requirements engineering process [6]. The effect of different communication media in the requirements engineering process has been marginally investigated in the literature. These studies are generally related to media theories, where a critical claim regards the assertion that the performance improves when a medium with the appropriate richness is used. These contributions do not deal directly with work groups or with how technology affects them.

Damian *et al.* [6] show an empirical study to compare five physical group configurations: one face-to-face and four distributed. A study on the effect of using synchronous unstructured text-based communication in distributed requirements workshops is proposed in [2]. Differently, Erra and Scanniello in [5] present an empirical study, where a traditional face-to-face meeting, an enhanced chat, and a three dimensional virtual environment are compared in the requirements negotiation. An interesting study on the effect of using mixed media (i.e., rich and lean) in distributed requirements negotiations is proposed in [3]. The main difference between our study and the ones above mentioned is that we investigate the effect of using a distributed implementation of Think-Pair-Square in a requirements engineering task.

3. THE CONTROLLED EXPERIMENT

In this section, we present the definition, the design, and the planning of the experiment, structured according to the guidelines suggested in [10]. For replication purposes an experimental package is available on line[1]. A technical report where further details on the experiment is available as well.

3.1 Experiment Definition

The media-effects theories suggest that face-to-face is the richest medium with respect to all the other media. Moreover, most theories assert that the negotiation performance decreases when leaner media are used. These claims were not perfectly supported by the results of our previous study [5]. In fact, we observed a difference in favor of face-to-face meeting on the time to negotiate software requirements, while the negotiation quality was not affected by the used media. Accordingly, it seems reasonable to assess whether a communication media implementing Think-Pair-Square is effective as the face-to-face communication.

3.2 Context

The context of the experiment was constituted of students in Computer Science at the University of Basilicata. In particular, 27 students were from a Software Engineering course of the Bachelor program, while 9 students were from a Computer Graphics course of the Master program. As laboratory activity of the Software Engineering course, the students were grouped in teams and allocated on software projects. They were also encouraged to conduct scheduled and unplanned meetings. The Master students have passed a Software Engineering course. In this course, they were also asked to design a software system conducting scheduled and unplanned meetings to disseminate and share project information. The laboratory activity of the Computer Graphics course was mainly based on the development of video games (e.g., doom-like game). Students worked individually or in small teams.

3.3 Hypotheses Formulation

The media used in this experiment are all synchronous (i.e., the communication happens in real time), while they have a different space dimension. This means that subjects are virtually and physically collocated when experimented the text-based chat implementing Think-Pair-Square (in the following we will refer to this media as TPS) and the face-to-face communication (F2F in the following), respectively. Since we are particularly interested in whether virtually collocated subjects may obtain similar performances than physically collocated ones to model use cases, the following two sided null hypotheses have been defined:

Hn1- There is no significant time difference to model a use case when using F2F and TPS.

Hn2- There is no significant difference in terms of the use case quality when using F2F and TPS.

In case these hypotheses can be rejected, it is possible to formulate the corresponding alternative hypotheses Ha1 and Ha2, which can be easily derived.

3.4 Selected Variables

In order to properly design and then analyze the results, the main factor on which our study is focused on is *Method* (i.e., F2F and TPS). We considered two tasks (i.e., Task1 and Task2) on two software systems: a car rental software to manage available cars, customers, and reservations (in the following we will refer to it as *Rent*) and an E-Commerce

[1] www.scienzemfn.unisa.it/scanniello/TPS_UCD

Table 1: Experiment Design

Groups	Task1	Task2
A	Rent, TPS	ECP, F2F
B	Rent, F2F	ECP, TPS

Platform to order CDs and books via the Internet from an on line catalogue (named *ECP* in the following). The systems are similar in complexity and refer to application domains on which the subjects are not completely familiar with. On each software system we selected a task. In particular, on Rent the subjects were asked to model the use case *insert a new reservation contract to hire a car*, while on ECP *search for a book within the on-line catalog* was asked to model.

To verify the null hypotheses, we considered the following dependent variables: **Time** (indicates the minutes that all the subjects within a team spent to perform the task); **Defects** (shows the number of defects in a use case).

To identify the number of defects within the use case produced by each team (both using F2F and TPS), we used an inspection process based on a checklist (see the technical report for details). The authors individually conducted the inspection and then a meeting to solve possible conflicts.

