A Project Management Framework for Global Software Development

Ritu Jain
Research Scholar, School of Computer Science& IT,
Devi Ahilya University, Indore,
M.P., India
91-9630870019
ritujain_doc@rediffmail.com

Ugrasen Suman
Professor, School of Computer Science& IT,
Devi Ahilya University, Indore,
M.P., India
91-9826953187
ugrasen123@yahoo.com

DOI: 10.1145/3178315.3178329

ABSTRACT http://doi.acm.org/10.1145/3178315.3178329
Global software development (GSD) is a prevalent trend which has fascinated most software companies. However, the failure rate of GSD projects reveals the fact that these types of projects are not an easy endeavor. Management of GSD project is a domain where standards are still lacking and companies are still struggling to acquire a win-win situation. Project management body of knowledge (PMBOK) provides a standard framework for managing projects. However, the framework does not consider the aspects of GSD. Thus, it can't be applied directly for GSD projects. In this paper, we have proposed a project management framework for GSD projects. This framework assimilates the knowledge areas of PMBOK with knowledge areas needed for effective management of GSD. It would guide GSD project manager about the aspects to be considered while executing distributed projects. This framework would also act as a baseline to researchers for further investigation in GSD project management domain.

Categories and Subject Descriptors

• **Social and professional topics~Project and people management** • *Software and its engineering~Collaboration in software development*

General Terms
Management

Keywords
Global software development, distributed software development, project management, challenges, framework, knowledge areas.

1. INTRODUCTION
Fierce competition and budgetary pressures have motivated software companies to distribute their development activities across national and organizational borders. This type of development in which team members belong to different countries is termed as Global Software Development (GSD). The main motivation behind GSD is the desire to reduce cost of development by utilizing pool of low-salaried, skilled software engineers belonging to less developed economies. Other impetus includes closer proximity to customer, reduced time to market by exploiting time zone differences, improved work modularization, innovation and learning. However, the potential benefits of GSD are only partially achieved due to several distances that interfere with management and execution of these projects. The distances that interplay between distributed teams are geographical, temporal, socio-cultural, and organizational which result into communication, coordination, control, and collaboration challenges [2].

Team members of distributed team have limited or no face-to-face interactions, belong to different cultures, speak different native languages, and work in different time-zones. Therefore, they have limited opportunities for coordination, collaboration, and trust building [4]. Large numbers of distributed projects fail due to the absence of effective management of distributed projects [6]. However, collocated project management techniques and strategies do not consider the impact of these distances thus, need to be reassessed and modified for distributed projects [9]. A strict communication plan, awareness, respect for each other culture and unbiased management is necessary for building cohesive team [9, 36]. Thus, a project management strategy that encompasses social as well as technical aspects can alleviate the impact of these distances on the working of distributed teams [4, 6].

This paper is structured as follows: Section 2 presents a brief overview of global software project management and its associated challenges. Section 3 discusses related work whereas section 4 presents the proposed framework for global software project management. Finally, section 5 concludes the paper.

2. GLOBAL SOFTWARE PROJECT MANAGEMENT
Project management is a discipline that governs skills, knowledge, tools, and techniques that can help in fulfilling project requirements towards successful software development. A project management framework consists of stakeholders, knowledge areas, tools, and techniques for managing, monitoring, and controlling projects. Project stakeholders are the individuals who are either involved in or influenced by the project. The knowledge areas for project management are scope, time, cost, quality, risk, human resource, communication, procurement, stakeholder, and integration management [46]. Some of the criteria that can help in achieving project success are customer involvement, clear business objectives, competent project leader, skilled team members, efficient delivery process, appropriate metrics, integrated tools and infrastructure. Effective project management techniques enhance reliability, productivity, employee morale, profit and reduce development time as well as cost. It helps teams to coordinate effectively, achieve strategic goals, and control physical and human resources in a better way [46]. Satisfied customers, reduced risks, effective decision making, better knowledge and quality management techniques are few more benefits of project management methodology [3].

Distributed projects face numerous challenges in addition to the general challenges of project management [9]. Geographical separation and time zone differences negatively influence frequency of communication, transparency, visibility, decision making, and issue resolution. These problems are compounded by socio-cultural dissimilarities and linguistic differences in distributed teams which adversely affect team cohesiveness, trust [27], and knowledge sharing [50]. Further, organizational distances result into process as well as tool mismatch and dissimilar working culture. These distances hamper communication, coordination, control, and collaboration processes.

The project management challenges encountered in GSD are consolidated in Table 1. Managing geographically distributed projects is a more complex task as compared to collocated software projects [9, 47]. These problems lead to misunderstandings, conflicts, distrust, fear, weak personal relations, frustration, rework, delays and project failure in many cases [27, 50]. Lack of careful planning, inexperience

and ignorance of these distances can result into project failure [31]. Thus, these projects need to be addressed through a well-planned project management strategy [9].

