An Empirical Study of Distributed Software Maintenance

Alessandro Bianchi*, Danilo Caivano*, Filippo Lanubile*, Francesco Rago° Giuseppe Visaggio*
*Dipartimento di Informatica – Università di Bari - Via Orabona, 4, 70126 Bari – Italy
{bianchi, caivano, lanubile, visaggio}@di.uniba.it
° Italy Solution Center, EDS Italia, Viale Edison, Lo Uttaro, 81100 - Caserta – Italy
francesco.rago@eds.com

Abstract

A large software project may be distributed over multiple sites when the organization believes that there are not enough people to staff a single collocated team. However, previous empirical research in the context of telecommunication organizations has shown that distance may increase cycle time and costs. We report on a large software massive maintenance project in the information systems domain, which in part has been carried out on a single site and in part across multiple sites of the same organization. We performed a comparative postmortem analysis of the two parts. Our results show that, with respect to cycle time and cost no significant differences exist among the distributed and collocated work. Indeed there is a significant difference in communication during project. This implies that for massive maintenance activities the distribution over multiple sites can be really helpful.

Keywords: Global Software Development, Post Mortem Analysis, Massive Maintenance

1. Introduction

The new forms of competition and cooperation that have arisen in software engineering as a result of the globalization process have had an impact on the whole software process. Software development and maintenance have become distributed across sites and now involve an increasing number of people with different cultural backgrounds. Carmel and Agarwal [1] report that at present, 50 different nations are collaborating in different ways in software development.

However, global software development has a number of drawbacks, which have been recognized by many authors, such as the need to apply ad hoc management methods [2], the need to use knowledge sharing tools [3, 4], and the overhead derived from staff communication interchanges [5]. Herbsleb and Moitra [6] classified the main drawbacks in global software development in a set of issues:

- *strategic issues*, concerning the decisions on how to divide the tasks among sites, so as to be able to work as independently as possible while maintaining efficient communication among sites;
- *cultural issues*, that arise when the staff come from different cultural backgrounds;
- *inadequate communication*, caused by the fact that geographical distribution of the staff over several sites increases the costs of formal communications among team members and limits the possibility of carrying on the informal interchanges that traditionally helped to share experiences and foster cooperation to attain the targets;
- *knowledge management*, that is more difficult in a distributed environment as information sharing may be slow and occur in a non uniform manner, thus limiting the opportunities for reuse;
- *project and process management issues*, having to do with all the problems of synchronization of the work at the various different sites;
- *technical issues*, that have an impact on the communication network linking the various sites.

Previous investigation on how geographical distribution affects software development and validation activities, have been carried out, respectively, at Lucent Technologies [7] and Alcatel [8]. Main findings were that distance negatively affects cost, time and quality. However, those studies were both conducted in the context of a telecommunication application domain and involved complex tasks.

Our research takes its rise from the hypothesis that the application domain and the software engineering task are both fundamental drivers of global software development costs and benefits. In other words, we suppose that previous results from first case studies represent an extreme case: on the opposite extreme lay the projects involving massive, well-defined and stable activities. For this kind of software projects, the distribution over different geographical sites would present just a project management overhead.

An explorative analysis was presented in [9]. In this paper we specifically look at the relationship between geographically distribution and project outcomes. Because of the context in which the present investigation was conducted, we focused on communication and project management issues.

To this end we pose the two following research questions.

- Is there any significant difference in process execution (i.e., duration, effort, staff and rework) when maintenance activities are executed on a single site rather than on multiple sites?
- Is there any significant difference in project management overhead when maintenance activities are executed on a single site rather than on multiple sites?

The paper is organized as follows: section 2 presents the maintenance project and the metrics used in the analysis; section 3 illustrates the data analysis; the results are discussed in section 4, and section 5 draws some conclusions.

2. Case Study Setting

2.1. Project Characterization

Our research can be characterized as a post mortem analysis on data concerning a maintenance project carried out by EDS-Italia. It was a large project requiring a high number of human resources to execute massive, non-routine maintenance of a large information system to solve the Y2K problem. The project involved 2 different geographically distant sites, both settled in Italy.

The software system had been decomposed into 4 *functional areas* (FA), each consisting of a variable number of *work-packages* (WP), each being assigned to a working team. FAs were partitioned in WPs according to some criteria established within the organization, which have not been taken into account in our research.

