Effects of Workspace on Engineering Software Projects

Faisal A. Abu Rub, Ayman A. Issa

***ABSTRACT*—In this pa[1]per, a combination of qualitative research methods is utilized to uncover new environmental factors affecting the management of software projects. Two new findings emerged from this study. First, workspace allocation directly affects resources productivity; and consequently, final deliverables of software projects. This effect could be positive, negative, and/or neutral nature. The second is the identification of sub-factors that control the nature of effect of this newly uncovered factor. Hence, a thorough evaluation for the project context should be undertaken to decide on the suitable workspace allocation strategy that will guide it to a safe end. Further work is being carried out to apply the same study on other environments so as to cross validate the results and generalize the conclusions of this research.**

***Index Terms*— Social Software Engineering, Workspace Allocation, and Project Team.**

I. INTRODUCTION

Since required software systems are increased and functionalities become complex, many attempts have been performed to improve software development. One of theses attempts is to improve how the software development team organize themselves and communicate each other. Software development process needs cooperation and collaboration between different levels of software development team [14]. Effective collaboration is important to produce large software systems and to deliver the software on time.

Many factors affect software development process and software engineers cooperation, one of these factors is workspace of software projects as workspace is where the software will be developed and produced. Hence, a better understanding of how to organize the workspace could ease and facilitate software development process. Workspace could be distributed or centralized. Every single or group of software development team in distributed workspace work in different location but in the centralized workspace, most of members of development team work in the same location. Optimization of the workspace to fit the nature of software development projects and facilitate collaboration between the team is a major challenge.

Although the software development team may belong to the same work location, software engineers may be separated in different buildings or in different floors of the same building. The effect of this factor on software engineers, and consequently, software projects is missing from relevant research studies. Also, current studies of workspace and communication effects on software projects consider the relationship between software engineers and their client within the framework of using a recent software process model such as prototyping and eXtreme Programming (XP). This neglects the importance of the internal communication level between software engineers themselves in the first place, which is highly affected by the surrounding workspace let alone the other technical factors such as team homogeneity and integrity.

This raises a number of research questions to which this paper aims to answer: (1) What form of effect does workspace allocation have on software projects?, (2) What factors control the effect of workspace allocation on software projects?, and (3) To what extent the effect of workspace allocation on software projects affect the project success?. This paper addresses these research questions using medium size software warehouse case study.

Section 2 summarizes the relevant literature work in social environmental studies on engineering software projects. The context of the field study and the details of the performed data collection and analysis activities of this research are presented in section 3. Consequently, empirical observation and findings are discussed in section 4. Finally, conclusion and future work are discussed in section 5.

II. SOCIAL ENVIRONMENTS IN SOFTWARE ENGINEERING

Since software is developed by humans, social environment is essential factor in software development. Social environment includes different cooperation, collaboration, and communication techniques among software engineers. According to Whitehead [15] software collaboration techniques have different goals such as establish the scope and capabilities of a project, dive convergence towards a final architecture and design, manage dependencies among activities, reduce dependencies among engineers, identify and resolve errors, and record organizational memory.

Software development process includes many activities and these depend on each other. Cooperation is defined by Malone and Crownston [10] as 'managing dependencies between activities'. Effective collaboration and communication could facilitate and ease managing of the activities and this reflects positively on software development.

A number of studies have been developed to support collaboration in software engineering. McChesney and Gallagher [11] investigated important factors that affect collaboration in software engineering such as separation of responsibilities across project roles, keeping people in the loop, programmer pairing, task allocation, formal team meetings, problem tracking reports, software tools, knowledge network, and queries. Other researchers focused

Faisal A. Abu Rub, Management Information Systems Department, Faculty of Administrative and Financial Sciences, University of Petra, faburub@uop.edu.jo
Ayman A. Issa, Software Engineering Department, Faculty of Information Technology, Philadelphia University, Jordan, aissa@philadelphia.edu.jo

on agile methods and communication theory in order to support social factors of software engineering [7].

Tools can play crucial role to support collaboration and communication between software engineers and to perform different activities of software development. Many tools and methods have been developed in order to achieve collaboration effectively. Arrow et al presented a theory of small groups as complex systems (SGACS) [1]. Gaffar et al [5] introduced mainstream pattern approach that encourages and guides both pattern writers and pattern users to "talk the same language" [5]. Tools can be used to support negotiation among stakeholders such as Boehm's and Egyed's [2] Win to Win approach. Other tools are presented by Maiden [9], Nguyen and Munson [12], and Maheshwari and Teoh [8].