Table 1: Project management challenges in GSD

Communication problems due to GSD distances
Geographical
– Reduced communication frequency [6, 27].
– Face to face meeting is difficult [1].
– Increased cost of communication [9, 42].
– Hard to convey urgency [27].
– Tacit knowledge transfer and management difficulties [27, 35].
– Preparation overhead of distributed meetings [27].
– Unplanned remote communication causes interruption in work [27].
– Insufficient communication causes developer to assume work [27].
Temporal
– Synchronous communication is difficult [4, 6, 27].
– Uncomfortable synchronous meetings [27].
– Slow and intermittent information transmission [27].
– Delayed doubt resolution [27].
– Low participation of remote members in distributed meetings [26].
Socio-cultural
– Misunderstandings due to different ways of expression [27].
– Offshore members hesitate to ask queries or discuss issues [27].
– Slow knowledge transfer [27].
– Difficult to initiate communication [42].
– Linguistic difference causes ineffective communication [27].
– Frustration due to different accents [14].
Organizational
– Incomplete domain knowledge [27].
– Misinterpretation due to diverse terminologies [27].
– Ineffective doubt resolution due to unawareness about remote member's tasks [27].
Coordination problems due to GSD distances
Geographical
– Lack of transparency [27].
– Poor visibility [4, 9, 27].
– Difficult to manage dependencies [27].
– Change management problems [27].
– Lack of common understanding in offshore members [27].
– Difficult to share artefacts [27].
– Unsystematic handover [27].
– Incompliance of documented process [27].
– Unresolved doubts [27].
– Obsolete documentation causes delay [27].
– Coordination cost is increased [1].
– **Temporal**
– Delayed issue resolution [27].
– Extended feedback loops [27].
– Coordination challenges [6, 27, 35].
– **Socio-cultural**
– Reduced cooperation [9].
– Different holiday pattern [27].
– Differences in work practices, ethics and values [27].
– Conflict management problems [27, 35].
– High attrition rate at offshore due to excessive work and cultural reluctance to refuse to impractical work or deadline [9, 27].
– **Organizational**
– Process mismatch [27, 35].
– Diverse process maturity, tools, standards [9, 27].
– Different expertise and experience levels [9].
– Tool mismatch [27].
– Safety of intellectual property [35].
– Lack of GSD experience [27].
– Disinterest in reviewing remote work [27].
– Different corporate culture [29].
– Problems hiding from client [29].
Collaboration problems due to GSD distances
Geographical
– Task allocation problems [27, 35]
– Lack of standard cost and effort estimation techniques [35].
– Political risk related to international relations [9]
– Risk identification and management difficulties [35].
– Difficult to evaluate performance of offshore team members [27].
– Rules, regulations, and laws vary across countries [27].
– No standard quality assurance principles and rules for GSD [53].
– Lack of standard method for requirement definition in GSD [4].
– Difficult to monitor compliance of documented process [42].
Temporal
– Slow decision making [27].
Socio-cultural
– Volatility associated with local and foreign exchange currencies [4].
Organizational
– Internal politics [4].
– Unplanned and sudden shrinking or expansion of teams [27].
– Client can't choose vendor developers [27].
– Vendors are usually not authorized to take decisions [27].
– Asymmetry in processes, policies, and standards [27].
Control problems due to GSD distances
Geographical
– Lack of common vision.
– Unawareness about remote team's work [6, 27, 35].
– Inadequate team cohesiveness [27].
– Low quality work is allocated to offshore team.
Temporal
– Burn out of offshore members in order to synchronize time zone difference [27].
Socio-cultural
– Lack of team cohesiveness and us-them culture [27, 42].
– Lack of trust [4, 27].
– Weak personal relations.
– Limited cooperation, knowledge sharing, low motivation, and reduced productivity due to fear of job loss [9].
– Limited training [27].
Organizational
– Low mutual understanding [27].
– Dissimilar work culture [27].
– Repetitive work reallocation, retraining due to high offshore attrition rate [9].
– Different view of authority and hierarchies [27].
– Low client participation [27].
– Disorganized command chain, and leadership [27].
– Improper IT infrastructure [35].
– Political risks [9].
– Client not ready to take responsibility for delay [27].
– Client doesn't want to share their real data with vendor [27]

3. RELATED WORK

There are many project management challenges in GSD. Most of the research has covered few challenging aspects of GSD, i.e., risk management, task allocation, effort estimation, virtual team structure, knowledge management, and applicability of Scrum. Experience sharing, empirical research as well as literature review has been performed for exploring challenges, best practices, lessons, tools, and models. However, none of the research has compiled all the essential knowledge areas needed for effective GSD project management. Also, the PMBOK does not consider the challenges of GSD. Thus, a project management framework needs to be exclusively designed for GSD.

Schwaig et al. explored issues and solutions for nine PMBOK knowledge areas for offshore outsourcing [45]. However, they have not identified any new knowledge area needed to combat GSD problems. Ramasubbu et al. proposed distributed process maturity framework similar to Capability Maturity Model, in which twenty four new key process areas (KPA) are identified for managing distributed projects. However, brief overview of the framework is presented without elaborating KPA details [41]. Richardson et al. identified GSD factors for which explicit or implicit process areas are available in CMMI. They have also explored GSE factors for which process areas are not specified in Capability Maturity Model Integration (CMMI) framework. Thereafter, they proposed a framework (global teaming model) which supplements CMMI in global software engineering [44]. Ralyte et al. proposed a framework for supporting management in

distributed information system development. The framework discussed problems and solutions encountered in communication, coordination, control, development and maintenance activities due to geographical, temporal, socio-cultural, organizational, technological, and knowledge distances in GSD [38].

In this research paper, we have first identified challenges in GSD project management, and then proposed a project management framework for GSD projects. It can alleviate GSD challenges and aid in effective management of GSD projects.

4. GLOBAL SOFTWARE PROJECT MANAGEMENT FRAMEWORK

In the global milieu of fierce competition and innovation, organizations should continuously monitor and assess their project management strategy to realize competitive advantage [4]. Project management knowledge areas that need to be addressed for successful distributed projects execution are shown in Figure1. The proposed framework covers feasibility and risk management, virtual team management, knowledge management, scope and resource management, performance management, and GSD integration management. Each of these knowledge areas are further discussed in the following subsections.

Figure 1. GSD Project Management Framework

4.1 Feasibility and Risk Management

GSD projects are more prone to failure as compared to collocated projects due to the inherent risks of geographic, temporal, cultural, and organizational separation. Thus, feasibility of executing the project in a distributed manner must be checked through comprehensive feasibility study before project commencement. Feasibility test need to be followed by meticulous risk management.

4.1.1 GSD Feasibility Study

GSD can exist in two forms: offshore outsourcing and offshore insourcing. More challenges are associated with offshore outsourcing as compared to offshore insourcing. The risks in offshore outsourcing can be reduced by checking economic, technical, behavioral, political, and legal feasibilities. Feasibility test can help a company to decide whether the project can be carried out globally.

4.1.1.1 Economic feasibility

Estimation of cost savings from GSD should consider the factors such as time zone difference, tool usage, previous working relations, experience, domain knowledge, and proficiency of offshore employees. Expenses incurred on travel, trainings, and tool infrastructure set up should not outweigh GSD cost benefits.

4.1.1.2. Technical feasibility

Factors considered are project complexity, resource availability; compatible tool infrastructure as well as business model for GSD. Architectural adequacy of project for distribution should also be verified. Vendor's process maturity level, project management capabilities, capacity, employee attrition rate, and technical skills should also be investigated [45].

4.1.1.3. Behavioral feasibility

GSD practitioners need to relocate, travel, and attend meetings in early morning or late night. Vendor should be ready to understand and embrace client's development processes, practices, vision, and priorities such as quality and deadline. Prior working relationships with the client reduces differences in work ethics and increases cultural understanding.

4.1.1.4 Political feasibility

Political stability and appropriate national infrastructure of a country is prerequisite while choosing a vendor [45].