In total the software system is constituted by 100 WPs, and the maintenance effort had to deal with 52 of them. The size of each WP is expressed by the number of items, where an item can be a program, a library element or a Job Control Language (JCL) procedure, i.e., a procedure written in a script language to control the program execution in batch systems.

Each WP included a variable number of items, making up a total of 26,739 items with an average of 514.21 items per WP involved in the maintenance effort.

The maintenance project was executed according to a process (Figure 1) that was enacted for each WP:

- a *Project Management* phase, aimed at managing and scheduling the activities for the WP;
- a *Configuration Management* phase, aimed at collecting and identifying all the artifacts produced within the WP;
- a *Change* phase, aimed at executing the maintenance activities over the items belonging to the WP;
- a *Review* phase, aimed at looking for defects into the maintained items;
- a *Software Quality Assurance* (SQA) phase, aimed at verifying that the maintained items comply with the company's Quality System.

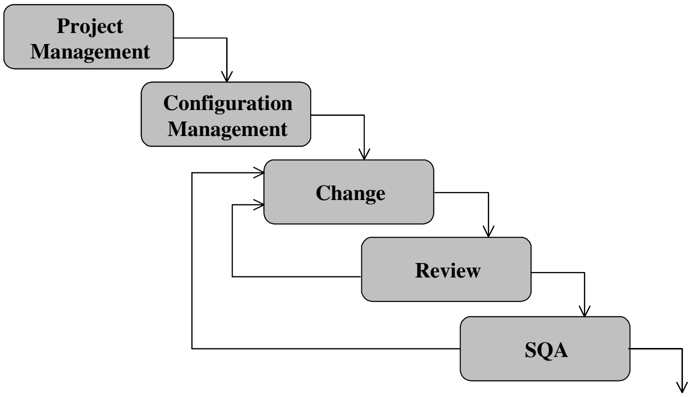

Figure 1. The process adopted for each WP in the maintenance project.

When the Review or the SQA phases identify defects, the maintained items are reworked looping from the Corrective phase.

For all the WPs, the Project Management established to start process execution on a single site (hereinafter referred to as *Site1*) but, depending on both rework needs and currently available resources, the execution of some phases (such as Change, Review and SQA) could also be switched to another site (hereinafter referred to as *Site2*).

This led to 17 WPs (33%) entirely executed at Site1 and 35 WPs (67%) executed at both sites.

According to Ebert and De Neve definition [5], we consider the WPs entirely executed at Site1 as part of a *collocated project*; conversely the WPs executed both at Site1 and Site2 as belonging to a *distributed project*.

In our context, the tasks executed in the collocated and distributed project did not present technical differences.

In fact, although the distributed project included about twice WPs than the collocated one, the number of items maintained was approximately the same: 14,163 items (53%) in the collocated project, and 12,576 items (47%) in the distributed one. Moreover, in both cases each item underwent a massive corrective maintenance, in a well-defined application domain well known by the maintainers.

Referring back to Herbsleb and Moitra classification of issues [6], from the above features of the project derives that:

- for what concerns the strategic issues, the WPs had been divided among sites according to staff availability; partitioning WPs was relatively easy because of the specific nature of the maintenance task to carry out;
- the two sites belonged to the same company and both were located in Italy, therefore problems related to different cultural backgrounds did not occur;
- no significant technical issues occurred during project execution related to the communication network linking the two sites;
- knowledge management issues were not critical: this was a massive maintenance project with loosely coupled WPs, and therefore the management of common knowledge was relatively easy.

Therefore, the remaining concerns generated by spreading work over distant sites are communication and project management issues.

2.2. Data Collection

The post-mortem analysis included all the work packages and covered the entire WP life cycle.

The following measures were collected:

- actual duration of the WPs execution, expressed as working days;
- effort spent to complete the WPs, expressed as working days/ person;
- staff size, i.e. number of people who took part in executing the WPs;
- number of rework cycles, i.e., number of times the working process had been repeated before WP completion;
- number of reports formally produced to describe the work progress;
- number of meetings officially held among the members working on the WPs;
- size of the WPs, expressed as number of items.

Since the WPs size is quite spread, with quartile values ranging from 68.5 items to 533 items (Figure 2), all our analysis were executed considering the metric values normalized with respect to WP size. We choose the number of items included into a WP as a normalizing factor for size because the organization uses it as a size measure for all WPs.

