Empirical Exploration of the Software Integration Success Factors in Global Software Development

Analyses based on Company Size and Practitioners' Experiences

Muhammad Ilyas
Software Engineering Research Group (SERG_UOM),
Department of Computer Science & IT,
University of Malakand, KPK, Pakistan
e-mail: milyasmkd@uom.edu.pk

Siffat Ullah Khan
Software Engineering Research Group (SERG_UOM),
Department of Computer Science & IT,
University of Malakand, KPK, Pakistan
e-mail: siffatullah@uom.edu.pk

***Abstract* — Software Integration is the most important and complicated phase of software development process. The integration phase becomes even more challenging in Global Software Development (GSD) environment. In our previous study, we identified nine Critical Success Factors (CSFs) using Systematic Literature Review (SLR). Further, for validation of the identified CSFs and for identification of additional success factors, we conducted an industrial survey in GSD environment. In this paper, we present some important analyses of the identified software integration CSFs in GSD environment, based on practitioners' experiences and company size, through industrial survey.**

Keywords-Software Integration; Success Factors; Empirical Study; Global Software Development.

I. INTRODUCTION

The advances in Information and Communication Technologies (ICTs) have resulted in an increase in software use and its size. The software development process has also changed from local to global software development [1]. Global software development paradigm has been adopted by many software vendors, from the last two decades, because of the perceived benefits that can be gained from GSD [1] e.g., cost savings, reduced time to market, proximity to market and customers' access to large skilled labor force, etc. However, in spite of the benefits gained from GSD, vendors also face communication, coordination, knowledge sharing and control problems due to temporal, cultural and linguistic differences [2]-[5]. These problems have also made software integration process more complicated [6]-[8]. Many of the uncovered problems of the previous phases start appearing in the integration phase [9]. These problems not only increase the workload of the global teams but also decrease the quality of the final working product. Researchers reported that more than 50% of the software development projects suffer from cost overrun and/or time overrun problem(s) due to the complexities and incompatibilities found at the software integration stage [10]. Keeping in mind the importance of the integration stage, we proposed the following research questions.

RQ-1: What are the critical success factors (CSFs), as identified in the literature and real-world practice, to be adopted by GSD vendors at various stages of the product integration in GSD environment?

RQ-2: Do the identified critical success factors, as identified in the survey, vary with the level of experience?

RQ-3: Do the identified critical success factors, as identified in the survey, vary with the organization size?

In order to answer RQ1, we identified a list of nine Critical Success Factors (CSFs), as shown in Table 1, in our previous study using Systematic Literature Review (SLR) method. To answer RQ1, Table 1 shows a list of nine Critical Success Factors (CSFs) identified in our previous study, via Systematic Literature Review (SLR) method [11]-[13]. These findings were further validated through a questionnaire survey in the industry. In this paper, we present analyses of the empirical data regarding the identified CSFs based on different variables such as expert's level of experience and organization's size. Thus, we have tried to answer RQ2 and RQ3 in this paper, whereas RQ1 has already been published [11][13].

TABLE 1. LIST OF SOFTWARE INTEGRATION CSFS

S.No	Software integration Critical Success Factor (CSFs)
1	Consistency in requirements and architecture design
2	Intra and inter team communication and coordination
3	Component/unit testing prior to integration
4	Advance and uniform development environment and training
5	Efficient incremental/continues integration
6	Efficient specification for interface compatibility
7	Proper documentation & configuration management
8	Early integration planning and centralized P3 management
9	Careful evaluation of the Commercial Off-The-Shelf/Open Source Software (COTS/OSS) components

The remaining of the paper is organized as follows. In Section 2, background and motivation is presented, while the research methodology is presented in Section 3. Results are presented in Section 4. Finally, Section 5 discusses the limitations of the study and Section 6 presents the conclusion and future work.

II. BACKGROUND

Lorson [9] defines the integration process as a set of procedures for combining components into one larger component, product, subsystem or system. It is the integration stage that enables the organization to assess the overall system functionality and performance that a system may have. In software systems, the integration is the first stage where the overall results of the software development efforts can be observed. Thus integration is a critical phase in the overall software development process.