4.1.1.5 Legal feasibility

Laws related to data privacy, international trade, labor and software piracy and their enforcement in vendor country should be analyzed for GSD feasibility [45]. Contracts should clearly specify each and every term, condition, milestones, payment method, expected quality, and project's schedule. Expectations about functional and non-functional requirements should also be specified [45]. Contracts should also specify how dispute or early termination will be managed [45].

4.1.2 Risk Management

Risk in software engineering can be defined as a particular aspect or characteristic of software development, which if neglected, will increase the probability of project failure [51]. GSD distances introduce many risks in addition to the risks observed in collocated development. Integration of people, processes, and skills distributed across geographical locations make GSD a complex endeavor [18]. There are several sub-processes in GSD which are more prone to risks due to GSD distances. These sub-processes are communication, coordination, control, and collaboration. Software development processes such as requirement engineering, architecture, configuration management, adaption of agility need more attention as compared to other processes. Project planning in GSD projects, training team members, cultural and social integration also incur heavy risks [51]. All the stakeholders involved in GSD should be attentive and be involved in identifying risks throughout product development. Project manager can also utilize risk repository to identify the risks of the project. Once a risk is identified, it should be analyzed, planned, and proper mitigation techniques be applied.

4.2 Virtual Team Management

Virtual team is a group of geographically, organizationly, and/ or temporally distributed individuals who work on a project by coordinating their activities with the help of information and telecommunication technologies [10]. Members of virtual team belong to different countries and/or organizations, possess different native languages, and can be temporally separated. Misunderstandings, fear, distrust, conflicts, and weak personal relations are some of the problems that these teams face during GSD. The factors behind these problems are team uncohesiveness, inadequate communication, cultural unawareness, poor team structure, demotivated members, ineffective conflict management and poor skills management. Thus, successful operation of GSD projects need robust virtual team

management ensuring team cohesion, team configuration, motivation, task allocation, and conflict management as shown in Figure 2 and discussed in the following paragraphs.

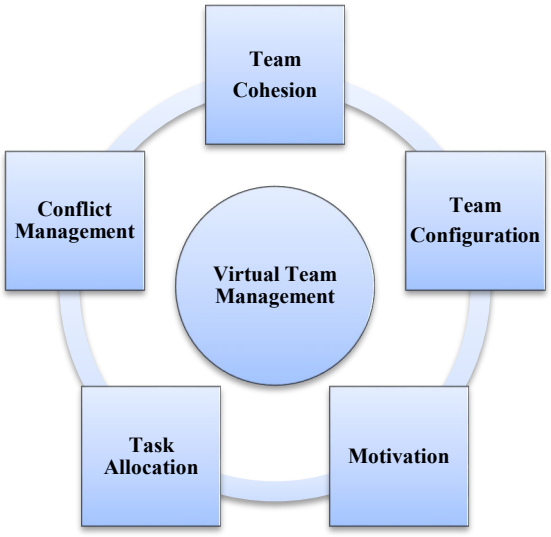

Figure 2: Virtual Team Management

4.2.1 Team Cohesion

Cohesiveness of a virtual team can be enhanced by collocated project inception, frequent exchange of team members across locations, relocation of an offshore senior member to onsite, rotation of senior manager to offshore sites, and cultural trainings [47, 34, 26]. Team building activities during the collocation helps members to get familiar with each other and will tremendously improve cultural awareness. These practices will also improve awareness of each other's working style, holidays, and orientation towards hierarchy. Organizational objectives, project vision and goals should be clearly articulated at the inception of project. Offshore team's awareness about expected quality, process, and schedule adherence will avoid misunderstandings and rework. Distributed members must be aware of rules and regulations to be followed during project. A shared vision for project can align team members towards shared goals [33].

4.2.2 Team Configuration

Team configuration aspects such as skill management, prior working relationship, team dispersion, roles and responsibilities, and project manager qualities need to be focused for GSD projects.

4.2.2.1 Skills management

Highly competent and proficient team members should be selected for GSD [14]. Offshore team members can acquire professional certification in technology and English language. These certifications can ensure technology competence as well as language acquaintance of offshore members to onshore managers. Soft skills also need to be encouraged. These practices will reduce the learning curve of offshore developer and improve trust for offshore team. Team members at a location should have complementary skills to complete a feature. In client vendor relationship, onshore manager should select offshore members to build a technology strong and competent team.

4.2.2.2 Prior working relationship

Prior working relationship between client and vendor get the benefit of already established cohesion, trust and cultural knowledge. Thus, it improves productivity of team and reduces the chances of misunderstandings [26].

4.2.2.3 Team dispersion

Number of locations is directly proportional to the coordination effort required for development. Abrupt increase/ decrease in team members should be averted as new members lack tacit knowledge and departing members carry tacit knowledge with them [26].

4.2.2.4 Roles and responsibilities

Roles and responsibilities should be clearly declared at the inception of the project [47, 44]. Project members should be aware of all other project participants and their roles and responsibilities.

4.2.2.5 Project manager

He should be able to continuously monitor and adjust the process according to the circumstances [42], be aware of cultural peculiarities of remote team's culture, working style, and their holiday pattern. He should be tech-savvy, unbiased, open, and able to resolve conflicts.

4.2.3 Motivation

Team motivation largely impacts productivity, software quality and success of project in software engineering [28]. Motivation of team members can be maintained through rewards and accomplishments, freedom of expression, unbiased management, frequent feedbacks, respect and autonomy.

4.2.3.1 Rewards and accomplishments

Clear rules should be defined for rewards and accomplishments irrespective of locations. Team coordination and collaboration should also be reviewed frequently, admired and rewarded accordingly. This will encourage team work, cohesion and cooperation [27].

4.2.3.2 Freedom of expression

All team members can express their ideas and present innovative solutions. Open discussions should be encouraged [33].

4.2.3.3 Unbiased management

Senior managers, sponsors, and management should treat all members in an unbiased manner irrespective of locations. Challenging and creative jobs should be evenly divided among onshore and competent offshore team members to signify impartiality on part of distribution as well as culture. This will motivate remote team members, improve team ownership, and reduce the friction between onshore and offshore teams [28].

4.2.3.4 Frequent feedbacks

Distributed design review, code review, sprint review, and quality assurance activities should be frequently performed to increase feedback of team members. Periodic measurement of team productivity can also keep developers motivated towards work and team cohesiveness. This will improve the quality of overall product as well as good performers will get motivated [28].

4.2.3.5 Respect and autonomy

Distributed members should respect each other's culture and values. Distributed sites should have enough autonomy and freedom for decision making.

4.2.4 Conflict Management

Conflict management between different parties is crucial for success of GSD project.

