Proposed Development Framework and Assessing Performance Model in Distributed Software Engineering Environment

نموذج مقترح لتطوير وتقيم الأداء في بيئة هندسة البرمجيات الموزعه

Prepared By:

Mustafa Saad Jeryo

Supervised by:

Dr. Akram Othman AL- Mashaykhi

This thesis is submitted in partial fulfillment of the Requirements for the Degree of Master in Computer Science.

Department of Computer Science

College of Computer Sciences and Informatics

Amman Arab University

2013

نموذج مقترح لتطوير وتقيم الأداء في بيئة هندسة البرمجيات الموزعة

إعداد

مصطفى سعد كاظم جريو

إشراف

الدكتور أكرم عثمان المشايخي

قدمت هذه الرسالة لقسم الحاسوب ، كلية العلوم الحاسوبية والمعلوماتية في جامعة عمان العربية ، لاستكمال متطلبات الحصول على درجة الماجستير في علم الحاسوب .

قسم علم الحاسوب

كلية العلوم الحاسوبيه والمعلوماتية

جامعة عمان العربية

2013

Authorization

I'm Mustafa Saad Jeryo , authorize Amman Arab University the right to provide copies of the dissertation to libraries , institutes , agencies or individual when necessary.

Name : Mustafa Saad Jeryo

Signature : 

Date : 11.12.2013

The Discussion Committee Decision

This dissertation entitled **" Proposed Development Framework and Assessing Performance Model in Distributed Software Engineering Environment "** was discussed and passed on November 26, 2013.

Discussion Committee Member	Title	Signature
Dr. Osama Al- Musleh	Chair	
Dr. Akram Al- Mashaikhi	Member and Supervisor	
Dr. Mohamed Nassar	Member	

Dedication

This dissertation is dedicated to my dear uncle Prof .Basim Jrew , my beloved wife , my baby , my father , my mother , my sisters , and my friend .

Acknowledgements

All praise is due to Allah for the bless that he gave .Without whose help, I would not have reached this far .

First and foremost , I am heartily thankful to my supervisor , Assistant Prof Dr. Akram Othman AL- Mashaykhi , whose encouragement , guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I am very lucky to have him as a supervisor as he is an experience lecturer .

I would like to thank the member of the assessment committee Dr. Osama Al- Musleh , and Dr Mohamed Nassar .

Special thanks to my dear uncle Prof Dr. Basim Jrew for his cooperation and guidance during my study .

Special thanks to my lovely wife Zainab Al-Hamadany ,my pretty daughter Fatima , my Father , Mother and Sisters for their prayers and for their loves , support and continuous encouragement during the period of my study .

Finally ,I would really like to extend my deep gratitude and sincere thanks to all my friends and colleagues who were a great help in the completion of this work .

Table of Contents

List of tables

List of figures

Abbreviations

Abbreviations	Meaning
ALM	Application Lifecycle Management
CCCM	Cooperation, collaboration and coordination model
DSDM	Dynamic System Development Method
DSE	Distributed Software Engineering
DSD	Distributed Software Development
GDSD	Globally Distributed Software Development
GSD	Global Software Development
GDT	Geographically Distributed Team
GDSDT	Globally Distributed Software Development Team
GSDF	Global Software Development Framework
GSDPAM	GSD Performance Assessment Model
IRC	Internet Relay Chat
IM	Instant Messaging
IDEs	Integrated Development Environments
PM	Project Management
RUP	IBM Rational Unified Process
RE	Requirements Engineering
SEDLC	Software Engineering Development Life Cycle
SED	Software Engineering Development
SRS	Software Requirements Specification
SW	Software
SCM	Software Configuration Management
SDVT	Software Development Virtual Team
SRRM	Satisfaction, recognition and reward model
TDD	Test Driven Development
VoIP	Voice over Internet Protocol
VCM	Value creation model
VST	Virtual Scrum Team
XP	Extreme Programming

Proposed Development Framework and Assessing Performance Model in Distributed Software Engineering Environment

Abstract

Global software development (GSD) is one of the contemporary trends in software engineering discipline. It is a result of development of information, communication technology, and raising what so called cyberspace.

As global common, mature and reliable environment as infrastructure, platform, software systems, application regulation, security measurement, and governance space for telecommunication function as global environment for human activities such as the hypotheses of mobilization and virtualization.

Global Software Development approach faces several problems, and challenges. Opportunities to these technological problems and non-technological problems are related to diversity in cultures, communication patterns and different time zone.