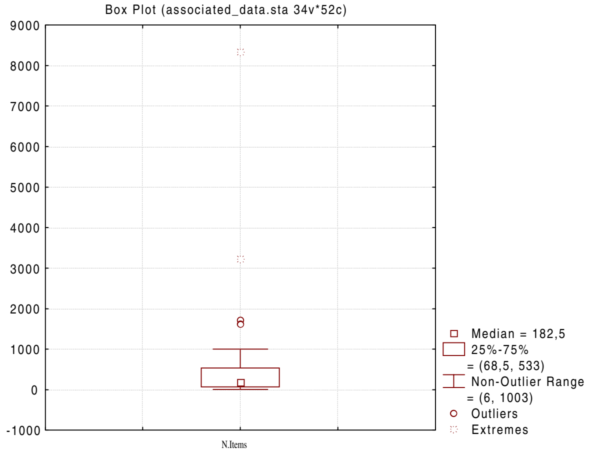

Figure 2 Boxplots of the WPs size expressed as number of items.

Therefore, the metrics taken into account during analysis have been calculated for each WP normalizing the values of the observed metrics over the number of items in a WP, that is WP size.

3. Data Analysis

In this section we analyze *process metrics* (i.e., duration of WPs, effort spent for their execution, required staff, number of rework cycles) and *project management metrics* (i.e., number of reports and number of meetings).

Because the two groups of observations that are to be compared are independent of each other, we might have used the *t*-test for independent samples. However, since the normality assumption was not always respected, we decided to use the Mann–Whitney *U* test. This nonparametric alternative can be applied when there are two samples from possibly different populations with the following assumptions [10]:

- both samples are random samples from their respective populations;

- in addition to independence within each sample, there is mutual independence between the two samples;
- the measurement scale is at least ordinal.

In our case, all the three previous assumptions were met.

In order to investigate whether the distribution between sites does affect process and project management metrics, for each metric M_i of this class the null and alternative hypotheses are formulated as follows:

Hi_0: There is no difference between the values of metric M_i for collocated WPs and for distributed WPs.

Hi_a: There is a difference between the values of metric M_i for collocated WPs and for distributed WPs.

3.1. Duration

The first analysis made on project data assessed the duration of the WPs executed on both collocated and distributed sites.

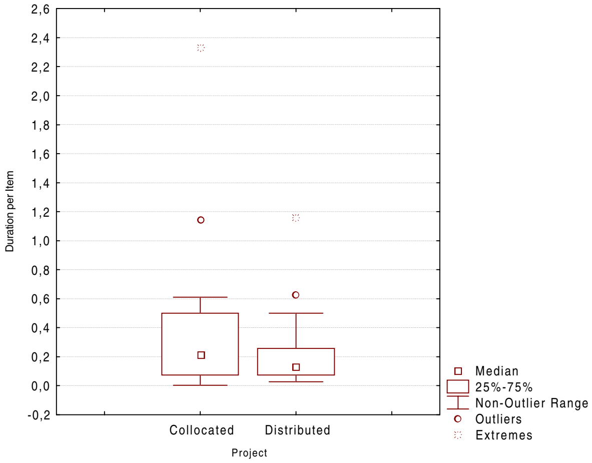

Figure 3 Boxplots of the duration of WPs normalized over items in collocated and distributed projects.

Figure 3 presents the distribution of WPs duration, normalized over the number of items, for both collocated and distributed projects. Boxplots graphically show some ordinal descriptive statistics, such as median, quartiles, and quartile range. It can be seen that, except of outliers and extreme values, the WPs duration is approximately the same in both cases. In fact, for the collocated WPs, the median is 0.213 and for the distributed WPS the median is 0.130, but the former presents outliers and extremes equals to 1.1430 and 2.333, respectively, while the latter presents outliers and extremes equals to 0.625 and 1.160, respectively.

The non parametric Mann-Whitney U test failed to reveal a significant difference between the two groups (p-level = 0.585).

3.2. Effort

Figure 4 shows the boxplots of the distribution of WPs effort, normalized over the number of items, for both collocated and distributed projects. For the collocated WPs, the median is 0.413 and for the distributed WPs the median is 0.278; the WPs in collocated case present an outlier which value is 1.917 and they have not any extreme; conversely, the WPs in distributed case does not present any outlier, but they have two extremes with value 1.387 and 1.516.