4.2.4.1 Between team and customer

Eagerness to achieve cost and time benefits of GSD may cause negligence of GSD problems and settings of unrealistic milestones, which may result into low quality products or undue pressure on team members. This undue pressure may also lead to high attrition rate of employees. Therefore, project manager must be capable of convincing

the customer, managing pressure, and maintaining quality of the product.

4.2.4.2 Between offshore and onshore team

The relationship between onshore and offshore team must be strengthened with trust, open as well as unbiased management. Fear of job loss in onshore members due to low salaried offshore developers should be dealt cautiously. Project manager should immediately resolve all the conflicts, issues and misunderstandings crop up between distributed team members during product development. Contracts should provide justice to client as well as vendor.

4.2.5 Task Allocation

Effective allocation of tasks to distributed teams can aid in accruing the potential benefits of GSD whereas incorrect allocation can increase the risks associated with GSD and can lead to project failure [30]. There are several factors that need to be considered while allocating tasks to distributed teams in GSD. Dependencies between tasks, stability of requirements, product architecture, and size complexity of task to be distributed need to be regarded during task allocation in GSD [30]. Other factors include technical expertise, temporal differences, geographic distance, resource cost, local government laws, intellectual property ownership; reliability and maturity level of vendor. Language proficiency, experience of team members, time that distributed members can devote, allocated travel budget, communication, coordination and knowledge sharing mechanism also need to be considered during task allocation.

4.3 Knowledge Management

Knowledge management is a process of creating, utilizing, distributing, and managing the knowledge of an organization. Appropriate knowledge management is particularly essential in GSD where project team is dispersed across time and space [43]. However, geographical and temporal distances in GSD, hinder knowledge sharing whereas, socio-cultural and organizational distances introduce differences in the way knowledge is managed [16]. Earl has proposed a framework in which knowledge management strategies are classified into seven schools [17]. Out of seven, six schools can be used for GSD knowledge management, as shown in Figure 3.

Figure 3: GSD Knowledge Management

4.3.1 System school

It captures project knowledge with the help of technology by storing it in repositories and knowledge bases [16, 17]. In GSD, globally accessible repositories can be used to store and share knowledge among local as well as dispersed team members. These repositories can store information about project vision, process policies, and project terminologies. The repositories also store artefacts related to user stories, presumed risks and their mitigation strategies, design and coding standards, product architecture design decisions, and test cases [25]. These repositories should be regularly updated and maintained. Communication tools such as email, skype, instant messenger, audio-video conferencing, live meeting, and chat rooms help GSD teams to share information effectively. Project management tools such as version control (subversion, virtual source safe, and concurrent version system), jira, team foundation server, redmine, electronic kanban board and Microsoft project plan support effective coordination and knowledge management. Collaboration tools, such as screen sharing software, remote desktop software, webcams, application sharing software, and shared whiteboard software also aid in knowledge sharing.

4.3.2 Cartographic school

It emphasizes mapping of organizational knowledge and is concerned with storing information about "who knows what" [17]. In GSD, wiki helps to create a community of members aiming to successfully communicate project information [34]. Storing member information with photo, contact number, role, responsibilities, and skills can help distributed members to find right person [25]. Relocation of offshore senior member at onshore location, visits, exchanges, and collocated inception of project are some of the ways in which tacit knowledge can be shared and managed [25].

4.3.3 Engineering school

It is concerned with business process reengineering and focus on improving processes to encourage knowledge flows [17]. Practices such as synchronization of work hours, cross-site design review, code-review, distributed daily scrum, weekly meetings, sprint review, and retrospectives enhance knowledge flow within distributed teams. Language, cultural as well as technical trainings can also contribute to knowledge flow. Dispersed members can use electronic kanban board to visualize the work performed at remote locations [25].

4.3.4 Organizational school

It is concerned with formation of collaboration networks and communities for sharing and pooling knowledge [43, 17]. Collocated inception of project, frequent visits, remote pair programming, and wikis help in the formation of community in GSD projects [25].

4.3.5 Spatial school

It uses the design of office space for exchanging knowledge. It helps people to socialize and exchange tacit knowledge [17]. Some of the measures taken to exploit this school for knowledge exchange in GSD are chat rooms, permanent video connections [21], corner and ceiling cameras as well as ambient microphones [52]. Electronic Kanban screens and information radiators screens can be placed at all locations. Photo chart of whole project team and clocks of all involved time zones at each office will also improve awareness.

4.3.6 Strategic school

It emphasizes organizational strategies that can use knowledge for value creation and acquire competitive edge over others [17]. Strategies such as establishing communication protocols, quality parameters, product vision and process policies during collocated inception of project and storing them in globally accessible repositories can help GSD companies in knowledge sharing. Enriching user stories with use case diagram, test cases, and video recordings of requirement meetings improve requirement understanding at remote locations.

4.4 Scope and Resource Management

Scope and resource management encompasses activities which decides the work to be included in the project, estimate and monitor the time

and cost needed to complete the project. It encompasses management of scope, cost and schedule as depicted in Figure 4.

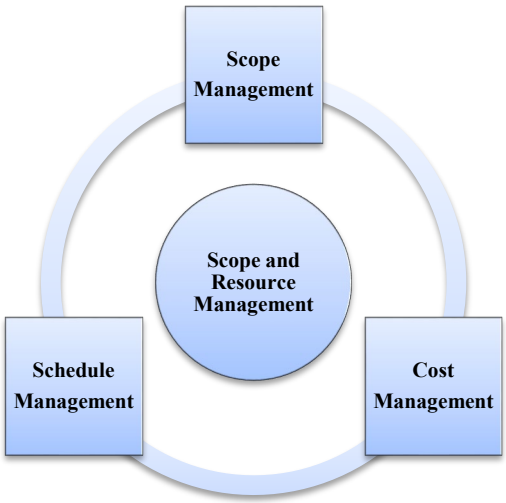

Figure 4: GSD Scope and Resource Management

4.4.1 Scope Management

Project scope management decides, defines and control the processes and work to be included in the project. It ensure that the project team and stakeholders has a common understanding of what is to be developed and which processes will be used to create them [46]. In case of GSD, it is difficult to gain this understanding due to geographical, organizational, and socio-cultural distances. Extensive planning during collocated inception of project, involving offshore members during scope establishment, repositories storing project scope statement, manager's expertise and experience in handling GSD projects are few ways to reduce the problems caused due to these distances.