Paloheimo [14] reported that "as the integration phase is usually the last to follow in a software development process, the unnoticed problems in the preceding phases tend to accumulate in this final phase". The author recommended joint/shared milestones, and incremental integration for successful integration of the software components in the GSD environment.

Van Moll et al. [10] report that the majority of the GSD projects suffer because of the integration complexities. The authors of the study recommended good planning, better monitoring and control, and assigning responsibilities to each and every team member in a well defined manner.

Vasilescu et al. [15] have quantitatively analyzed the continuous integration practice of software engineering. They have concluded that the success or failure of a build process is dependent on the way the code is modified. The code can be modified in two ways:

- Direct change in the code: In this case, a small group of developers, who have the write access to the main project repository, modifies the code.
- Indirect/pull request: In this case, developers who fork the main repository, change their copies locally and tender pull request for review and merge.

Their analysis showed that pull request method of code change is more likely to cause integration testing failures as compared to the direct method. The main limitation of their results is their applicability to open source projects only.

Adams et al. [16] reported in an empirical study that, although the reuse of COTS/OSS components is the best practice, the integration process of these components may also introduce unexpected maintenance costs. They pointed out a need of increased empirical research in software engineering for successful reuse and integration of COTS/OSS software components [11][13].

III. RESEARCH METHODOLOGY

The empirical methods such as case studies, controlled experiments, surveys and post-mortem analysis are essential to the researchers for evaluation and validation of research results in the field of software engineering because the software development process is human intensive work [17]-[19]. In survey design, a survey or questionnaire is administered to a small group of people, also called the *sample*, for identification of trends in characteristics, opinions, attitudes or behavior of a large group of people, also called *population* [20]. Interview and questionnaire are the two main methods of gathering the quantitative or qualitative data. In both methods, a sample representing a population is studied. The results obtained from the survey are analyzed for derivation of explanatory and descriptive conclusions. These conclusions are then generalized to the population from which the sample was taken and studied [17]. In view of the available resources and diverse range of respondents, we have used the questionnaire method as the data collection tool.

The purpose of conducting the survey was to validate the findings of the SLR through industry practitioners and to identify new practices, if any. A similar approach has been used by other researchers [5][6][18].

We have designed the questionnaire survey based on the inputs from our previously published SLR study [21]. The questionnaire survey was properly conducted as done by other researchers [5][22]. We have used both open and close ended questions in this survey. The close ended questions were used as an instrument for collecting self-reported data. In our case, we have used the close ended questions for collecting data about the software integration success factors identified through SLR. We have also used open ended questions to gain the tacit knowledge on the success factors from the industry experts.

The questionnaire used in the survey was designed for eliciting the significance that each respondent has placed on each software integration success factors as identified through SLR. In order to expose the importance of each factor, we have used a seven point Likert scale i.e., Extremely Agree, Moderately Agree, Slightly Agree, Not Sure, Slightly Disagree, Moderately Disagree and Extremely Disagree. The respondents were requested to mention each practice relative value. We used a 7 point Likert scale in the survey, however, for the analysis purposes mentioned in this paper, we have considered Extremely Agree (EA) view point of the survey participants. The number of responses got for the other 6 view points were very low and are therefore not analyzed in this paper.

IV. RESULTS

This section discusses the results and examines the identified software integration critical success factors for each of the Research Questions stated in Section I.

RQ-2 Do the identified critical success factors, as identified in the survey, vary with the level of experience?

We received consent from 232 experts for participation in the survey. A total of 99 experts participated in the survey from 22 different countries. We received a total of 96 valid responses from participants of the questionnaire survey and have used seven point Likert scale (EA: Extremely Agree, MA: Moderately Agree, SA: Slightly Agree, NS: Not Sure, SD: Slightly Disagree, MD: Moderately Disagree and ED: Extremely Disagree). In order to answer RQ-1, we classified the survey participants into three groups, as shown in Table 2, based on their experience level, as follows:

- Junior level experts (JLE): 1 to 5 years experience
- Intermediate level experts (ILE): 5+ to 10 years experience
- Senior level experts (SLE): 10+ experience

It should be noted that these three classes of experts were defined after discussion with the industry experts and external reviewers. Other researchers may however define their own criteria for deciding different levels for experts.