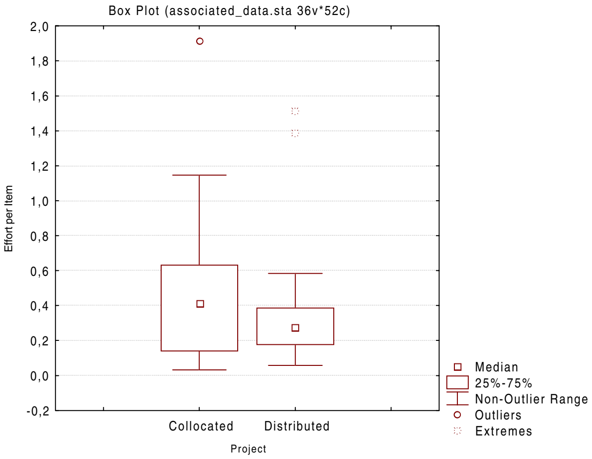

Figure 4 Boxplots of the effort of WPs normalized over items in collocated and distributed projects.

The non parametric Mann-Whitney U test failed to reveal a significant difference between the two groups (p-level = 0.441).

3.3. Staff

Figure 5 shows the boxplots of the distribution of WPs staff, normalized over the number of items, for both collocated and distributed projects. For the collocated WPs, the median is 0.043 and for the distributed WPS the median is 0.038; the WPs in collocated case present an outlier which value is 0.167 and an extreme with value 0.571; the WPs in distributed case present four outliers, which values range from 0.130 to 0.154, and two extremes with values 0.208 and 0.250, respectively.

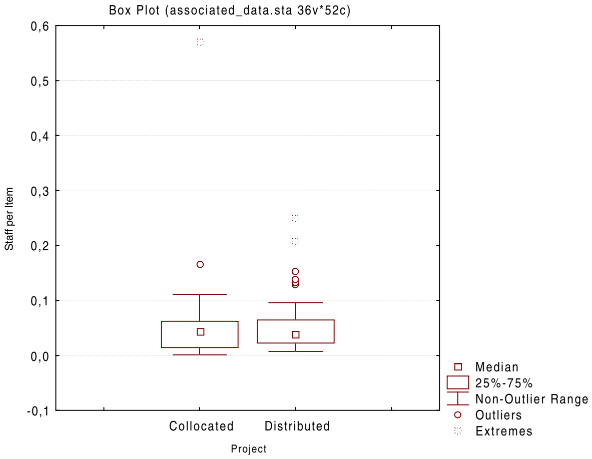

Figure 5 Boxplots of the staff required for WPs normalized over items in collocated and distributed projects.

The non parametric Mann-Whitney U test failed to reveal a significant difference between the two groups (p-level = 0. 930).

3.4. Rework Cycles

Figure 6 shows the boxplots of the distribution of WPs rework cycles, normalized over the number of items, for both collocated and distributed projects.

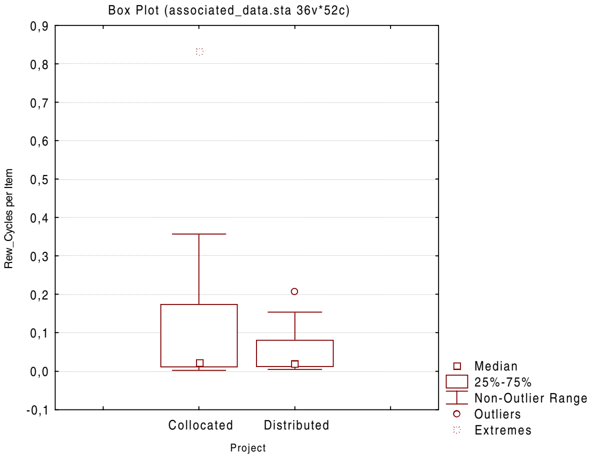

Figure 6 Boxplots of the rework cycles of WPs normalized over items in collocated and distributed projects.

For the collocated WPs, the median is 0.022 and for the distributed WPs the median is 0.020; the WPs in collocated case does not present any outlier and they have an extreme with value 0.833; conversely, the WPs in distributed case does not present any extreme, but they have an outlier with value 0.208.

The non parametric Mann-Whitney U test failed to reveal a significant difference between the two groups (p-level = 0. 654).