On the client communication and personality characteristics level, Young et. al [16] demonstrated an explanatory study in nine systems development teams. The study was an integrated data assessment in order to work out stereotypical and specific personality characteristics appropriate to particular software development roles. As part of a larger study, an eXtreme Programming (XP) development team was investigated using a repertory grid approach. The repertory grid analysis of the different stakeholders identified significant personality characteristics (e.g. sharer/doer, organiser/leader) for different role types (e.g. XP team member, technical architect, operation manager, and client.). In addition, the concepts for describing good and bad team members were determined.

As shown above, there is a strong bonding relationship between successful delivery of operational software system and its both development and environmental contexts. A number of methods and techniques have been developed to better inform these factors in managing software projects. But, more efforts are required to investigate the effects of workspace on software projects, particularly, collaboration among physically adjacent and non-adjacent software engineers. Hence, organizing workspace to support software development raises the research questions, as detailed in section 1, in which this paper aims to investigate.

III FIELD STUDY

A. Environmental Context

This research was conducted on a medium size software warehouse [6], TestWarehouse, which includes 85 software engineers. The main office is located in California with two other regional offices in the Middle East. TestWarehouse is specialized in developing financial solutions in general, and loan servicing systems, in particular. The main unit of software development projects is a team. Each team consists of up to 18 resources of different roles: project manager(s), IT technical support officer(s), system and business analysts, developers, and quality assurance software engineers.

The adopted software development process differs from one project to another according to project context including project type, technical experience, application domain, delivery constraints, resources, and surrounding risks. However, the software development processes recently used in TestWarehouse are: prototyping, rapid application development, incremental development, and eXtreme Programming.

Projects differ according to three main distinguishing properties: technology, type, and size. In TestWarehouse, size is determined according to the number of estimated hours. Three size-based project categories are distinguished: small, medium, and large. Also, three main technologies are supported in TestWarehouse: .Net, Java, and Oracle. Project types are new development, customization and changes, and support projects. Also, projects may be in-house systems, outsourcing, custom development, or products.

Resource utilization is considered in this medium size organization to maximize the profit. Hence, it may be the case of having the same resource assigned on multiple projects with different percentages of involvement and roles. This adds more complexity on managing team homogeneity and workspace allocation.

Each office consists of 3 floors. Operations and resources manager holds the responsibility of allocating workspace for teams, departments, and individuals. Average meantime of resource residency in allocated workspace is 4 months according to the adopted policies and other factors such as project assignment and departmental issues. Hence, any group of physically adjacent resources may represent: (1) integrated project team, (2) members of one department, or (3) technically independent individuals.

This software warehouse is chosen here to be a foundation of the analysis and investigation of the effect(s) of workspace on software projects. It provides sufficient platform for the purposes of this research as it includes projects with both physically adjacent and non-adjacent team members in both centralized and distributed environments which could be used to evaluate the effects of all setup combinations on software projects.

B. Research Method

Research methodology is the approach of the research process [11]. This research adopted a hybrid positivistic (quantitative) and phenomenological (qualitative) approach to collect, analyze, and test data from multiple perspectives so as to support triangulation [11] and increase research validity, reliability and accuracy as detailed in sections 3.2.1 and 3.2.2, respectively.

B.1. Data Collection

In this research, the workspace allocation of five ongoing software projects is investigated. Table 1 summarizes the main characteristics of the investigated projects.

The main data collection method in this research is interviews. Interview is a phenomenological method that involves discussing issues with people. 49 software engineers have been interviewed during the period of three months to gain a detailed understanding of the problems caused by workspace and identify its pros and cons in the different projects contexts. Resources of different roles (i.e. project managers and developers) on different project types (i.e. new development and support) were interviewed to understand the effect of workspace allocation on software projects. The statistics of the interviewed resource roles on the different projects is presented in table 2. During the interview, the interviewee is asked 25 questions. A sample of these questions is shown in table 3. Each question is intended to inspect the relationship between a predefined sub-factor, as summarized in table 4, and the main factor being investigated in this research: workspace allocation.

On the other hand, the 36 non-interviewed resources were approached by sending them a copy from the prepared questionnaire (i.e. consisting of the same 25 questions used

Table 1: Characteristics of Investigated Projects.

Project	Size	Type	Technology	Physical Allocation
1	Medium	Product	ASP .net	Adjacent
2	Medium	Outsourcing	ASP .net	Non-adjacent
3	Medium	In house development	Java	Non-adjacent
4	Large	Support Project	Java	Adjacent
5	Large	Custom Development	ASP .net	Adjacent

Table 2: Interviewed Resources Statistics.