The research problem of this thesis attempts to answer questions related to the framework, models, methodology, and tools that can bridge the technical and non-technical gap. Constraints and limitations of this study and inspected a group of methodologies and tools by review, critic, analysis of adapted models in management and performance evaluation of GSD projects.

This thesis was accomplished through proposing general framework with four models that can be adapted to ensure the effective performance during the GSD phases from requirements engineering to the implementation as follows:

The proposed General framework contains four components. Agile software as development methodology, cloud computation model as infrastructure and platform development environment,communication media as productivity tools and techniques, and performance assessment model .Therefore, the proposed model for assessing the performance of GSD contains two sub models such as independent variables. Cooperation, Collaboration and Coordination Model (CCCM), Satisfaction, recognition, and reward model (SRRM). The last model, Value Creation Model (VCM) through Disruptive Technology (Innovation) is proposed as dependent variable.

This thesis also presents the major conclusion and future research work.

Arabic summary

نموذج مقترح لتطوير وتقيم الأداء في بيئة هندسة البرمجيات الموزعة

الملخص

التطوير الموزع للبرمجيات من الاتجاهات الحديثه في مجال هندسة البرمجيات نتيجة للتطور الحاصل في تكنولوجيا المعلومات والاتصالات وظهور ما يعرف بالفضاء الالكتروني كبيئة عالميه مشتركه لممارسة مختلف الأنشطة من جانب ومن الجانب الأخر التحول في النماذج والطرائق والأدوات المساعدة مثل مفهوم الافتراضية Virtualization ومفهوم الانتقالية mobilization .

يواجه اسلوب التطوير الموزع للبرمجيات مشكلات وتحديات البعض منها مرتبط بالجانب التكنولوجي والبعض الاخر مرتبط بالمحددات الثقافيه وأنماط الاتصال والتواصل واختلاف الوقت في المواقع الجغرافيه المتباعدة.

مشكلة البحث الذي تتعرض لها الرسالة تتعلق في الإجابة عن مجموعه من التساؤلات حول الإطار والنموذج والمنهجية والأدوات الأنسب لتجاوز المحددات التكنولوجيه وغير التكنولوجيه من خلال دراسة واختبار مجموعة الطرائق والأدوات ونقد وتحليل النماذج المعتمده في ادارة وتقويم الاداء في اسلوب التطوير الموزع للبرمجيات.

تنتهي الرسالة على ضوء دراسة المشكلات والمحددات لأسلوب تطوير مشروعات هندسة البرمجيات الموزع المراكز باقتراح **اطار مفاهيمي عام لمكونات التطوير الموزع للبرمجيات** يضمن كفاءة وفاعلية الاداء خلال مراحل المشروع كافة ، بالإضافة الى اقتراح **تعديل النموذج الخاص بفحص واختبار الأداء في مشاريع التطوير الموزع للبرمجيات** الذي كان قد انجزه عدد من الباحثين في المجال ، وذلك من خلال اضافة ثلاثة نماذج تهدف الى خلق القيمة وتحفيز روح الابتكار والإنتاجية في المنتوج البرمجي عبر تحويل المحددات مثل اختلاف الثقافات واختلاف مواعيد وأماكن العمل وضعف التواصل وجها لوجه بين اعضاء الفريق الموزع المراكز الى عناصر تميز وتنسيق وتشارك في المعرفة ، اول النماذج الثلاثة هو **نموذج التنسيق** وثانيها **نموذج الرضا والعرفان والعائد** كمتغيرين مستقلين في نموذج فحص الأداء، وثالثهما **نموذج خلق القيمة** في المنتوج البرمجي كمتغير تابع في نموذج فحص الاداء . هذا بالإضافة إلى عرض أهم الاستنتاج ومقترحات العمل المستقبلي.

Chapter One

Introduction

1.1 Overview

The Software Engineering Development Life Cycle (SEDLC) is a framework that defines tasks performed at every stage in the software development process. SEDLC is a structure followed by development team software within the organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. This life cycle defines a methodology for improving the quality of software, control the process of developing an information system and the overall development process.

There are several methods of software development process followed by various organizations: waterfall model , prototype model , incremental model , iterative development , spiral model , rapid application development , RUP (IBM Rational Unified Process) , Agile software development , scrum development , XP (Extreme Programming) , TDD (Test Driven Development) , FDD (Feature – Driven Development) , DSDM (Dynamic System Development Method) , V- model (software development) .

All of these methodologies have activities and steps such as, Requirement, Specification, Architecture, construction, design, testing, debugging, deployment, and maintenance.