3.5. Number of Reports

Figure 7 shows the boxplots of the distribution of the number of produced reports in the WPs, normalized over the number of items of each WP, for both collocated and distributed projects. For the collocated WPs, the median is 0.059 and for the distributed WPS the median is 0.180; the WPs in collocated case does not present any outlier and they have two extremes with values 0.696 and 1.167, respectively; the WPs in distributed case present four outliers with values in the range from 1 to 1.5, and they do not present any extreme.

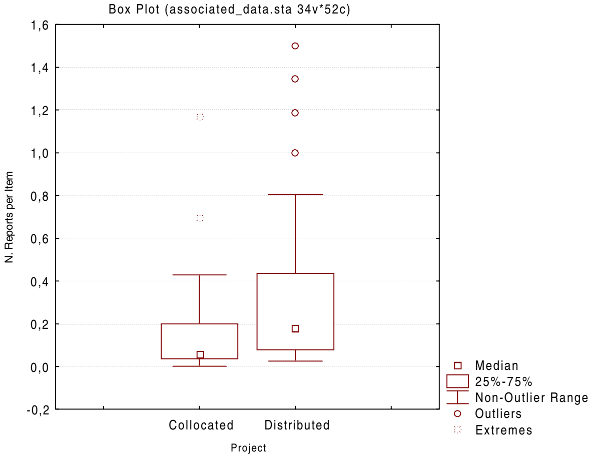

Figure 7 Boxplots of the number of reports produced during WPs normalized over items in collocated and distributed projects.

The non parametric Mann-Whitney U test revealed a significant difference between the two groups (p-level = 0.038).

3.6. Number of Meetings

Figure 8 shows the boxplots of the distribution of the number of held meetings within a WP execution normalized over the number of items of the WP, for both collocated and distributed projects. For the collocated WPs, the median is 0.045 and for the distributed WPS the median is 0.139; the WPs in collocated case present an outlier with value 0.7 and an extreme with value 1.167; conversely, the WPs in distributed case do not present any extreme and they have an outlier with value 1.292.

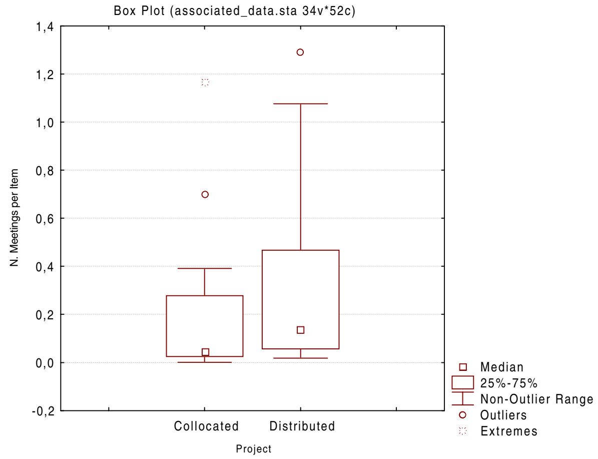

Figure 8 Boxplots of the number of meetings held during WPs normalized over items in collocated and distributed projects.

The non parametric Mann-Whitney U test revealed a significant difference between the two groups (p-level = 0.037).

4. Discussion

In general, the execution of the maintenance activities for WPs completion did not differ with respect to time, effort, rework, and staff whether the project was collocated or distributed over two sites. In the two cases, the observed differences were all not statistically significant at the conventional 0.05 *p* level.

These results can be explained if we consider how the entire maintenance activities were managed. Project management switched the execution of WPs from collocated to distributed, when there was a risk of schedule out of control. Then, the second site was used only in an emergency situation, after having spent some time and effort at the first site. Both in ordinary cases, when WPs were collocated, and in emergency cases, when WPs were distributed, their execution required all currently available resources.

This case study is characterized by three main factors:

- the task carried out: Y2K maintenance;
- the application domain: banking information systems;
- the homogeneity of sites: two national sites of the same company.

The specific task carried out in the maintenance activities concerned the Y2K problem in a large banking information system. This kind of task is conceptually simple and it is characterized by a massive and repetitive nature. The main skills required to execute the maintenance are the generic skills for the Y2K problem, and the specific skills for the application domain and the software system to maintain. Therefore, the choice of the most adequate maintenance team to assign a WP is straightforward, even when teams are geographically distant.

The majority of maintainers had a deep knowledge of both the application domain and the system, because of previous experience maintenance related to the same system. Moreover, all of them had been trained on the Y2K problem, and many maintainers had been already involved in other Y2K activities.