Project	Number of Interviewed Resources Per Role				
	Business Analyst	Project Manager	Developer	Tester	Designer
1	1	1	4	2	1
2	0	1	5	3	0
3	1	1	4	4	2
4	1	1	9	6	0
5	2	2	11	4	2

Table 3: Sample Interview Questions.

Number	Description
Q 3)	a) What is your current workspace structure? b) Does the current workspace structure help in performing your tasks? c) Specify How?
Q 8)	a) Do you think that workspace is purely allocated based on strategic planning ? b) How satisfied are you with it?
Q 15)	a) Does the workspace structure change when the project type change? b) What is the frequency of your workspace reallocation?

Table 4: Sub-factors of Workspace Allocation.

Sub-factor	Description
Project Type	Project could be new development project that should go through the whole software development life cycle, or support project which is already developed and subject to some modifications and maintenance.
Team Technical Background	This represents the role of each member of the project team (e.g. project manager, developer, tester, and system analyst).
Resources Personalities	This sub-factor reflects the cultural and social aspects of each resource.
Environmental Structure	This represents the physical location structure which could be open cubes or closed rooms/sections.
Managerial Political Strategies	This reflects project managers' strategic planning and management approaches.

in interviews) with full filling and understanding instructions. 25 of the surveyed resources returned completed questionnaires. This provided the researchers with better coverage of the organization and more comprehensive information that enriched the findings of the research as detailed in section 4. Observation is another phenomenological data collection method that was used in this research. The researchers dedicated three hours daily throughout the three months research period to observe team's interaction and communication in the different projects being investigated.

B.2. Data Analysis

The first step in analyzing data collected using the three methods detailed in section 3.2.1 was to categorize the information and use statistical techniques to extract embedded relationships between the major factor, workspace allocation, and sub-factors, summarized in table 4. Spearman's Correlation [4] used as an indicator of such relationships. Table 5 shows a sample from the resulted correlations. Among the very strange finding from these correlation factors is that the same sub-factor (e.g. Team Technical Background) has two contradictory correlations with the same major factor, workspace allocation, in the different projects. The outcomes of the observation sessions were utilized to explain this strange behaviour. They showed that the effect of any sub-factor is subject to the overall project context as will be discussed in section 4.

Table 5: Sample Spearman's Correlations.

Project	Sub-factor	Adjacent Workspace Allocation
Project 4	**Managerial Political Strategies**	0.120
	Team Technical Background	0.349
Project 5	**Managerial Political Strategies**	0.870
	Team Technical Background	0.783

IV. DISCUSSION OF EMPIRICAL OBSERVATIONS

Three main setups of workspace team allocation were observed and analyzed in this research: (1) all team members are physically adjacent in one workspace, (2) team members are distributed in different floors of the same centralized regional office, and (3) team members are distributed in different regional offices. Resources of the second and third workspace setups use different communication approaches to share and disseminate project related knowledge. This includes meetings, conference calls, emails, and drop in discussions.

Interviews and questionnaires responses have been statistically analyzed, as discussed in section 3.2.2, and showed that workspace could positively, negatively and/or neutrally affect software projects according to the situation of a number of related sub-factors: project type, managerial political strategies, team technical background, resources personalities, and environmental structure. The effects of the

different workspace allocation setups are discussed in sections 4.1 - 4.3.

A. Positive Effects

When Some projects need face to face discussion to achieve a set of software development tasks, physical adjacency of project team members promote better face to face instant communication compared to calls for meetings in distributed workspace environments. This also minimized overhead time (e.g. waiting, traveling, and follow up time) required to arrange and manage meetings. In addition, face to face communication adds new dimension of better social atmosphere among the project team, in particular, and organization staff, in general.

Another technical background related positive effect of per project team physical allocation represents the easier and more professional technical support between the team members when needed. In some highly secured organization, some resources may work together on a number of projects for a quite large number of years without even knowing each other. This may not be a good habit in terms of technical support as some technical problems need face to face discussions with respective technical resources to (1) better explain the context and consequences of the problem, and (2) learn from previous experiences. Furthermore, technical support is not only beneficial between the members of the team from the same background (i.e. developer to developer), rather inter-disciplinary (e.g. developers and testers, system analysts and developers) technical discussion may add another dimension to the diagnosing process that will lead to more robust and structured solution.

The last finding of positive effects of adjacent physical team allocation is a managerial one that is of direct relationship to project managers tasks. It was found that adjacent workspace facilitates the process of (1) collecting progress updates from the different team members, (2) sharing project managerial updates between team members, (3) monitoring and controlling project resources, and, most importantly, (4) managing resources communication in relation to project risks.

