

Quality Assurance Assessment in Global Software Development

Khalid Khan, Adnan Khan, Muhammad Aamir and M.N.A. Khan

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad, Pakistan

Submitted: Jul 28, 2013; **Accepted:** Sep 11, 2013; **Published:** Sep 18, 2013

Abstract: Quality assurance in the context of Global Software Development (GSD) has multiple facets. Generally, it refers to the software development which is done in lesser cost, within the resources and in the given timeframe as well as the software bears high performance. However, the quality cannot be measured from a single perspective. In this study, a number of GSD benefits have been identified such as the reduced development cost, improvement of face to face communication skills, enhancement of knowledge through sharing of expertise, exploiting time zones differences to plausibly managing resources in order to increase productivity. Despite several benefits, GSD faces some challenges which include cultural differences, lack of communication, coordination, trust and security issues. Techniques used in GSD focus on information flow within the development team members to ensure efficiency, durability and scalability of the products by strictly adhering to time, cost and resources constraints. Various software development models allow us to understand and implement these concepts with ease. In this paper, we have also critically evaluated different GSD related tools and techniques by drawing their merits and demerits.

Key words: Global Software Development • Trust • Coordination • Communication • Quality Assurance

INTRODUCTION

Organizations are moving towards ERPs and enterprise software development rapidly due to their prominent benefits. The importance of global software development (GSD) is increasing day by day because of its cost effectiveness and short time to market the products. Industries cash the expertise of available skilled manpower across the world where the per unit working hours cost is low [1]. GSD means that software development work is distributed through different teams located in the same continent or more than one continent [2]. GSD has many advantages; for example, high resources are available throughout the world on affordable rates and a product becomes available in the global market in shorter span of time as well as the developed product stays competitive with the contemporary products [3]. The knowledge can be shared among different skilled people who belong to different continents and have diverse background in terms of ethnicity, religion, culture and spoken language. In GSD, the communication process is done through email, fax, text

messages and video chat etc., so it also improves written and verbal as well as face to face communication of the team members.

GSD also bears some disadvantages. The lack of communication is one of the big issues because in GSD every continent and every region may have different spoken languages and the project documentation is badly hampered due to contextual and subjective understandings of the team members. The physical distances among the development teams also matter a lot while coordinating the development process. The other key issue is the cultural barriers among the teams belonging to more than one continent. Other factors include time zone differences, quality of the process maturity, trust issues and the coordination issues.

Because of the cost effectiveness, many software development companies have turned to the global software development process. For this purpose, it is imperative that the software development processes and tasks should be well tested and managed as per the software scope. Mishra and Mishra [4] discussed inspection process for improving the software quality

process. In this regard, three approaches can be used for quality improvement:- tools must not be dependent on time; process must be acceptable and adaptable; and process runs on every platform. In terms of quality perspective, the project is successful if it is done within cost, in the specific timeframe and the software performance is acceptable [2].

The distribution of development task is very difficult when the distance among the teams is significant and the modules are coupled with each other. In such situation, culture issue also comes in and we cannot get the desired quality product. However, we can get the quality product when the modules are loosely coupled and teams are co-located. In this regard, Task Allocation based on Multiple Criteria (TAMRI) tool can define several scenarios for allocation of the distributed tasks [5].

Most of the risks identified in GSD are related to the field of management sciences. Verner *et al.* [6] state that most of the common issues get into software development process because of the human resources which include culture difference, training of team members having diverse educational background and effective communication among the team members. In addition, there could be management issues which involve the lack of planning, coordination and the control. Ramasubbu [7] talks about the configurationally choices to improve the productivity and enhancing quality of enterprise level GSD process. For profit orientation, a client chooses that continent in which cost per person per hour is within the acceptable range and does not have any implication on quality of the software i.e., cost should be selected in such a way that there is no compromise on quality of the product.

Jimenez [8] reviews distributed software development within the context of the FABRUM project. Result of the research identify main challenges in this area and suggests that further research is needed to address these challenges. The main challenges identified include communication, group awareness, configuration management knowledge management, coordination and time etc. The model for the distributed software development is the linking among planning, designing and production of the software development.