3.5 Pilot and Experimental Procedure

A pilot experiment was performed with four students of the Bachelor program in Computer Science at the University of Basilicata. They were volunteers and were not successively involved in the controlled experiment. The goal of the pilot was twofold: *(i)* to get some indications on the complexity and the needed time to accomplish the tasks; *(ii)* to test TPS. This study revealed that the tasks were well suited and indicates minor changes to be accomplished on the experimental material.

The day before performing the controlled experiment, the subjects attended a training session to provide them an equal prior knowledge on TPS. The goal here was also to increase the tendency for the subjects to stick together and remain united in the pursuit of the same objectives (i.e., team cohesion). Once concluded this session, the subjects were asked to fill in a pre-questionnaire to obtain information on: working experience, passed exams, and grade point average. We used the gathered information to distribute high and low ability subjects among the teams of the actual experiment.

3.6 Design and Preparation

The experiment was performed in a controlled laboratory setting according to the design summarized in Table 1. The adopted design ensured that each team worked on two different objects (i.e., Rent and ECP) in two tasks (i.e., Task1 and Task2), receiving each time a different Treatment (i.e., TPS or F2F). The teams were randomly assigned to the groups A and B. In particular, 5 and 4 teams composed of 4 students (3 Bachelor and 1 Master) were assigned to A and B, respectively.

At the end of the experiment, the subjects were asked to fill in a post experiment survey questionnaire. This questionnaire mainly aimed at getting the perceived: *(i)* quality of the provided material, *(ii)* effort to perform the task and its clearness, and *(iii)* efficacy of the used media.

Regarding the PCs preparation, we installed the CoFFEE controller [4] on 5 desktops (i.e., the servers) to let the subjects accomplish the tasks with TPS. On the other hand, we installed on the subjects' PCs the plug-in CoFFEE discusser [4]. The subjects machine were connected to the servers using a LAN network.

3.7 Execution and Data Analysis

In order to carry out the experiment, the subjects were provided with a pencil, some paper sheets, and the paper copy of following material: *(i)* the introductory presentation of both the experiment and TPS; *(ii)* the problem statements[2] of Rent and ECP; *(iii)* the template to use in the use case modeling; *(iv)* the post-experiment survey questionnaire.

The size of the text within the problem statements of the two systems was nearly the same. Indeed, they differ as the problem statement of ECP included some screen mockups [1] to clarify some functionality to implement. A screen mockup of the functionality *search a book within the on-line catalog* was also included in this document. This difference was deliberately introduced and properly controlled.

To verify the null hypotheses, non-parametric statistical tests have been used. In particular, we employed the Wilcoxon test since it has been applied in the literature for purposes similarly to ours. We also measured the effect of a co-factor, namely Task, on the dependent variables using a two-way ANOVA. To apply this test, four assumptions should be verified. However, in the literature ANOVA is used when the assumptions are not valid or, as in our case, the sample size is small. This is possible tanks to the robustness of the test. In all our statistical tests, we decided (as usual) to accept a probability of 5% of committing type-I-error [10], i.e., of rejecting the null hypothesis when it is actually true.

4. RESULTS

Some descriptive statistics (i.e., median, mean, and standard deviation) on Time and Defects are shown in Tables 2 and 3, respectively.

4.1 Data Analysis Results

The Wilcoxon test revealed that Hn1 can be rejected (p-value = 0.008). This together with the descriptive statistics indicate that significantly less time is needed to accomplish the task with F2F. Differently, Hn2 cannot be rejected (p-value = 0.18), thus suggesting that there is not significant difference in the quality of the produced use cases when using TPS or F2F.

Table 4 summarizes the results achieved by applying two-way ANOVA on Method and Task. The table also presents the results of the interaction between Method and Task. The test reveals a significant interaction between Method and Task on the dependent variable Time. Note that ANOVA confirms a positive effect of Method on Time.

4.2 Survey Questionnaire Results

The analysis of the post-experiment survey questionnaire showed that the time to carry out the experiment was judged appropriate and the objectives and the tasks were considered clear. Also, the complexity of tasks was judged appropriate. The greater part of the subjects found useful TPS to model use cases in cooperation and distributed fashion. Finally,

[2]The problem statement describes the current situation, the functionality the system should support, and the environment in which it will be deployed.

Table 2: Descriptive Statistic on Time

Method	Rent			ECP		
	Med.	Mean	Std. Dev.	Med.	Mean	Std. Dev.
TPS	155	150.2	25.95	195.5	196.5	18.77
F2F	60	67	16.77	60	60.8	33.39

Table 3: Descriptive Statistic on Defects

Method	Rent			ECP		
	Med.	Mean	Std. Dev.	Med.	Mean	Std. Dev.
TPS	0	1.2	2.16	0.5	1	1.41
F2F	0.5	1.5	2.38	1	2.2	2.58

the experiment from the pedagogical point of view was considered very useful. Further details can be found in the technical report.