TABLE 2. SUCCESS FACTORS, EXTREMELY AGREE VIEW POINT OF EXPERTS HAVING DIFFERENT EXPERIENCE LEVELS

Critical Success Factors (CSFs)	**Expert's experience level**			**Chi Square Test (Linear-by-linear Association ∝=0.05, Df =1)**	
	Junior (1 to 5y) (n=39)	***Intermediate (5+ to 10 y) (n=26)***	***Senior (10+ y) (n=31)***		
	% of EA	***% of EA***	***% of EA***	X^2	***P***
CSF1-Consistency in requirements and architecture design	72	81	87	2.462	0.117
CSF2-Intra and inter team communication and coordination	74	58	77	3.897	0.048
CSF3-Component/Unit testing prior to integration	54	65	55	0.20	0.888
CSF4-Advance & uniform development environment and training	38	54	45	0.385	0.535
CSF5-Efficient incremental/continuous integration	38	46	48	0.710	0.399
CSF6-Efficient specification for interface compatibility	31	46	39	0.548	0.459
CSF7-Proper documentation & configuration management	44	42	45	0.014	0.904
CSF8-Early integration planning and centralized P3 management	23	27	35	1.275	0.259
CSF9-Careful evaluation of the COTS/OTS components	51	54	55	0.090	0.765

The data in Table 2 shows that all CSFs excluding CSF8 "Early integration planning and centralized P3 management" have been cited by >=30% in the sample of extremely agree responses from the three levels of experts. The most common success factors which have >=50% of extremely agree responses in the sample, across all three level of experts are CSF1-"Consistency in requirements and architecture design", CSF2-"Intra and inter team communication and coordination", CSF3-"Component/Unit testing prior to integration" and CSF9-"Careful evaluation of the COTS/OTS components". It is worth mentioning that the factor "Consistency in requirements and architecture design" is the top ranked factor for all three experience levels of experts. Therefore, proper care should be taken at the design time of software architecture and gathering and specification of requirements because consistent software architecture is positively correlated with the ease of the integration process [23]. Similarly, Kommeren et al. [24] suggested that, for achieving a unified interpretation of requirements, they should be discussed repeatedly with all the development teams. This will result in an optimal design of software components that can be easily integrated. On the other hand, any deficiency in the common understanding of requirements may yield poor design decisions leading to delay in the integration process and the project as a whole.

RQ-3: Do the identified critical success factors, as identified in the survey, vary with the organization size?

According to the Australian Bureau of Statistics [25] definition of organization size, we divided the questionnaires on the basis of organization size into three groups as follows:

- Small (<20 employees)
- Medium (20 – 199 employees)
- Large (>=200 employees)

In order to answer RQ-3, the distribution of the success factors reported by various groups of experts, in the survey, from the three size of organization, is presented in Table 3.

The data in Table 3 shows that all success factors have cited as extremely agree across various groups of experts in all three types of organizations. It should also be noted that all CSFs have been reported with >=30% by experts of all three size organizations except CSF8-"Early integration planning and centralized P3 management", which has 20% occurrence in the large size organization. The reason of low frequency for CSF8 may be that large organizations may have already implemented better planning and management for the activities related to software integration. Again, there are some factors which have got >=50% in the extremely agree response sample in two or more than two types of organizations. These factors are CSF1-"Consistency in requirements and architecture design", CSF2-"Intra and inter team communication and coordination", CSF3-"Component/Unit testing prior to integration", CSF4-"Advance & uniform development environment and training", CSF7-"Proper documentation & configuration management" and CSF9-"Careful evaluation of the COTS/OTS components".

It should be noted that the CSF1-"Consistency in requirements and architecture design" and CSF2-"Intra and inter team communication and coordination" are the two top most ranked factors which have got >=50% across the experts of all the three size of organizations i.e., small, medium and large. Further, Chi Square Test shows that there is no significant difference because no column has p<0.05. Hence, it is obvious that these factors should be implemented on priority basis in organizations of all sizes.