B. Negative Effects

Adjacent physical allocation of team members could affect negatively according to the findings of analysis of interviews and questionnaires responses. One of the most important issues of workspace and social atmospheres is resource cultural background and personalities. When people get together, the possibility of loud noise to emerge is increased. This downgrades resources concentration, and consequently, their productivity. Therefore, tasks are candidate to consume more time which directly push the project behind plans and schedules. Also, not all mentalities can match each other. Hence, tight coupling of people of contradictory mentalities and interests may directly affect their performance due to their low spirit.

Recent trends in software development are moving towards rapid development due to high competency and high dynamicity in recent technologies. In that, recent reports in the software industry showed that the average software development projects duration is 3-4 months. Also, resource utilization may result in having key resources participating in more than one project with 50% [13] involvement percentage or even less. So, maintaining a physical adjacency between team members in such cases is almost impossible or will surely lead to high resource instability resulting in considerable amount of time wasted moving team stuff around.

One last political factor that is in relation to resources physical adjacency and affects software projects negatively is power misuse. Few resources agreed that managerial decisions for team workspace restructuring are not always in favor for software projects. Rather, hidden motivations may be attributed to: (1) better utilize space, (2) separation of concerns, and (3) personal issues.

C. Neutral Effect

Support projects have a really small team with interrupted tasks of independent change and bug fixing natures. Hence, no team based activities such as analysis and design are required. Therefore, physical adjacency of such teams should not have any positive or negative input towards project deliverables and schedules.

Closed doors and black box management strategies do not allow information exchange or social activities between resources. Hence, in such organizations physical adjacency of team members will add nothing to the project.

Junior resources feedback showed that their limited experience in software development prevented them from critically evaluate the usefulness of one team allocation strategy over another.

Finally, these negative and neutral effects of the workspace allocation justify the contradictory correlations between it and the same sub-factor in the different projects as described in section 3.2.2.

V. CONCLUSION AND FUTURE WORK

In this research, a new factor affecting the management of software projects has been uncovered namely workspace allocation. The effect of workspace allocation on engineering software projects was investigated. In particular, the pros and cons of centralized versus distributed workspace allocation. A number of sub-factors influencing the usefulness of both types of workspace allocation were identified and evaluated using three qualitative research techniques: (1) interviews, (2), questionnaires and (3) observation. Examples of these sub-factors include: nature of project, team technical background, and managerial political strategies.

The findings of data analysis and evaluation showed that workspace allocation may have one of three effects on engineering software projects: positive, negative, or neutral according to the identified sub-factors and the project context. For example, it was shown that physical adjacency may have positive effect on new development projects but negative or neutral on support projects due to the absence of team-based activities such as functional analysis and system design. Figure 1 depicts the cause and effect diagram of the newly identified factor and its corresponding sub-factors on the successful delivery of software projects.

This has led us to conclude that the decision of workspace allocation of software project team cannot be blindly taken. A thorough evaluation of that specific context should be undertaken so as to drive the project to a safe end and successful delivery.

Future work is being carried out to conduct the same study in other software development environments. Also, the same study needs to be re-conducted in the same