4.3 Threats to Validity

In the following we present the threats that could affect the validity of the experiment results. The internal validity threats were mitigated by the experiment design. However, there is still the risk that subjects might have learned how to better perform the experimental tasks. Another issue concerns the possible information exchanged among the subjects. We prevented this in several ways. To further reduce the internal validity, subjects did not know the hypotheses and were not evaluated on their performance.

External validity may present when experiments are conducted with students, since they could not be representative as software professionals. In the experiment presented here, students could be more comfortable with TPS than professionals. This is unusual enough for student experiments. In our study the deliberate use of students may also bring out phenomena related to the learning of new communication media. In fact, they could be more comfortable to learn the new investigated medium. To increase our confidence in the achieved results, it will be also important to investigate in case that subjects have cultural and background diversities. Another threat may be represented by the template adopted to model use cases. This represents a possible future direction for our work.

The construct validity threats concern the possibility that the relationship between cause and effect is causal. This validity was mitigated by a proper design that allowed separating the analysis of the different treatments. Also, the selection and the measurement of the dependent variables could threaten the construct validity.

The desing of the experiment also allowed mitigating the conclusion validity threat. This threat is also related to the selection of the population. This threat was mitigated as the subjects were not far from junior developers and software engineers. Moreover, to verify the defined null hypotheses non-parametric tests were used. In case differences were present but not significant, this was explicitly mentioned and discussed. The conclusion validity could be also affected by the sample size. Accordingly, further replications on a larger sample are needed.

5. DISCUSSION AND FINAL REMARKS

The results of the experiment presented here have practical implications with respect to the computer-mediated environment. In fact, in case the time distance is not an issue, but moving people might be a problem, a communication medium based on Think-Pair-Square could be a viable solution. Furthermore, the studied medium is simple to use, set up, and maintain [6].

Table 4: ANOVA Results

Factor	Time	Defects
Method	0.00	0.487
Task	0.119	0.815
Method vs. Task	0.047	0.675

The data analysis also showed a significant interaction between Method and Task on Time. In particular, the subjects spent on average more time on the ECP system when using TPS and less time on the same system when employing F2F (see Table 2). This could be due to the fact that the problem statement of the ECP system included some screen mockups [1] for explaining relevant functionality, which could have increased the effort to accomplish the task when using TPS. On the other side, screen mockups could have better supported subjects when used F2F. This is an interesting point that needs a further investigation. A further analysis has been conducted on the subjects' performances among the steps of Think-Pair-Square. As expected, this analysis indicated that on average the performances increased through the steps. Further on we plan to study the effect of these steps on subjects' performances.

6. REFERENCES

[1] B. Bruegge and A. H. Dutoit. *Object-Oriented Software Engineering: Using UML, Patterns and Java, Second Edition*. Prentice Hall, 2003.

[2] F. Calefato, D. Damian, and F. Lanubile. An empirical investigation on text-based communication in distributed requirements workshops. In *Proc. of Intl. Conf.on Global Soft. Eng.*, pages 3–11, Washington, DC, USA, 2007. IEEE Computer Society.

[3] D. Damian, F. Lanubile, and T. Mallardo. On the need for mixed media in distributed requirements negotiations. *IEEE Trans. Software Eng.*, 34(1):116–132, 2008.

[4] R. De Chiara, A. Di Matteo, I. Manno, and V. Scarano. Coffee : Cooperative face2face educational environment. In *CollaborateCom*, pages 243–252, 2007.

[5] U. Erra and G. Scanniello. Assessing communication media richness in requirements negotiation. *IET Software*, 4(2):134-148, 2010.

[6] D. E. Herlea Damian, A. Eberlein, M. L. G. Shaw, and B. R. Gaines. Using different communication media in requirements negotiation. *IEEE Soft.*, 17(3):28–36, 2000.

[7] M. Jackson. *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. Addison Wesley, 1995.

[8] F. T. Lyman. The responsive classroom discussion: The inclusion of all students. In *In A. Anderson (Ed.), Mainstreaming digest*, pages 109–113. College Park, MD: University of Maryland Press, 1981.

[9] B. Sengupta, S. Chandra, and V. Sinha. A research agenda for distributed software development. In *Proc. Intl. Conf.on Soft. Eng.*, pages 731–740, New York, NY, USA, 2006. ACM.

[10] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.