TABLE 3. SUCCESS FACTORS, EXTREMELY AGREE VIEW POINT OF EXPERTS ACROSS VARIOUS SIZE OF ORGANIZATIONS

Critical Success Factors (CSFs)	**Company size**			**Chi Square Test (Linear-by-linear Association ∝=0.05, Df =1)**	
	Small (n=16)	***Medium (n=36)***	***Large (n=44)***		
	% of EA	***% of EA***	***% of EA***	***X^2***	***P***
CSF1-Consistency in requirements and architecture design	63	89	77	0.389	0.533
CSF2-Intra and inter team communication and coordination	75	83	59	3.145	0.076
CSF3-Component/Unit testing prior to integration	63	67	48	1.983	0.159
CSF4-Advance & uniform development environment and training	50	53	36	1.592	0.207
CSF5-Efficient incremental/continuous integration	56	44	39	1.401	0.237
CSF6-Efficient specification for interface compatibility	38	44	32	0.009	0.926
CSF7-Proper documentation & configuration management	63	50	32	2.245	0.234
CSF8-Early integration planning and centralized P3 management	44	31	20	3.260	0.071
CSF9-Careful evaluation of the COTS/OTS components	56	67	41	2.646	0.104

V. Limitations

The data presented for analysis in this paper was obtained by conducting a questionnaire survey in the GSD industry. A general problem with the survey is that it has a very low response rate and has the possibility of subjective biasness. The results of the survey exhibit opinions of the respondents about a phenomenon under investigation. Literature reveals that the opinions obtained through a survey may be biased as well as different from the real population distribution [26]. In our study, we have tried to explore the perceptions and experiences of GSD experts, but it was not possible to verify these perceptions and experiences directly. Moreover, practitioner's opinions and perceptions may not be accurate. Additionally, the respondents of the questionnaire survey were self-selecting. However, the results of piloting studies give a satisfactory level of internal validity since the variables incorporated in this research study were obtained from comprehensive literature review and piloting of the questions survey. The external validity is addressed by receiving survey responses from a total of 96 experts, among which 56 experts belong to 22 different countries, providing a good representative sample.

VI. Conclusion And Future Work

The analyses presented in this paper show that our identified software integration practices are important from various experts point of view. This means that implementation of these success factors may help GSD vendors to easily and effectively integrate their software components. The frequency percentages of each CSF in the questionnaire survey (EA: extremely agree) show the relative importance of each factor within the group of software integration success factors. The implementation of software integration CSFs, especially those reported with greater percentage, may boost the performance of GSD vendors by effectively integrating their software components.

Further analysis of the CSFs based on different variables, such as expert's position, time etc. is reserved for future work.

The ultimate aim of this research work is to develop our proposed Software Integration Model (SIM) for GSD vendors [27].

References

[1] P. J. Agerfalk, B. Fitzgerald, H. H. Olsson and E. O. Conchuir, "Benefits of global software development: the known and unknown," in *Making Globally Distributed Software Development a Success Story*, ed: Springer, 2008, pp. 1-9.

[2] R. A. Khan, S. U. Khan and M. Niazi, "Communication and Coordination Challenges Mitigation in Offshore Software Development Outsourcing Relationships: Findings from Systematic Literature Review," in *The Tenth International Conference on Software Engineering Advances (ICSEA 2015)*, Barcelona, Spain, 2015, pp. 45-51.

[3] M. Jimenez, M. Piattini and A. Vizcaino, "Challenges and improvements in distributed software development: A systematic review," *Advances in Software Engineering*, vol. 2009, pp. 1-14, 2009.

[4] P. J. Agerfalk *et al.*, "A framework for considering opportunities and threats in distributed software development," in *Proceedings of the International Workshop on Distributed Software Development*, Paris, 29, 2005, pp. 47-61.

[5] A. W. Khan and S. U. Khan, "Solutions for Critical Challenges in Offshore Software Outsourcing Contract," *Pakistan Academy of Sciences*, vol. 52, pp. 331-344, 2015.

[6] M. Niazi, S. Mahmood, M. Alshayeb and A. Hroub, "Empirical investigation of the challenges of the existing tools used in global software development projects," *IET Software*, vol. 9, pp. 135-143, 2015.

[7] M. Ilyas and S. U. Khan, "Software integration in global software development: Challenges for GSD vendors," *Journal of Software: Evolution and Process*, vol. 29, pp. 1-17, 2017.