The Application Lifecycle Management (ALM) coordinates the life-cycle activities of software development, including requirements

management, and architecture management, change and software configuration management, build and deploy management and quality management [22].

1.2 Distributed Team for Global Software Development (GSD)

Today's software projects are often distributed and developed across multi geographic locations because of the Globally Distributed Software Development (GDSD) which becomes very important issue for organizations currently with the increase the tendency towards globalization and global outsourcing.

This distribution poses new challenges produced by cooperation across different countries, different times zones, different cultures, different languages, reduce level of trust factors, communication across temporal, spatial distances, lack awareness of contextual shared, project management, risk management and coordination is important in software development because it leads to benefits such as cost saving, shorter development cycles, and better-integrated products but this perspective has only been investigated in real-time collocated tasks and we know few about which types of team knowledge best help coordination in the most geographically distributed software work and the main reason of distributing software

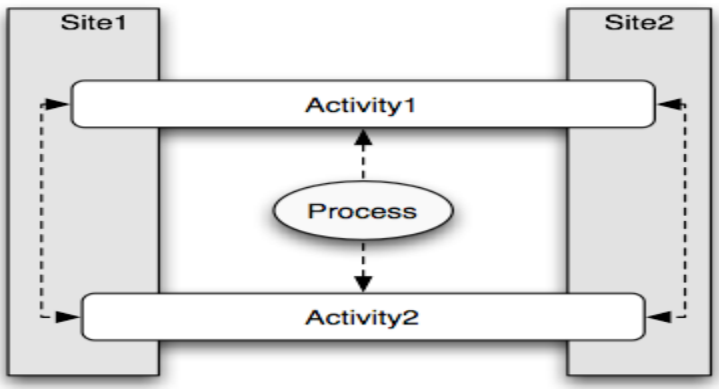


Figure (1-1): Vertical process model of distributed software development [18]

Development is to improve time-to-market by round the- clock development or to increase flexibility and take advantage on merger and acquisition opportunities [5, 17,39].

Other reasons include the access to cheaper labor, increasing knowledge of customers and local conditions by market, or take advantage on the global talent pool. In fact, shortage of highly skilled science and talented engineering, more generally, needs for the access to qualified personnel are important explanatory factors for off-shoring innovation decisions, in line with these trends, distributed software development is now no longer only an option for most enterprises; rather, it is a business need [5].

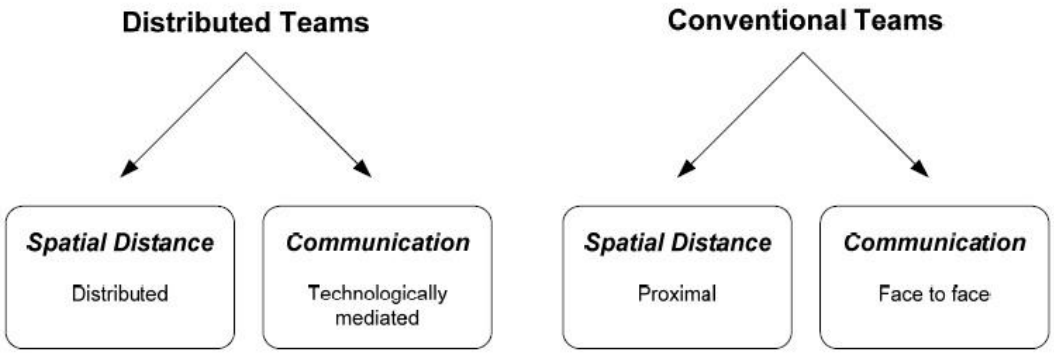


Figure (1-2). Distributed versus conventional teams.[19]

1.3 Management of Distributed Team

Management of GDSD are projects that consist of two or more teams are working together to accomplish project goals from different geographical locations. In addition to geographical dispersion, global distribution teams face time-zone and cultural differences that are not limited to different language, national traditions, values and norms behavior [6].

A team development environment usually needs support such as functions version management, build management, project management, parallel development, team coordination, and process management. Besides sharing codes and documents, a team development environment should further enable team members to share technical skills with each other and the main challenge when the member change of a development team causes different experienced members to cooperate with each other in the same development process these new challenges require a distributed team development environment to support the cognitive cooperation [7].

Technological improvements in the fields of communication and networking make distributed group work more feasible to increase quality and availability of this technology makes distributed software engineering efforts more feasible option for organizations. Software development projects, which utilize distributed teams, as members of the project group are geographically distributed among different physical sites.