4.4.2 Cost Management

Cost management involves estimating cost of completing the project, creating the budget and monitoring it. In case of first time collaboration, instant cost savings are difficult, as, initially offshore developers lack domain knowledge, business logic, and experience. Therefore, their productivity level may be lower as compared to their onshore counterparts. Gradually, with time their productivity improves with increase in their domain knowledge and experience. Cost management strategy need to consider this productivity gap during initial phase while estimating cost [9].

Cost estimation in GSD should consider expenses incurred in setting up as well as maintaining infrastructure, travelling, and training. Other factors that negatively influence cost are time zones involved, linguistic and cultural differences, task dispersion, design and technology newness, and low requirements comprehension. Process compliance, effective team structure, high trust, sufficient client participation, appropriate knowledge management, team cohesion, and reusability can aid in reducing the cost of GSD projects.

Ineffective management of the cost drivers can result into additional management overhead, insufficient quality, rework, slipped deadline, dissatisfied customers, fear, reduced productivity, and even project failure [8, 48]. These hidden cost drivers can invariably increase cost of development and diminish the expected cost benefits of GSD. Management should be aware of these hidden costs to avoid unrealistic expectation and support appropriate cost estimation. Metrics can be developed to assess cost on the basis of these factors. A case repository which stores effort estimation data of previous GSD projects with contextual information can support metrics [39].

4.4.3 Schedule Management

Schedule management involves processes that define activities, estimates time duration, formalize project schedule, monitor and control the schedule for timely completion [46].GSD distances introduce communication, coordination, and collaboration problems. Time zone difference and cultural divergence induces non-overlapping holidays, weekends, working hours and ethics. In GSD, time need to be reserved for travelling. Delays in GSD is caused by delayed responses due to time zone differences, rework due to requirement misunderstandings, waiting time due to mishandled dependencies, difficulties faced during cross site change requests, unawareness about remote colleagues work, confusions caused by loose work plans, incomplete handover and inadequate knowledge management. Loosely formed communication network, draining of tacit knowledge due to unplanned staffing as well as employee turnover can also cause delay. These problems cause GSD work to extend 2.5 times longer than collocated development work [23]. These delays can be prevented by effective management techniques. Effective communication, coordination and collaboration (3C) techniques can save ample amount of development time by performing round the clock development [19, 24].

4.5 Performance Management

Performance in GSD can be managed along four dimensions; process, product, communication, and coordination management as illustrated in Figure 5. Project manager should clearly specify performance criteria as well as other expectations to offshore teams during project inception to set a common vision and clear goals towards project success. A project is considered successful if it is delivered within scope, time, and cost as well as exhibit high quality and satisfies customer requirements. Other performance criteria in GSD are successful collaboration, high productivity, and improved learning. Long collaboration history increases the probability of success as offshore teams develop mutual understanding about performance expectations of the onshore team [37].

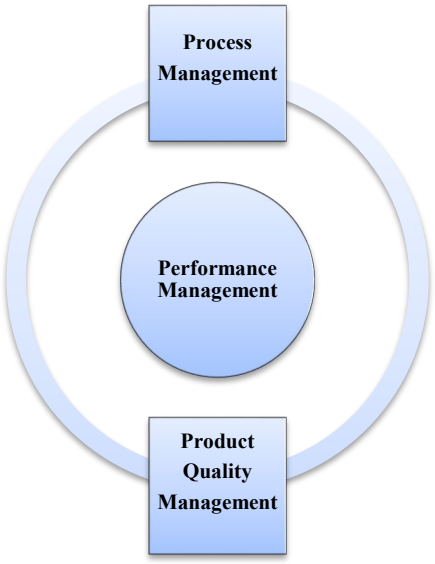

Figure 5: GSD Performance Management

4.5.1 Process Management

Process management includes methods used for monitoring, reviewing, and assessing the progress of software development activities. GSD project manager should clearly elaborate milestones, deadlines, and deliverables to whole team. He can track the project status of distributed teams through e-kanban board and several distributed agile practices such as daily scrum, weekly coordination

meetings, sprint reviews, scrums of scrum, sprint, and continuous integration [26, 25]. Burn down charts and backlogs can also be used to measure the amount of completed work against the planned work [26]. Measures related to project progress, completion status of allocated tasks to distributed members, code quality can be collected and analyzed using distributed project management tools. Distributed retrospective meetings can be used to review and assess the ongoing project work. Globally accessible repositories are used for sharing source code and documentation. Continuous integration servers can automate the code build up, analysis, and report generation to provide status feedback of the project [52]. Integration of tools with development environment helps team to share information about project progress, quality, code, and team progress. Meetings can be recorded for further reference. Project manager can monitor compliance of same process at all sites to reduce delay, defects and rework [27]. Software metrics which interweave characteristics of GSD process and project management can be developed for distributed projects [49]. Metrics used for Kanban software development, such as tracking work in progress (WIP) through cumulative-flow diagram, amount of business value delivered, due date performance according to classes of service, spectral analysis of lead time, issues and blocked work item chart, and flow efficiency can be adapted for GSD. Initial quality and failure load metrics can also be used to measure initial poor quality and amount of extra work generated due to poor quality [5].

4.5.2 Product Quality Management

Software quality can be used to measure the performance of the software product [20]. Software quality management is concerned with processes, standards, and techniques to assure and control quality of the product [46]. Several researchers have reported that GSD products exhibit lower quality than collocated products due to different distances present in GSD [12]. GSD distances causes inadequate communication, coordination breakdowns, and misunderstandings which results into increased number of defects in the product [12]. The approaches which can be used for quality management in GSD are based on prevention, appraisal, and failure [40, 7].

4.5.2.1 Prevention based approach

This approach focuses on improving quality by incorporating strategies that prevent quality problems. Frequent visits, liaison, knowledge management techniques, use of common processes as well as tools at all sites reduce misunderstandings, incompatibilities, and thus reduce the probability of defects. Trainings related to domain, process, technology, and tool usage can prevent quality problems in GSD. Collocated inception of project will imbibe common vision and goals, improve awareness about remote members and their culture and thus create a cohesive team. Globally accessible repositories allow remote members to access relevant project documents from anywhere, anytime and resolve their confusions [25]. In case of offshore outsourcing, careful vendor selection on the basis of CMMI level as well as experience, domain expertise, and professional certifications of offshore members can prevent quality problems [45]. Distribution of developers across sites must be evenly balanced to reduce the probability of defects [12]. Syntactic dependencies among architectural components as well as logical dependencies need to be carefully detected and managed [11]. Risk analysis during early phases of software development lifecycle can identify the probable risks and can prevent quality problems [25].