The requirement gathering is also a big issue in GSD. Solution to this problem is linked to the understanding of requirements, resolving conflicts, arranging face to face meetings and overcoming delays in meetings because of time zone differences. Increase in the number of

continents from where GSD teams belong to usually results in low quality product since developing small but interlinked modules separately is immaterial. Software requirement specification (SRS) has an important role in developing quality software products as it covers all the aspect of software requirement which should ideally be defect free, verifiable and fully traceable. The knowledge is transferred from customer to the developer and correct representation of it is still an unresolved challenge. Salgar and Engels [9] used a two stage approach to minimize the impact of ill-defined SRS and knowledge transfer issues. In the first step, some review techniques are used for improving product quality and in the second step it is delivered to the offshore development team which uses it for further development processes. GSD also faces trust, convincing and the security issues which require further considerations.

The rest of this paper is structured as follows: Section II provides a brief background on GSD followed by literature review in the third section. Based on the critical evolution of the literature, we discuss challenges and benefits of GSD in section IV and section V respectively. Finally, we conclude with some possible future work in the last section.

Background: Outsourcing is a state of affairs in which a company imports all or some parts of their product from another vendor [3]. Outsourcing has different subtypes e.g., on- shore or domestic outsourcing (in which both clients and vendors are from same country but different locations), near-shore outsourcing (in which the teams are from neighbor countries) and Off-shore outsourcing (in which teams are from the far-flung regions/countries) [3]. GSD, Distributed Software Development and outsourcing are primarily different names for a unique software development technique through which the clients get services from an outside organization known as vendors. This technique is of the interest to the clients due to lower labor cost, faster delivery and access to skilled human resources. Despite the supplementary benefits, this technique suffers from some limitations like: teams involved in the software development activities usually have different cultural backgrounds; face to face meetings are difficult due to their remote locations; time zones and working hours of the clients and vendors do not match; there are language barriers among both the parties which lead to communication and coordination issues; and trust between both the parties always remain a major concern.

Literature Review: Mishra and Mishra [4] elaborated global software development process and addressed management and quality assurance issues for providing global software scrutiny process with particular reference to the area of distributed software development. GSD platform facilitates users globally to develop high quality software. The various stages in GSD process include follow-up stage, meeting stage, rework, individual inspection stage and the setup stage. The role of global software inspection is becoming significant for distributed software development and it is useful tool to check software quality. Bird *et al.* [10] described the effects of distributed software development on software quality and pointed out that a number of issues such as communication, cultural and expertise transfer difficulties could arise in the distributed software development environment. The past failures in the distributed software development can be linked to code churn, dependency and code testing. Still the distributed environment faces lots of challenges such as late reviews, less shared project attention, limited interaction, trust, unreliable development and difficulty of synchronous interaction to build effective environments.

Quality is one of the major factors in every project. The software development in distributed environment, especially for the large projects, has become a popular area of interest nowadays. GSD methodologies and practices offer a lot of advantages in terms of lower cost, smaller development time and better quality [2]. Planning the QA activities is a critical task in GSD environment. These activities are applied in project analysis phase, project execution phase and project conclusion phase. In project analysis phase, the main focused is on budget and project tasks distribution among the development teams. In project execution phase, different QA strategies are applied such as quality management network, change control, active risk management, quality audits, inspection strategy, delivery strategy, reporting and performance measurements. In project conclusion phase, the experiences gained from the projects are summarized. It has been observed that clear communication, close coordination and early fault detection reduce the time and cost of the project and improve quality of the product.

Lamersdorf and Munch [5] identified six main goals of GSD and factors influencing these goals. These goals include cost, time, quality, human resource, intellectual property protection and proximity of the client and customer. The main factors influencing these goals are physical distances, language differences, organizational

differences, cultural differences, infrastructure differences and working hour differences. However, still most of the work assignment is done unsystematically in GSD and there is a need for proper and systematic decision support model for task allocation in GSD projects. The GSD tasks are often large-scale, challenging and exposed to an enhanced level of risk of disappointment. Vernon *et al.* [6] categorized GSD risk and strategies to control these risks so that project outcome becomes a success story and the client gets satisfied. Bird *et al.* [10] analyzed GSD at project level in terms of quality, income and the productivity. The main consideration in GSD is to determine how many continents and time zones are involved in the software development activities.

GSD industry is assigning more preference to team relocation i.e., groups of development teams working on the same project are distributed and are located in different countries and continents. One of the main reasons of team relocation is to save cost and ensure better quality of the product. Industries search and hire skilled workers from multiple countries having different time zones and cultural background to achieve a quality product with lower cost. Jimenez *et al.* [8] carried out a review on distributed software development within the context of the FABRUM project. Result of the research identify main challenges in this area and suggests that further research is needed to address these challenges. The main challenges identified include: communication, group awareness, configuration management, knowledge management, coordination and time differences.

