Software Integration Challenges for GSD Vendors: An Exploratory Study Using a Systematic Literature Review

Muhammad Ilyas*, Siffat Ullah Khan
Software Engineering Research Group (SERG_UOM), Department of Computer Science & IT, University of Malakand, KPK, Pakistan.

* Corresponding author. Tel +92 333 9506261; email: milyasmkd@gmail.com
Manuscript submitted December 12, 2015; accepted May 10, 2016.
doi: 10.17706/jcp.12.5.416-422

Abstract: The trend of software development has changed from local to global software development (GSD) due to advances in information and communication technologies (ICTs). Despite of the benefits gained from GSD, vendors also face challenges during the integration of components developed by global teams in isolation. The objective of the current study is to find out the list of critical challenges/barriers (CBs) that obstruct the integration process at any stage. To achieve the objective we conducted a systematic literature review (SLR) and extracted the data from 88 papers found in six digital libraries. A total of 16 barriers were found among which 10 barriers are ranked as CBs. Some of the top ranked barriers are "Lack of Communication", "Lack of Proper Documentation", "Lack of Compatibility" and "Architecture Mismatch".

Key words: Global software development, systematic literature review, software integration, barriers, challenges.

1. Introduction

The advances in information and communication technologies have changed the world in a global village. This has also changed the trend of software development from local to global software development (GSD). This revolution has some good reasons like reducing the cost, availing latest technology and experts, exploitation of 24-hours working day, showing presence in the global market, innovation and shared best practice [1]. In GSD not only the clients and vendors but the development team may itself be scattered at different geographical locations of the globe [2]. This geographical dispersion has also raised challenges like communication and coordination challenges, cultural and language differences, time zone differences, poor contract management and knowledge management [3]-[5]. These issues have also led to the technical and complicated issue of software components integration [6], [7].

The integration of software components is the most challenging and attention demanding phase in GSD environment [8], [9]. Integration is a vital phase in all types of software projects, whether the component are developed in-house or outsourced/purchased from the market as off the shelf (OTS) component [10]. It is important to explore what barriers stand in the way of successful integration and what factors ease and speed the integration process. Schneider *et al*. [7] point out that "areas like product integration have received surprisingly little attention". Similarly Tekumalla *et al.* [6] state that "regarding the research on Integration of Components, we observed that the number of industrial case studies is not adequate enough to cover various issues associated with it".

Despite the criticality of integration phase, it has got very little attention of the research community in GSD

environment. Little empirical research has been done on integration practices/solutions in general and for identifications of integration challenges in particular [7].

The main aim of our research work is to develop a software integration model (SIM) [11] to assist GSD vendors in successful and effective integration of software components/products. In order to bridge the gap and to ease the integration process in GSD environment, we have formulated the following research questions:

RQ1: What are the success factors, as identified in the literature, to be adopted by GSD vendors at various stages of the product integration, i.e. before, during and after the integration process in GSD environment?

RQ2: What are the challenges, as identified in the literature, faced by GSD vendors in product integration based on project size and product type?

RQ2-a Do the identified challenges vary from decade to decade?

The findings of RQ1 has been published in SNPD 2015 conference [12] while in this paper we have reported the findings of RQ2 supplemented by RQ2-a.

2. Background

Software integration means to assemble the separately developed components/modules into a bigger unit/subsystem or final working product/system. Many faults that remain hidden in the early stages appear during the integration stage or even worse in the verification/validation stages [13]. Thus it is very important to investigate the area of software integration to improve the overall software development process [14].

The literature shows that integration has not got enough attention of the practitioners and researchers [7], and the developers face the consequences in the form of challenges and barriers in the later stage during integration, delaying the overall project, increasing the cost while decreasing the quality of the final product [15]. The matter becomes more complicated in GSD environment where the teams develop software units independently without having the bigger picture in mind when these units will be integrated into one unit. Thus before embarking into the development of the software components/modules it is necessary to properly plan the integration strategy in advance [16].

Cataldo *et al.* [17] in an empirical analysis have found that the main failure in product integration is the cross-feature interaction. The cross-feature interaction is a measure of architecture dependencies between two products features. The main limitation of this case study is that it was performed on a single system developed by only one organization.

Tekumalla *et al.* [6] have conducted a SLR and concluded that certain areas of component based software engineering (CBSD), especially integration and testing, need to be investigated through industrial case studies and experiments. The numbers of industrial case studies are not sufficient and the issues design trade-off in integration of components and interaction and compatibility between components need to be thoroughly investigated.