4.5.2.2 Appraisal based approach

It emphasizes activities in which progress, performance, and quality of intermediate artifacts are proactively assessed [40]. Cross-site design review, code review, distributed daily scrums, weekly coordination meetings, retrospectives, demo of completed functionality, electronic kanban board, usage of 3C and quality management tools constitute appraisal based quality management approach [25].

4.5.2.3 Failure based approach

It verifies the compliance of product against customer specifications and involves defect detection and correction activities. Test driven development at distributed sites, continuous integration, quality assurance, user acceptance and various testing techniques are failure based approaches for GSD quality management [25].

4.6 GSD Integration Management

The knowledge area integrates and coordinates the activities of all the remaining knowledge areas throughout project life cycle. It involves planning, managing and adapting project plans according to the circumstances. It encompasses managing communication and coordination between remote teams as shown in Figure 6.

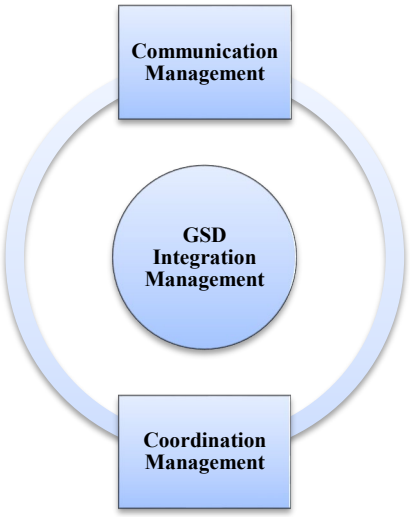

Figure 6: GSD Integration Management

4.6.1 Communication Management

Effective communication management is a prerequisite to institute a high performance GSD team as it enhances trust, interpersonal relations, and cultural awareness [15]. Usage of communication tools such as large screen electronic-smart board supported by a full time voice over internet protocol (VOIP) facilitate distributed planning meetings, cooperative design discussions, knowledge and vision sharing within distributed members. These tools reduce the impact of geographical distance, enhance team cohesion, and thus act as a powerful collaborative problem solving tool. Communication protocols, frequency, tools, and mechanism need to be initially planned, continuously monitored, and adapted according to the need [27]. Unplanned and over communication can also distract the team members from their work. Several practices such as member rotation, visits, liaison officer, and collocated inception can help to improve awareness, cultural understanding, communication, and establish trust.

4.6.2 Coordination Management

Coordination management in GSD involves managing the set of interdependent activities, tasks, and artifacts across different locations without distressing development productivity. Inadequate communication due to GSD distances can disrupt coordination and can lead to integration problems [13].Technical dependencies need to be detected as soon as possible for effective task allocation. Architecture should be used to coordinate teams efficiently. Therefore, tasks allocation in GSD needs to be congruent to distribution in order to handle dependencies effectively. Change history data can also be used to figure out coordination requirements [22]. Updated architectural documentation should be kept in globally accessible repositories. Effective collaboration between distributed teams through visits, trust, common processes and tools, cultural as well as task awareness and respect for each other, shared view of project activities; adequate

communication and knowledge management can create harmony and improve coordination between distributed teams. Continuous integration, distributed meetings, design reviews, and appropriate organizational structure can maintain required coordination between remote teams.

5. Conclusion

With the advent of high speed, cheap and reliable communication networks; most of the software companies have started distributing their development activities. However, almost 40% of GSD projects were unsuccessful in delivering the expected benefits. High failure rate clearly demarcate the insufficiency of effectively managing GSD projects. In this paper, we have first identified project management challenges faced in communication, coordination, collaboration, and control processes due to geographical, temporal, socio-cultural, and organizational distances. Subsequently, we have proposed a project management framework for GSD projects which can either eliminate or at least reduce these problems. This framework characterizes the aspects needed to be considered during GSD project management. It would be helpful to researchers as well as practitioners as it has integrated the aspects of traditional PMBOK with the aspects needed for successful GSD management.

6. REFERENCES

[1] Ågerfalk, P. J., and Fitzgerald, B. 2006. Flexible and Distributed Software Processes: Old Petunias in New Bowls? *Commun. ACM.* 49, 10 (Oct 2006). 26-34. DOI=http://dx.doi.org/10.1145/1164394.1164416.

[2] Ó Conchúir, E., Ågerfalk, P.J., Olsson, H.H., and Fitzgerald, B. 2009. Global software development: where are the benefits? Commun. ACM. 52, 8 (Aug. 2009), 127-131. DOI=http://doi.acm.org/10.1145/1536616.1536648.

[3] Špundak, M. 2014. Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? *In Procedia - Social and Behavioral Sciences*. 119 (2014), 939-948, DOI= https://doi.org/10.1016/j.sbspro.2014.03.105.

[4] Anantatmula, V. and Thomas, M. 2010. Managing global projects: A structured approach for better performance. *Proj Mgmt Jrnl*, 41: 60–72. doi:10.1002/pmj.20168.

[5] Anderson, D.J. 2010. Kanban: Successful Evolutionary Change for Your Technology Business, Blue Hole Press: Sequim, WA.

[6] Babar, M. A. and Zahedi, M. 2012. *Global Software Development: A Review of the State-Of The-Art (2007- 2011).* IT University Technical Report Series. IT University of Copenhagen, 2012.

[7] Boehm, B. W. 1981. *Software engineering economics*. Prentice Hall, Upper Saddle River, NJ.

[8] Britto, R., Freitas, V., Mendes, E., Usman, M. 2014. Effort Estimation in Global Software Development: A Systematic Literature Review. In *Proceedings of the 9th International Conference on Global Software Engineering (*Shanghai, China, August 18-21, 2016*),* ICGSE'14. IEEE Computer Society, Washington, DC, USA, 18-21. DOI=http://dx.doi.org/10.1109/ICGSE.2014.11.

[9] Casey, V. 2010. Virtual software team project management. *J. Braz Comput Soc*, 16, 2 (2010) 83-96. DOI=https://doi.org/10.1007/s13173-010-0013-3.

[10] Casey V. and Richardson, I. 2006. Uncovering the reality within virtual software teams. In *Proceedings of the 2006 international workshop on Global software development for the practitioner* (GSD '06). ACM, New York, NY, USA, 66-72. DOI=http://dx.doi.org/10.1145/1138506.1138523.