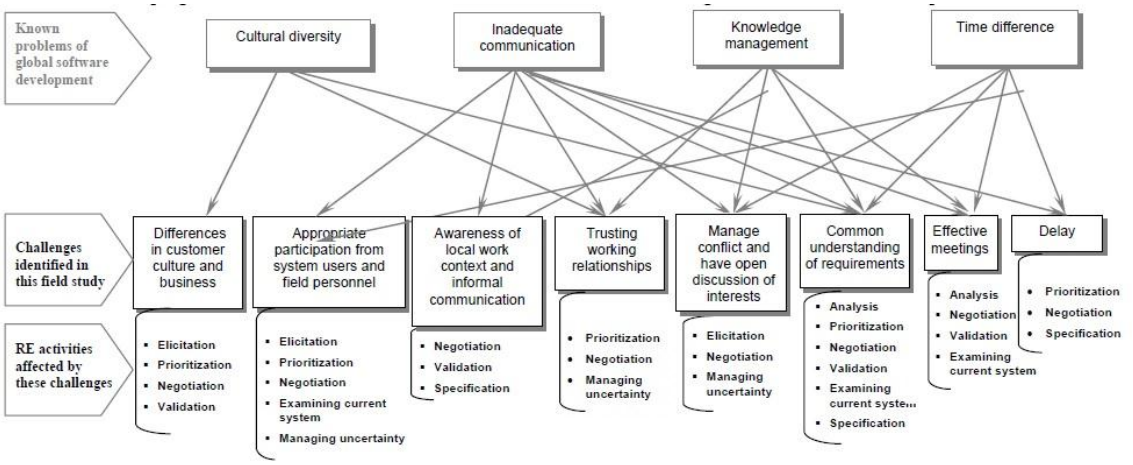
1.4 Requirement Engineering in Globally Distributed environment

Requirements Engineering (RE) plays a very important role in software development. For some years now, it has been recognized that problems associated with requirements engineering are the major reasons for software project failures where the end product does not meet the real needs of the problem owners, the steps or activities of the requirements engineering are: requirement specification, requirement elicitation, requirement analysis and negotiation, requirement validation and requirement management [1,2] .

RE in globally distributed environment becomes critical, due to characteristics of the distributed development (physical and temporal distance, cultural differences, trust, communication, coordination and control, etc.). RE is also the most expensive activity communication of software development since it is at this phase of development that most of the interactions occur between the problem owners and problem solvers. Geographical distance between development sites has a direct impact on all forms of communication, this communication is usually less frequent and more constrained [6].

Communication problem at the beginning of the project leads to bad distribution of activities between teams and also lead to bad planning in the beginning of the project.

Figure (1-3). A model of impact and the affected RE activities due to problems of cultural



diversity, inadequate communication, knowledge management and time differences in global software development [11]

1.5 Using Scrum in Global Distributed Software

To explain one of these methods Scrum development model: Scrum is an agile method; it is an iterative process and aims to improve the development speed and product quality using different methods. Some of the main benefits of Scrum are that it like other agile processes are good for rapid changing requirements and doesn't allows the customer to see the result of the development after each sprint (Each development cycle in Scrum is called a sprint).

If the software project has several distributed teams that all have their daily Scrum meetings, then there are several tools are used to communicate these teams, such as teleconference, videoconference, chat and web cameras had to be set up only once. If video or teleconferencing is not possible because of technological or other issues, such as problems with spoken language, then using chat is an option to some people who might feel more comfortable

with writing instead of speaking due to (subjective) difficulties with pronunciation or understanding spoken foreign language one of our case

Projects used Internet Relay Chat (IRC) for arranging daily Scrum meetings. The daily interviewees reported that distributed Scrum meetings were the most useful practice for distributed projects. The benefits of daily Scrums are numerous. They provide frequent possibilities to share information and coordinate work between distributed team members. They help to recognize possible problems early on. They also provide a possibility to create contacts, and they encourage team members from different sites to communicate more actively, also facilitating off-line communication after the meetings [22].

Even though there are several benefits in arranging distributed daily Scrum meetings, but there are also some challenges. The biggest challenge for distributed teams is the same as for co-located teams: understanding what the correct amount of information to report in a daily scrum meeting is. This is challenging even in a co-located project, but in a distributed project it is even more difficult and due to cultural differences may have big impact on what people find appropriate to report in a daily scrum meeting [23]. As referred to it is at the proposed general framework of GSD mentioned in chapter four.

1.6 The Problem Statement

With the release of globalization, a major improvements in infrastructure telecommunications in many parts of the world is much easier to acomplaish the projects by people working together from different plase. This enabled organizations to exploit time differences, the availability of skills, without mention the cost differences to get projects done faster, better, and cheaper.