Our analysis of literature shows that none of the studies have performed in-depth analysis of software integration challenges and their solutions. The purpose of our research study is to bridge this gap and to assist GSD vendors in effective integration of the components into a final product.

3. Study Design

We have conducted a comprehensive systematic literature review (SLR) [18] to obtain the results of the study. A SLR is used for the identification, evaluation and interpretation of all primary studies relevant to a specific research question or topic. It is a type of secondary study and up to some extent repeatable because it utilizes a well-defined methodology [18], [19]. There are three main phases of the SLR process namely,

planning the review, conducting the review and reporting the review. The planning phase of the SLR is the development of the SLR protocol, which for this study was written, reviewed and has been published [20].

3.1. Search Strategy and Search

The data in Table 1 list the libraries that we searched and the number of papers found in each library. The following search string has been used for searching through these libraries:

Search String: (("Software integration" OR "Product integration" OR "Component integration" OR "system integration") AND ("Global software development" OR GSD OR "Global software engineering" OR GSE OR "Distributed software development" OR DSD OR "Distributed software engineering" OR DSE) AND (Challenge OR risk OR problem OR issue OR barrier OR trouble OR "critical factor" OR "key factor" OR "success factor")).

Some search engines (Google scholar etc) recognize only 256 characters of the search string. For this reason we have divided the search string into four sub strings whose details can be found in the SLR protocol published in [20].

Table 1. Search Results

S. No	Library Name	No. of publications found	Initial selection	Final selection
1	Science Direct	185	13	07
2	ACM	195	23	11
3	Springer Link	205	15	05
4	IEEE eXplore	417	34	16
5	Google scholar	1031	163	88
6	Wiley online library	81	08	02
7	Using **Snow Balling** technique	80	80	43
Total Papers		2194	336	172
Duplicate Papers				067
Net Total= (Total – Duplicate)				105

3.2. Inclusion/Exclusion Criteria and Quality Assessment

The publication selection in SLR process is mainly based on inclusion/exclusion criteria and quality assessment of the publications, which for the current study is available in our SLR protocol [20].

3.3. Selecting Primary Studies

The papers selection for data extraction was performed in two steps. Firstly the papers were selected by reading the title and abstract of the paper and this step resulted in 256 papers. Secondly we read the full paper for final selection and got 63 papers as finally selected. After that we also performed the snowballing technique [21], [22]. Thus our sample size was increased from 67 to 88 papers for data extraction and identification of integration challenges.

3.4. Data Synthesis

Using the SLR guidelines we grouped similar factors and obtained a list of 16 challenges as shown in Table 2 among which 10 barriers were ranked as critical barriers as shown by bold type fonts.

4. Results

In this section, we discuss the results and analyze the identified barriers to answer RQ2 and RQ2-a, as mentioned in Section 1. The details are given in the following subsections.

4.1. Barriers/Challenges for Software Product Integration

To answer RQ2, we identified a list of 16 challenges/barriers, through SLR, as presented in Table 2. Among these challenges, 10 challenges are decisive or critical challenges/barriers (CBs) shown in bold

fonts. The criterion for deciding a barrier to be critical is based on the percentage with which it has appeared in the sample of finally selected papers. If a barrier has got a frequency percentage >=30%, we consider it to be a critical one. Other researchers have also used the similar approach [23]-[25]. However, GSD practitioners may define their own criteria for deciding the significance of a barrier to be critical or not.

Table 2. List of Integration Challenges/Barriers

S. No	Barriers	Freq	%(*n*=88)
1	**Lack of Communication**	36	41
2	**Lack of Proper Documentation**	34	39
3	**Lack of Compatibility**	32	36
4	**Architecture Mismatch**	31	35
5	**Lack of Integration Planning and lack of Management**	29	33
6	**Heterogeneous Development Environment and Platforms**	28	32
7	**Improper/No unit testing**	28	32
8	**Wrong OTS Product selection and customization**	27	31
9	**Lack of Resources, Knowledge and Skills**	26	30
10	**Lack of proper Component Interfaces**	26	30
11	Unclear responsibilities	13	15
12	Configuration and Versioning complexity	13	15
13	Lack of common understanding of requirements	11	13
14	Timely availability of Components	7	8
15	Big bang integration	5	6
16	Lack of common development process	5	6

4.2. Decade Wise Comparison of the Success Factors

To answer RQ2-a, we have divided the search period in two decades i.e. from 1994 to 2003 and from 2004 to 2013. It should be made explicit that we have put no date boundaries on our search process but we did not find any relevant paper before 1994. The number of publications found in each decade is presented in Table 3.