[11] Cataldo, M. and Nambiar, S. 2012. The impact of geographic distribution and the nature of technical coupling on the quality of global software development projects. *J. Softw. Evol. and Proc.*, 24, 2, 153-168. DOI=10.1002/smr.477.

[12] Cataldo, M. and Nambiar, S. 2009. On the relationship between process maturity and geographic distribution: an empirical analysis of their impact on software quality. In *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (Amsterdam, Netherlands, August 24 - 28, 2009) ESEC/FSE '09. ACM, New York, NY, USA, 101-110. DOI=http://doi.acm.org/10.1145/1595696.1595714.

[13] Cataldo, M., Bass, M., Herbsleb, J.D., and Bass, L. 2007. On Coordination Mechanisms in Global Software Development, In *Proceedings of the 2nd International Conference on Global Software Engineering* (Munich, Germany, Aug. 27-30, 2007), ICGSE '07. IEEE Computer Society, Washington, DC, USA, 71-80. DOI=https://doi.org/10.1109/ICGSE.2007.33.

[14] Chow, T. and Cao, D.B. 2008. A survey study of critical success factors in agile software projects. *J. Syst. Software*, 81, 6, (Jun. 2008) 961-971. DOI=https://doi.org/10.1016/j.jss.2007.08.020.

[15] Daim, T.U., Ha, A., Reutiman, S., Hughes, B., Pathak, U., Bynum, W. and Bhatla, A. 2012. Exploring the communication breakdown in global virtual teams. *Int. J. Proj. Manag.* 30, 2, 199-212. DOI=https://doi.org/10.1016/j.ijproman.2011.06.004.

[16] Dingsøyr, T., and Smite, D. 2014. Managing Knowledge in Global Software Development Projects, IEEE IT Professional, 16, (2014) 22-29, DOI= http://doi.ieeecomputersociety.org/10.1109/MITP.2013.19.

[17] Earl, M. 2001. Knowledge management strategies: Toward a taxonomy. *J. Manage. Inf. Syst.*, 18, 1 (May 2001), 215-233.

[18] Ebert, C., Kuhrmann, M. and Prikladnicki, R. Global software engineering: evolution and trends. In *Proceedings of the 2016 11th International Conference on Global Software Engineering (*Irvine, CA, USA, August 2-5, 2016*),* ICGSE'16. IEEE Computer Society, Washington, DC, USA, 144-153. DOI=10.1109/ICGSE.2016.19.

[19] Espinosa, J. A. and Carmel, E. 2003. The impact of time separation on coordination in global software teams: a conceptual foundation. *Softw. Process: Improve. Pract.* 8, 249–266. DOI=10.1002/spip.185.

[20] Gopal, A., Espinosa, J.A., Gosain, S. and Darcy, D.P. 2011. Coordination and performance in global software service delivery: The vendor's perspective. *IEEE T. Eng. Manage.* 58, 4, (Nov. 2011), 772-785. DOI= 10.1109/TEM.2010.2096821.

[21] Hallikainen, M. 2011. August. Experiences on agile seating, facilities and solutions: multisite environment. In *Proceedings of the 6th International Conference on Global Software Engineering* (Helsinki, Finland, Aug. 15-18, 2011) ICGSE'11. IEEE Computer Society, Washington, DC, USA, 119-123.

[22] Herbsleb, J.D. 2007. Global Software Engineering: The Future of Socio-technical Coordination. In *Proceedings of the Future of Software Engineering* (Minneapolis, MN, USA, May, 23-25, 2007). *FOSE '07*, IEEE Computer Society, Washington, DC, USA, 188-198. DOI=http://dx.doi.org/10.1109/FOSE.2007.11.

[23] Herbsleb, J.D., Mockus, A., Finholt, T.A., and Grinter, R.E., 2001. An empirical study of global software development: Distance and Speed. In *Proceedings of the 23rd International Conference on Software Engineering* (Toronto, Ontario, Canada, May 12 - 19, 2001). ICSE'01. IEEE Computer Society, Washington, DC, USA, 81-90.

[24] Herbsleb, J.D., Mockus, A., Finholt, T.A. and Grinter, R.E. 2000. Distance, dependencies, and delay in a global collaboration. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work* (Philadelphia, Pennsylvania, USA, Dec. 2000). CSCW '00. ACM, New York, NY, USA, 319-328. DOI=http://dx.doi.org/10.1145/358916.359003.

[25] Jain, R. and Suman, U. 2017. An Adaptive Agile Process Model for Global Software Development. *International Journal on Computer Science and Engineering (IJCSE). 9, 6 (Jun. 2017), 436-445.*

[26] Jain R. and Suman U. 2016. Effectiveness of Agile practices in Global software development. *International Journal of Grid and Distributed Computing.* 9, 10 (Oct. 2016), 231-248.

[27] Jain, R., and Suman U. 2015. A systematic literature review on global software development life cycle. *SIGSOFT Softw. Eng. Notes* 40, 2 (Apr. 2015), 1-14. DOI=http://dx.doi.org/10.1145/2735399.2735408.

[28] Junior, I.H.d.F., Duarte, L., Oliveira, J.P.N.d., Dantas, A. R.N., Barbosa, J.F. and Moura, H.P. d. 2012. Motivational Factors for Distributed Software Development Teams. In *Proceedings of the* 2012 7*th International Conference on Global Software Engineering Workshops* (Porto Alegre, Brazil, Aug. 27-30, 2012), ICGSEW'12 IEEE Computer Society, Washington, DC, USA, 49-54. DOI= 10.1109/ICGSEW.2012.17.

[29] Kamaruddin, N.K., Arshad, N.H. and Mohamed, A. 2012. Chaos issues on communication in agile global software development. In *Proceedings of the Business Engineering and Industrial Applications Colloquium* (TBD Kuala Lumpur, Malaysia, Apr. 7-8, 2012) BEIAC'12. IEEE, 394-398. DOI= http://dx.doi.org/10.1109/BEIAC.2012.6226091.

[30] Lamersdorf, A., Munch, J., and Rombach, D. 2009. A Survey on the State of the Practice in Distributed Software Development: Criteria for Task Allocation. In *Proceedings of the 4th International Conference on Global Software Engineering* (Limerick, Ireland, Jul. 13-16, 2009). ICGSE'09. IEEE Computer Society, Washington, DC, USA, 41-50. DOI= 10.1109/ICGSE.2009.12.

[31] Lee, S. and Yong, HS. 2010. Distributed agile: project management in a global environment. *Empirical Softw. Eng.* 15, 2 (Apr. 2010), 204-217. DOI=http://dx.doi.org/10.1007/s10664-009-9119-7.