Finally, there was a strong organizational and cultural cohesion between the two sites because they were part of the same company and located in the same country, at a distance no more than 300 Km.

Nevertheless, the distribution of the work between the two sites caused an increase of the communication, expressed by both the number of formal reports produced by the teams and the number of meetings held during the project execution.

The significantly increasing number of reports can be explained by considering that all working groups produced reports about executed activities with the same frequency. Since the duplication of working sites led to a duplication of working groups it is reasonable to infer that the number of reports produced by distributed WPs is about twice the reports produced by collocated WPs.

Analogous considerations can be developed with respect to the higher number of meetings for distributed projects with respect to collocated ones. Within the company it is mandatory to hold periodical meetings among people within the same working team. Since the number of working teams is higher for the distributed execution of WPs, then the number of meetings is higher too.

This study shows that cycle time and cost can not being affected by geographical distribution. This finding contrasts with other reported studies [7, 8]. Differences in findings can be explained by the differences of contexts. In fact, in our study the following issues were easily faced:

- strategic problems: geographically distant sites belonged to the same company and were strongly linked;
- cultural problems: working groups were culturally homogeneous;
- technical problems: the communication network offered the desired reliability degree.

Moreover, since it was a massive maintenance project, the project components were loosely coupled and therefore the need to manage a common knowledge was kept to a minimum.

Indeed, with respect to communication and project management, some problem emerged, but they were successfully faced. In fact, despite of the greater effort spent for communicating and managing the projects, the overall effort does not significantly differs between the two projects. This can be explained by observing that in the distributed WPs execution, a skilled team can be chosen from a larger population; on the other hand, in the collocated WPs, if the required skills are not available, it is necessary to make use of inexperienced personnel. Since in the distributed WPs there is a greater availability of the required competences, they can be executed with a lower effort and duration.

In the project we analyzed, the effort and time saved in the maintenance execution compensated for the greater management effort, therefore the collocated and the distributed WPs had comparable costs.

5. Conclusions and Future Works

In this paper we investigated the effects of distributed work on process and project management performances for a large massive maintenance project. The results of our analysis show that the time and effort required for the collocated and distributed software processes are similar, when an adequate management of the strategic, cultural, and technical issues is possible. The distribution of the process over geographically distant teams makes it possible to include skilled people, wherever they are available, even though project management can be more burdensome. We found that the efficiency gain can compensate for the higher project management costs.

This study is one step towards a model of impact of geographical distance on critical factors of software development and evolution, which still needs further empirical investigation.

References

[1] E. Carmel, R. Agarwal, "Tactical Approaches for alleviating Distance in Global Software Development", *IEEE Software*, Mar-Apr 2001, pp. 22-29.

[2] A. Cockburn, "Selecting a Project's Methodology", *IEEE Software*, July-August 2000, pp.64-71.

[3] K. Nakamura, Y. Fujii, Y. Kiyokane, M. Nakamura, K. Hinenoya, Y.H. Peck, S. Choon-Lian, "Distributed and Concurrent Development Environment via Sharing Design Information", *Proc. of the 21st Intl. Computer Software and Applications Conference*, 1997.

[4] J. Suzuki, Y. Yamamoto, "Leveraging Distributed Software Development", *Computer*, Sep 1999, pp.59-65.

[5] C. Ebert, P. De Neve, "Surviving Global Software Development", *IEEE Software*, Mar-Apr 2001, pp.62-69.

[6] J.D. Herbsleb, D. Moitra, "Global Software Development", *IEEE Software*, Mar-Apr 2001, pp. 16-20.

[7] J.D. Herbsleb, A. Mockus, T.A. Finholt, R.E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed", *Proc. Intl. Conf. on Software Engineering*, 2001, pp. 81-90.

[8] C. Ebert, C.H. Parro, R. Suttels, H. Kolarczyk, "Improving Validation Activities in a Global Software Development", *Proc. Intl. Conf. on Software Engineering*, 2001, pp.545-554.

[9] A. Bianchi, D. Caivano, F. Lanubile, F. Rago, G. Visaggio, "Distributed and Colocated Projects: a Comparison", *Proc. of the IEEE Workshop on Empirical Studies of Software Maintenance*, 2001, pp. 65 – 69.

[10] W.J. Conover, *Practical Nonparametric Statistics*, John Wiley and Sons, 1980