[8] M. Ilyas and S. U. Khan, "Software integration challenges for GSD Vendors: An exploratory study using a systematic literature review," *Journal of Computers*, vol. 12, pp. 416-422, 2017.

[9] S. Larsson, "Key Elements of the Product Integration Process," Ph.D. Thesis Proposal, Department of Computer science and Electronics, Malardalen University Sweden, Malardalen, 2007.

[10] J. Van Moll and R. Ammerlaan, "Identifying Pitfalls of System Integration-An Exploratory Study," in *IEEE International Conference on Software Testing Verification and Validation Workshop (ICSTW)*, Lillehammer, 2008, pp. 331-338.

[11] M. Ilyas and S. U. Khan, "Software Integration in Global Software Development: Success Factors for GSD vendors," in *16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2015)*, Takamatsu, Japan, 2015, pp. 119-124.

[12] M. Ilyas and S. U. Khan, "Software Integration Challenges in Global Software Development Environment: A Systematic Literature Review Protocol," *IOSR Journal of Computer Engineering (IOSRJCE)*, vol. 1, pp. 29-38, 2012.

[13] M. Ilyas and S. U. Khan, "An Exploratory Study of Success Factors in Software Integration for GSD Vendors " *Proceedings of the Pakistan Academy of Sciences*, vol. 53, pp. 239–253 (2016), 2016.

[14] H. Paloheimo, "Feasibility of SW Architecture Integration in a Distributed R&D Environment," HUT / SoberIT 2003, Fall T-76.6512003.

[15] B. Vasilescu, S. Van Schuylenburg, J. Wulms, A. Serebrenik and M. G. van den Brand, "Continuous integration in a social-coding world: Empirical evidence from GitHub," in *International Conference on Software Maintenance and Evolution (ICSME)*, BC, Canada, 2014, pp. 401-405.

[16] B. Adams, R. Kavanagh, A. E. Hassan and D. M. German, "An empirical study of integration activities in distributions of open source software," *Empirical Software Engineering*, pp. 1-42, 2015.

[17] C. Wohlin, M. Höst and K. Henningsson, "Empirical research methods in software engineering," in *Empirical methods and studies in software engineering*, ed: Springer, 2003, pp. 7-23.
[18] T. Ambreen, N. Ikram, M. Usman and M. Niazi, "Empirical research in requirements engineering: trends and opportunities," *Requirements Engineering*, vol. 21, pp. 1-33, 2016.
[19] M. Alshayeb, "Empirical investigation of refactoring effect on software quality," *Information and software technology*, vol. 51, pp. 1319-1326, 2009.
[20] J. W. Creswell, *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*, 4th ed.: Prentice Hall, 2008.
[21] M. Ilyas and S. U. Khan, "Practices for Software Integration Success Factors in GSD Environment," in *15th International Conference on Computer and Information Science (ICIS), 2016 IEEE/ACIS*, Okayama, Japan, 2016, pp. 1-6.
[22] M. Niazi, D. Wilson and D. Zowghi, "A Framework for Assisting the Design of Effective Software Process Improvement Implementation Strategies," *Journal of Systems and Software,* vol. 78, pp. 204-222, 2004.
[23] R. Land and I. Crnkovic, "Software systems in-house integration: Architecture, process practices, and strategy selection," *Information and Software Technology (IST)*, vol. 49, pp. 419-444, 2006.
[24] R. Kommeren and P. I. Parviainen, "Philips experiences in global distributed software development," *Empirical Software Engineering,* vol. 12, pp. 647-660, 2007.
[25] D. Trewin, "Small Business in Australia," Australian Bureau of Statistics, Canberra 2001.
[26] B. A. Kitchenham *et al.*, "Preliminary guidelines for empirical research in software engineering," *IEEE Transactions on software engineering,* vol. 28, pp. 721-734, 2002.
[27] M. Ilyas and S. U. Khan, "Software Integration Model for Global Software Development," in *15th International Multitopic Conference (INMIC)*, Islamabad, Pakistan, 2012, pp. 452-457.

 ISBN: 978-1-61208-666-8