A linear by linear association Chi-square test has been used to find the significance difference in the barriers, if any, between the two decades. The linear by linear association Chi-square test is considered more powerful than Pearson ' s χ 2 test [26].

The data in Table 3shows that integration has got the attention of researchers more in the second decade than in the first one with the revolution of GSD and its increase effect on the integration process and its associated challenges. The wide spread use of GSD process has brought new barriers and challenges associated with integrating the components developed by these GSD teams. A comparisons of each barrier based on the two decades is presented in Table 4.

Table 3. Decade Wise Breakup of Publications

Decade	Frequency	Percentage
1994-2003	21	24%
2004-2013	67	76%

The frequency percentage of each barrier in the corresponding decade shows that the barrier "Lack of Communication", "Lack of Proper Documentation", "Improper/No unit testing", and "Lack of Resources, Knowledge and Skills" are the challenges whose intensity has been increased in the second decade with increase in GSD process. While the barriers "Lack of Compatibility", "Architecture Mismatch", "Heterogeneous Dev: Environment and Platforms", "Wrong COTS/OTS Product selection and customization" and "Lack of proper Component Interfaces" are challenges whose intensity has been decreased in the second decade as compared to the first decade. One reason for this may be that these

barriers are not only related to GSD environment but also relevant to local development and they may have got maturity with the passage of time. The barrier "Lack of Integration Planning and lack of Management" (with 38% frequency in the first decade and 31% frequency in the second decade) have almost got the same importance in both decades as the lack of both planning and management creates problems during the integration stage. The two barriers "Lack of Proper Documentation" and "Wrong COTS/OTS Product selection and customization" have significant difference in the two decades. The first one may have got more importance with GSD environment while the second one may have got maturity with the advent of component based software engineering (CBSE) in the second decade. We suggest further research for finding the reasons behind this significant difference.

Table 4. Comparisons of Critical Barriers Based on Each Decade

Critical Barrier	**Decade 1 1994-2003 (*n*=21)**		**Decade 2 2004-2013 (*n*=67)**		**Chi-square test (linear-by-linear association) α = 0.05**		
	Freq	%	Freq	%	**X^2**	**Df**	***P***
CB1 Lack of Communication	07	33	29	43	0.647	1	0.421
CB2 Lack of Proper Documentation	**04**	**19**	**30**	**45**	**4.413**	**1**	**0.036**
CB3 Lack of Compatibility	11	52	21	31	3.023	1	0.082
CB4 Architecture Mismatch	11	52	20	30	3.516	1	0.061
CB5 Lack of Integration Planning and lack of Management	08	38	21	31	0.326	1	0.568
CB6 Heterogeneous Dev: Environment and Platforms	09	43	19	28	1.532	1	0.216
CB7 Improper/No unit testing	05	24	23	34	0.806	1	0.369
CB8 Wrong COTS/OTS Product selection and customization	**11**	**52**	**16**	**24**	**6.037**	**1**	**0.014**
CB9 Lack of Resources, Knowledge and Skills	05	24	21	31	0.431	1	0.512
CB10 Lack of proper Component Interfaces	08	38	18	27	0.958	1	0.326

The values which have statistical significance difference (p<0.05) have been highlighted as bold.

5. Limitations

Some of the papers that we have used for data extraction have not explicitly mentioned that why they consider a factor to be a challenge which may be threat to internal validity. Similarly we may have also missed some relevant papers in the libraries that we have not searched. Some of the studies were case studies and self-experience reports which may be a threat to external validity.

6. Conclusion and Future Work

A total of 16 barriers in software integration have been found in this SLR study, which may hurdle the software integration phase in GSD environment. Among these, 10 barriers have been ranked as critical barriers based on the 30% criteria used by many researchers in the literature [23]-[25]. We also analyzed the barriers on the basis of two decades and found that the intensity of the barriers "Lack of Communication", "Lack of Proper Documentation", "Improper/No unit testing", and "Lack of Resources, Knowledge and Skills" has been increased in the second decade due to the wide spread use of GSD.

References

[1] Conchir, E., Agerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: Where are the benefits? *Communications of the ACM, 52,* 127-131.

[2] Romero, M., Vizcaíno, A., & Piattini, M. (2008). Toward a definition of the competences for global requirements elicitation. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 364-364).

[3] Khan, S. U., Niazi, M., & Ahmad, R. (2011). Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology (IST), 53*, 693–706.