Cataldo and Nambiar [11] report how a quality system can be achieved in distributed development environment. Component and modules coupling have a great impact on software quality. The degree of coupling based on logical dependencies is reported in the study. The logical dependencies increase the number of functional bugs when the software is being developed globally. Furthermore, it becomes evident that balanced distribution of tasks can improve quality of product and minimizes the number of defects. Cohesion and coupling are two important factors which need special attention in software projects. But, in global software development, the distribution of work and assignment of tasks are the key pillars which make the project success. In GSD environment, proper planning is required to reduce the coupling and project modules dependencies.

The knowledge transfer in traditional software development is very important and comparatively easier, but in GSD it is quite difficult. Salgar and Engels [9]

highlight that review of software requirement do not include entire information of stakeholder and, therefore, the knowledge transfer from the client to the development team using knowledge representation method is necessary. These issues can be resolved in two stages. In the first stage, we can increase the quality of SRS and the acceptance test case specification derived from the acceptance test driven development. And in the second stage, we can send it to the offshore team and link it to transfer of knowledge to the customer. In the context of GSD, software quality management is the collection of methods and procedures to produce a highly quality software through development process. Managing product quality in GSD is a key challenge and for this purpose, it is necessary to train persons in their own preferred tools to improve their skill level. Colomo-Palacios *et al.* [12] used qualitative research methods for addressing the above issues. The mentoring tool enables both the senior and junior management in their career building and transfer of knowledge.

Ramasubbu *et al.* [13] highlight the development of strong governance plans duly facilitated by models based on distributed software development. The authors focus on development and testing of governance rules that are best suitable for distributed software development projects. In case of absence of governance plans, software distributed teams make use of software engineering rules and normal process frameworks based on the co-located context. Software development spans across multiple development locations which results in various management challenges. This necessitates a framework for software distributed development performance that is based on empirically test and existing project schemes associated to cost estimation as well as team organization to accommodate in distributed measurements. Ågerfalk *et al.* [1] identified potential benefits of GSD and classified them in three different classes which are organizational benefits, team benefits and process level benefits. An organization benefits includes improved resource allocation. Team benefits include reducing the development cost and improving the team autonomy. Process level benefits include improved documentation and clearly defined processes. Furthermore, the benefits are categorized in two broad categories: known (tangible) benefits and unknown (intangible) benefits.

Issues and Challenges in GSD: GSD is becoming popular day by day and software industry hires manpower from

the countries where the labor cost is low. The cost, quality and time are the three main factors taken into consideration in every software development project. The cost and time is easy to assess in the global software projects, but the research is not mature enough to measure quality of the product. Instead of these benefits, there are some issues and challenges in the global software development which require further studies and research to address these issues and challenges.

Communication: Communication plays an important role in the software development projects. Mainly two types of communication are carried out during the software development: communication among the development teams and communication with the customers [2]. Both types of communication play vital role in the project success. Teams' relocation increases the communication gap. Teams share and exchange a large amount of information during communication and in distributed environment it leads to several challenges which occasionally create misunderstanding among the teams. Customer collaboration plays an important role in software development especially when teams follow agile software development methodologies [8]. In such environments, GSD is a challenge to apply.

Configuration Management: Change control and software versioning are the most important areas of software project management. The conflict occurs in GSD environment related to source code control. Synchronization becomes more complex when teams are located in different time zones and especially when the team size grows.

Quality: Software industry assigns much importance to software cost and quality. The cost goal can be achieved by applying GSD methods by hiring skilled manpower from the areas where labor cost is low. But the research is still immature to evaluate the quality of the product in global environment [8]. It is suggested that automated code inspection and use of coding standards can be adopted to address the issue of quality. But how to evaluate quality of the product during the global software development is a critical question which requires further study and research.

Benefits of GSD: In this section we describe some common benefits of GSD.

Cost Effectiveness: Cost is the most critical factor which requires proper attention and planning in all the disciplines. Nowadays software industries are just taking the advantage of co-relocated teams to reduce the development and other associated costs. The cost includes development costs, management and coordination cost. The software developing companies hire people from those places where per hour work rate is less.