[32] Mahmood, S., Anwer, S., Niazi, M., Alshayeb, M., Richardson, I. 2017. Key factors that influence task allocation in global software development. *Inform. Software Tech.* 91, C, (Nov. 2017), 102-122, DOI=https://doi.org/10.1016/j.infsof.2017.06.009.

[33] Mohapatra, P., Bjorndal, P. and Smiley, K. 2010. Causal analysis of factors governing collaboration in global software development teams. In *Proceedings of the 2010 5th International Conference on Global Software Engineering* (Princeton, NJ, USA, August 23- 26, 2010), ICGSE'10. IEEE Computer Society, Washington, DC, USA, 128-132.

[34] Niazi, M., Mahmood, S., Alshayeb, M., Qureshi, A.M., Faisal, K. and Cerpa, N. 2016. Toward successful project management in global software development. *Int. J. Proj. Manag.* 34, 8, (Nov. 2016), 1553-1567. DOI= https://doi.org/10.1016/j.ijproman.2016.08.008.

[35] Niazi, M., Mahmood, S., Alshayeb, M., Riaz, M.R., Faisal K., Cerpa, N.,Khan, S. U., Richardson, I. 2016. Challenges of project management in global software development: A client-vendor analysis. *Inform Software Tech.* 80 (2016), 1-19. DOI=https://doi.org/10.1016/j.infsof.2016.08.002.

[36] Nuells, N. 2013. Effective Project Management of Virtual Teams. PM World Journal. II, XI (Nov. 2013).

[37] Poikolainen, T. and Paananen, J. 2007. Performance criteria in inter-organizational global software development projects. In *Proceedings of the 2nd International Conference on Global Software Engineering* (Munich, Germany, Aug. 27-30, 2007), ICGSE '07. IEEE Computer Society, Washington, DC, USA, 60-70. DOI=https://doi.org/10.1109/ICGSE.2007.35.

[38] Ralyte, J., Lamielle, X., Arni-Bloch, N. and Leonard, M. 2008. A framework for supporting management in distributed information systems development. In *Proceedings of the 2nd International Conference on Research Challenges in Information Science.* (Marrakech, Morocco, June 3-6, 2008), 381-392. DOI= 10.1109/RCIS.2008.4632128.

[39] Ramasubbu, N. and Balan, R.K. 2012. Overcoming the challenges in cost estimation for distributed software projects. In *Proceedings of the 34th International Conference on Software Engineering* (Zurich, Switzerland, Jun. 2-9, 2012), IEEE Press, Piscataway, NJ, USA, 91-101.

[40] Ramasubbu, N. and Balan, R.K. 2007. Globally distributed software development project performance: an empirical analysis. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. (Cavat near Dubrovnik, Croatia- Sept. 3 - 7, 2007). ESEC-FSE '07, ACM, New York, NY, USA, 125-134. DOI=http://dx.doi.org/10.1145/1287624.1287643.

[41] Ramasubbu, N., Krishnan, M.S. and Kompalli, P. 2005. Leveraging global resources: A process maturity framework for managing distributed development. IEEE *Softw.* 22, 3 (May 2005), 80-86. DOI=http://dx.doi.org/10.1109/MS.2005.69.

[42] Ramesh, B., Cao, L., Mohan, K. and Xu, P. 2006. Can distributed software development be agile? *Commun. ACM*, 49, 10, (Oct. 2006), 41-46. DOI=http://dx.doi.org/10.1145/1164394.1164418.

[43] Razzak, M.A., 2015, July. Knowledge Management in Globally Distributed Agile Projects--Lesson Learned. In *Proceedings of the 10th International Conference on Global Software Engineering* (Ciudad Real, Spain, Jul. 14-16, 2015*),* ICGSE'15. IEEE Computer Society, Washington, DC, USA 81-89. DOI= 10.1109/ICGSE.2015.22.

[44] Richardson, I., Casey, V., Mccaffery, F., Burton, J. and Beecham, S., 2012. A process framework for global software engineering teams. *Inform. Software Tech.* 54, 11, (Nov. 2012), 1175-1191. DOI=http://dx.doi.org/10.1016/j.infsof.2012.05.002.

[45] Schwaig, K. S., Stephen H. G., and Leeds, E. 2006. Project management issues in IT offshore outsourcing. *International Journal of e-Collaboration* 2, 4 (2006), 53-73.

[46] Schwalbe, K. 2015. Information technology project management. Ed. 8th. Cengage Learning.

[47] Silva, F. Q. B. da, Costa, C., França, A.C.C., Prikladnicki, R. 2010. Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review. In *Proceedings of the 2010 5th International Conference on Global Software Engineering* (Princeton, NJ, USA, August 23- 26, 2010), ICGSE'10. IEEE Computer Society, Washington, DC, USA, 87-96. DOI=http://dx.doi.org/10.1109/ICGSE.2010.18.

[48] Smite, D., Calefato, F., Wohlin, C. 2015. Cost Savings in Global Software Engineering: Where's the Evidence? *IEEE Software*, 32, 4, (July-Aug. 2015), 26-32. DOI=10.1109/MS.2015.102.

[49] Tihinen, M., Kommeren, R. and Rotherham, J. 2012. Metrics and measurements in global software development. *International Journal on Advances in Software*, 5, 3 and 4, (2012), 278-292.

[50] Tjørnehøj, G., Balogh, M.B., Iversen C., and Sørensen S. 2014. Designing Project Management for Global Software Development. *In: Bergvall-Kåreborn B., Nielsen P.A. (eds) Creating Value for All Through IT. TDIT 2014. IFIP Advances in Information and Communication Technology,* 429. Springer, Berlin, Heidelberg.

[51] Verner, J.M., Brereton, O.P., Kitchenham, B.A., Turner, M., and Niazi, M. 2014. Risks and Risk Mitigation in Global Software Development: A Tertiary Study. *Inf. Softw. Technol.* 56, 1 (January 2014), 54-78. DOI= http://dx.doi.org/10.1016/j.infsof.2013.06.005.

[52] Yagüe, A., Garbajosa, J., Díaz, J. and González, E. 2016. An exploratory study in communication in Agile Global Software Development. *Comput. Stand. Interfaces*, 48, C (Nov. 2016), 184-197. DOI= http://dx.doi.org/10.1016/j.csi.2016.06.002.

[53] Yu, L., and Mishra, A. 2010. Risk Analysis of Global Software Development and Proposed Solutions. Automatika. 51, 1 (2010) 89-98.