[4] Smite, D., Wohlin, C., Gorschek, T., & Feldt, R. (2010). Empirical evidence in global software engineering: A systematic review. *Journal of Empirical Software Engineering, 15*, 91-118.

[5] Khan, K., Khan, A., Aamir, M., & Khan, M. (2013). Quality assurance assessment in global software development. *World Applied Sciences Journal, 24*, 1449-1454.

[6] Tekumalla, B. (2012). *Status of Empirical Research in Component Based Software Engineering*. Master of Science Thesis in Software Engineering and Management, Department of Computer Science and Engineering, University of Gothenburg, Sweden.

[7] Schneider, S., Torkar, R., & Gorschek, T. (2013). Solutions in global software engineering: A systematic literature review. *International Journal of Information Management, 33*, 119-132.

[8] Herbsleb, J. D., & Grinter, R. E. (1999). Splitting the organization and integrating the code: Conway's law revisited. *Proceedings of the 21st International Conference on Software Engineering* (pp. 85-95), NY USA.

[9] Land, R., & Crnkovic, I. (2003). Software systems integration and architectural analysis-a case study. *Proceedings of the International Conference on Software Maintenance* (pp. 338-347), Netherlands.

[10] Stol, K.-J., Babar, M. A., Avgeriou, P., & Fitzgerald, B. (2011). A comparative study of challenges in integrating open source software and inner source software. *Information and Software Technology, 53*, 1319-1336.

[11] Ilyas, M., & Khan, S. U. (2012). Software integration model for global software development. *Proceedings of the 15th International Multitopic Conference* (pp. 452-457), Islamabad, Pakistan.

[12] Ilyas, M., & Khan, S. U. (2015). Software integration in global software development: Success factors for GSD vendors. *Proceedings of the 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 119-124). Takamatsu, Japan.

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

[14] Larsson, S. (2005). *Improving Software Product Integration*. Ph.D Thesis, Department of Computer Science and Electronics, Malardalen University, Sweden.

[15] Bosch, J., & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of System and Software, 83*, 67-76.

[16] Farcas, C., Farcas, E., Krueger, I. H., & Menarini, M. (2010). Addressing the integration challenge for avionics and automotive systems from components to rich services. *Proceedings of the IEEE: Vol. 98* (pp. 562-583).

[17] Cataldo, M., & Herbsleb, J. D. (2011). Factors leading to integration failures in global feature-oriented development: An empirical analysis. *Proceedings of the 33rd International Conference on Software Engineering* (pp. 161-170).

[18] Kitchenham, B., & Charters, S. (2007). G*uidelines for Performing Systematic Literature Reviews in Software Engineering*, Keele University, UK.

[19] Niazi, M., Mahmood, S., Alshayeb, M., & Hroub, A. (2015). Empirical investigation of the challenges of the existing tools used in global software development projects. *IET Software, 9*, 135-143.

[20] Ilyas, M., & Khan, S. U. (2012). Software integration challenges in global software development

environment: A systematic literature review protocol. *IOSR Journal of Computer Engineering, 1*, 29-38.
[21] Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology, 56*, 54-78.
[22] Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology, 55*, 2049–2075.
[23] Niazi, M. (2004). *A Framework for Assisting the Design of Effective Software Process Improvement Implementation Strategies*. Ph.D thesis, University of Technology Sydney.
[24] Khan, S. U. (2011). *Software Outsourcing Vendors Readiness Model (SOVRM)*. Ph.D thesis, School of Computing & Maths, Keele University, UK.
[25] Khan, A. W., & Khan, S. U. (2013). Critical success factors for offshore software outsourcing contract management from vendors perspective: An exploratory study using a systematic literature review. *IET Software, 7*, 327-338.
[26] Bland, M. (2000). *An Introduction to Medical Statistics* (3 rd ed.). Oxford University Press.

Muhammad Ilyas is Assistant Professor of computer science at Federal Government College Batkhela, Ministry of Defense Pakistan. He is doing his PhD in computer science from University of Malakand, KPK, Pakistan. He is also a member of Software Engineering Research Group (SERG) University of Malakand. His research interest includes global software development, software integration, systematic literature review and empirical software engineering.

Siffat Ullah Khan is Assistant Professor of Computer Science in the Department of CS & IT, at University of Malakand Pakistan. He holds a PhD degree in computer science from the Keele University, UK. He is the founder of SERG University of Malakand. His research interest includes software outsourcing, empirical software engineering, systematic literature review and Software process improvement.