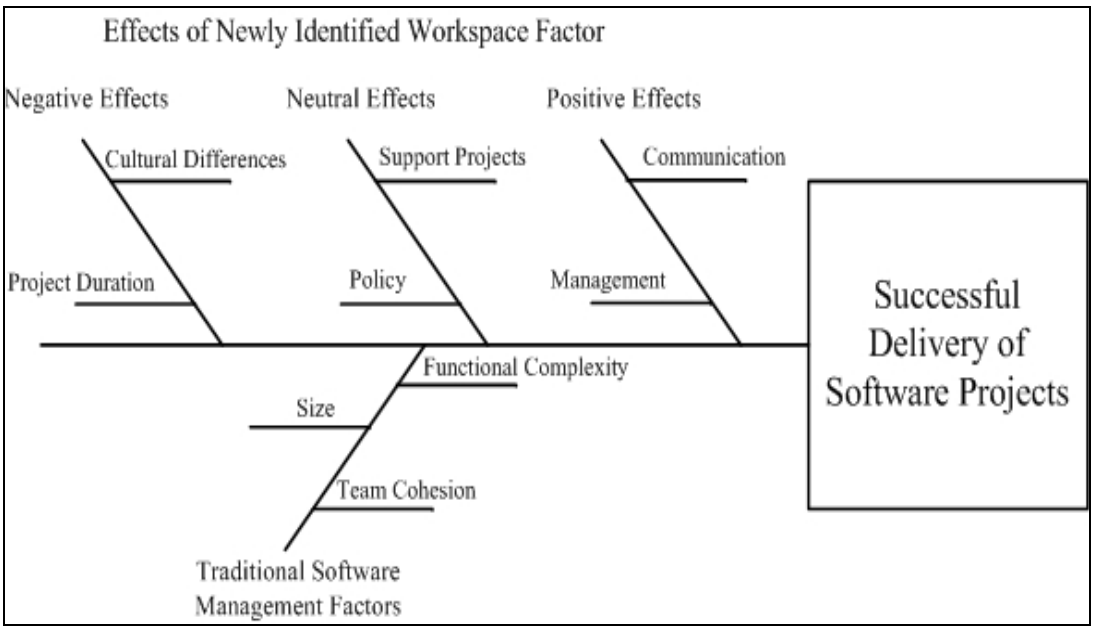

Figure 1: Cause and Effect Diagram of Workspace Allocation on Software Projects.

TestWarehouse environment in few years time to consider feedback of resources that were identified as junior and cross validate the results with the findings of this study.

Finally, the application of software measurements and metrics in the context of social software engineering is emerging as a research area that is anticipated to participate in better project control and monitoring [3,13]. Hence, the application of software metrics participation in (1) making the decision of workspace re-structuring and re-allocation, and (2) quantifying the effect of workspace sub-factors on the project costs is being considered as an extension for this research.

REFERENCES

[1] Arrow, H., McGrath, J., and Berdah, J., (2000). Small As Complex Systems: Formation, Coordination, Development and Adaption. CA: Sage: Thousand Oaks .

[2] Boehm, B. and Egyed, A., (1998). Software Requirements Negotiation: Some Lessons Learned. Information Systems Frontiers, 3 (2), pp. 333-347.

[3] Boehm, B. and Sullivan, K., (2000). Software Economics: a Roadmap in Proceedings of the conference on The future of Software engineering New York, NY, USA ACM Press, pp.319-343.

[4] Field, A., (2005). Discovering Statistics Using SPSS. 2nd edition. London: Sage.

[5] Gaffar, A., Seffah, A., Javahery, H., and Sinnig, D., (2003). MOUDIL: A Platform for Capturing and Sharing Patterns, Patterns I Practice Workshop in ACM CHI Conference, Florida

[6] Integrant Inc. (2003). Integrant Business Solutions Products [online]. US: Available from: http://www.integrantinc.com/products.aspx [Accessed 4/11/2004].

[7] John, M., Maurer, F., and Tessem, B., (2005). Human and Social Factors of Software Engineering- Workshop Summary ACM SIGSOFT Software Engineering Notes, 30 (4).

[8] Maheshwari, P. and Teoh, A., (2005). Supporting ATAM With a Collaborative Web-Based Software Architecture Evaluation Tool Science of Computer Programing, 57 (1), pp. 109-128.

[9] Maiden, M., (2004). Discovering Requirements With Scenarios: The ART-SCENE Solution CRM news, 58.

[10] Malone, T. and Crowstone, K., (1994). The Interdisplinary Study of Co-Ordination Computing Survey, 26 (1), pp. 87-119.

[11] McChesney, I. and Gallagher, S., (2003). Communication and Co-Ordination Practicies in Software Engineering Projects Information and Software Technology, 46 (1), pp. 473-489.

[12] Nguyen, T. N. and Munson, (2005). Object Oriented Configuartion Mnagement Technology Can Improve Software Architectural Traceability in third ACIS International Confrence on Softwar Engineering Research, Management and Applications Michigan, USA Mount Pleasant, pp.86-93.

[13] Reifer, D., (2004). Industry Software Cost, Quality and Productivity Benchmarks The DoD SoftwareTech News , 7 (2), pp. 3-19.

[14] Sommerville, I., (2001). Software Engineering. 6th ed. Harlow, England ; New York : Addison-Wesley.

[15] Whitehead, J., (2007). Collaboration in Software Engineering: A Roadmap [online]. CA, USA: University of California. Available from: http://www.cs.ucsc.edu/~ejw/papers/whitehead-future-of-collaboration-in-se.pdf [Accessed 1/5/2007].

[16] Young , S., Edwards, H., McDonald, S., and Thompson, J., (2005). Personality Characteristics in an XP Team: A Repertory Grid Study in Human and Social Factors of Software Engineering ACM, p.1.