Formal Documentation: In GSD, teams generally avoid face to face communication. Most of the communication between teams is carried out on emails and formal project documentation. Documentation plays an important role in distributed development.

Team Experience: In GSD, the development teams are located at different cities and countries across the world. Their languages, culture and norms are not same and even the technologies they are using might be different from each other. If the teams and personnel are working in such environment, then their collaboration can play an important role in terms of experience sharing, technology understanding and the dynamics of the technological world. The team members share their experiences with each other and knowledge domain of the team members grow in such environments.

Conclusion and Future Work: In this study, we identified benefits related to GSD which improve the quality of software products. This study revealed different aspects of software development that GSD can enhance. We found cost selection as an important factor in GSD. Cost should be selected in such a way that there is no compromise on quality and trust. We studied approaches related to systematic literature review, systematic review procedure, acceptance test case specification and process-based learning to obtain the expected quality results. Furthermore, this review helped us find challenges and issues we faced in global software development such as cultural issues, lack of communication, coordination, trust and security issues. In this study, we identified GSD challenges such as cultural issues, lack of communication, coordination, trust and security issues. As such there is no model that covers all aspect of quality factors to measure the impact of defects and how it affects the product quality. In future we intend to propose a model which covers quality factors in GSD.

REFERENCES

1. Ågerfalk, P.J., B. Fitzgerald, H.H. Olsson and E.Ó. Conchúir, 2008. "Benefits of Global Software Development: The Known and Unknown," Making Globally Distributed Software Development a Success Story, Lecture Notes in Computer Science Volume 5007, 2008, pp: 1-9.
2. Ivèek, M. and T. Galinac, 2008. "Aspects of Quality Assurance in Global Software Development Organization," In: The 31st International Convention MIPRO 2008, pp: 150-155.
3. Khan, S.U., M. Niazi and R. Ahmad, 2010. "Factors Influencing Clients in the Selection of Offshore Software Outsourcing Vendors: An Exploratory Study Using a Systematic Literature Review," The J. Syst. And Softw., 84(4): 686-699.
4. Mishra, D. and A. Mishra, 2012. "A Global Software Inspection Process for Distributed Software Development," J. Univ. Comput. Sci., 18(19): 2731-2746.
5. Lamersdorf, A. and J. Munch, 2010. "Studying the Impact of Global Software Development Characteristics on Project Goals: A Causal Model," The Open Softw. Eng. J., 4: 2-13.
6. Verner, J., O.P. Brereton, B. Kitchenham, M. Turner and M.K. Niazi, 2012. "Evidence-based Global Software Engineering Risks Extracted from Systematic Literature Review," Technical Report TR-2012-01, School of Computing and Mathematics, Keele University, Keele, Staffordshire, UK.
7. Ramasubbu, N., M. Cataldo, R.K. Balan and J.D. Herbsleb, 2011. "Configuring Global Software Teams: A Multi-company Analysis of Project Productivity, Quality and Profits," The International Conference on Software Engineering (ICSE'11), May 21-28, 2011, Waikiki, Honolulu, HI, USA.
8. Jiménez, M., M. Piattini and A. Vizcaño, 2009. "Challenges and Improvements in Distributed Software Development: A Systematic Review," Adv. Softw. Eng., 1: 1-16.
9. Salgar, F. and G. Engels, 2010. "Knowledge Transfer in Global Software Development: Leveraging Acceptance Test Case Specifications," In: Proceedings of the 32nd International Conference on Software Engineering, Cape Town, 2-8 May 2010.

10. Bird, C. and N. Nagappan, 2009. "Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista," In: Proceedings of the 2009 International Conference on Software Engineering (ICSE'09).
11. Cataldo, M. and S. Nambiar, 2012. "The Impact of Geographic Distribution and the Nature of Technical Coupling on the Quality of Global Software Development Projects," *J. Softw. Evol. and Process*, 24(2): 153-168.
12. Colomo-Palacios, R., P. Soto-Acosta, A. Mishra and Á. García-Crespo, 2011. "Software Quality Management Improvement through Mentoring: An Exploratory Study from GSD Projects," In: Proceedings of the 2011th Confederated international conference on On the move to meaningful internet systems (OTM'11), pp: 190-199.
13. Ramasubbu, N. and R.K. Balan, 2008. "Towards Governance Schemes for Distributed Software Development Projects," In: Proceedings of the 1st International Workshop on Software Development Governance (SDG) in ICSE, Leipzig, Germany.