The purpose of this study is to asses some useful techniques responds to the challenges faceing the domination, control and quility assuerance of the software project requierments engineering process is developed in the two main ways of project teams geographicaly distrbuted : the first, is when a big project is divided broken into many smaller projects and each smaller project is done by a team in a certain location.

The seconed when the team is doing a project that each of its members are in separated places.

Working with virtual teams show new challenges to the leaders and the managers. Time zones difference, culture, communications lags, such varitiations create problems in making teams feel connected and enthused about their work. Team leaders and managers need to work harder in such settings to overcome these issues.

The research study will attempt to answer the following questions:

Question one: why do we use the software engineering distributed development?

Question two: what are the problems of adapting the SE distributed development?

Question three: what are the suitable methodologies and tools for SE distributed development?

Question four: what are the problems facing the scale and adaption of agile approach for the SE distributed development mainly in Scrum team and sprint process design and pair programming?

Question five: what are the suitable means, methods and technological tools for daily work review, assuring the Quality of each success releases and the final product?

Question Six: what are the obstacles and other problems facing the development in virtual environments?

Question Seven: How are distributed projects are organized in terms of software engineering Process patterns?

Question eight: Are there successful or unsuccessful process patterns for Requirement Management?

1.7 Research Methodology

This research will use the empirical study to fulfill its goals and objectives.

In Software Engineering Development (SED) there are many models that can be used to develop software product such as waterfall model , prototype model , incremental model , iterative development , spiral model , rapid application development , RUP (IBM Rational Unified Process) , Agile software development , scrum development , XP (Extreme Programming) , TDD (Test Driven Development) , FDD (Feature – Driven Development) , DSDM (Dynamic System Development Method) , V- model (software development) .

This research will examine the tools and methodologies which support distribution in agile software development model.

The following steps will guide the work :

1. Inspect the factors are affecting team performance of GSD.
2. Inspect the problems and challenges are facing the Application lifecycle management such as: Communication, Cultural, Configuration Management, Time zone, Coordination and Collaboration
3. Identify groupware tools which best support distribution Software Engineering environment.
4. Define the advantages and disadvantage of these tools.
5. Based upon the above, the work attempts to propose a framework model for GSD and Application Lifecycle Management (ALM) which is suitable for distributed environment

1.8 Thesis Organization

This thesis splatted in to five chapters as the following:

Chapter one deduction for introducing the subject of global software development, motivation of the research topics and mainly the research problem examine by this work, display the study research questions, statement of the research problem, and finally presenting the methodology followed by this work.

Chapter two committed to present the background and literature review of related works, studies, and papers concerned with Distributed Software Engineering development, and its problems elaborating and focusing on different technical and non-technical problems.

Chapter three dedicated for Global Software Development GSD and Discussion of available frameworks, models, approaches and tools in managing and assessing the performance of virtual teams of such projects implementation.

Chapter four presenting the details of proposed framework models, the main contribution of this work are:

- Schematic representation of the general framework of components GSD
- Proposed Model for assessing the performance of GSD
- Cooperation, collaboration and coordination model (CCCM)
- Satisfaction, recognition and reward model (SRRM)
- Value Creation Model (VCM)

Chapter Five committed for presenting the conclusions and draw the trained for future work.

Chapter two

Background and related works

2.1 Distributed Software Engineering Development

The term "distributed software engineering" is not clear. Trying to cover the research are aspects of the Distributed Software Engineering (DSE). The following description: "Distributed Software Engineering includes both distributed software engineering, and possibly local, distributed software development process, such as collaborative team work". Therefore, we can distinguish between two types of definitions based on whether the worries in software product engineering or software development process. First definition, Distributed software engineering is the field that covers the issues of software products that is published in a distributed environment. Second definition, distributed software development is the process of distributing (synchronized) collaborative software development activities within a geographically dispersed group of developers [18].

Technological improvements in the areas of infrastructure communications and networks are make distribution group work more rewarding and there are several factors are contributing to the dissemination of distributed development projects, technological, technical, economical, organizational, and social as well. Increasing quality and availability of internet based technologies makes distributed software development more viable option for many organizations .

The development software project, which uses distributed team, where team members of the project group are geographically distributed across different physical location.

GDSD is a response to an increased common strategy for issues such as the availability of a skill acquisition, increasing code size, cost and complexity, and limitations of other resources. A team distributed is the team which is spread through different physical locations, but they are connected by communications technology of some sort, although the team is geographically in separate companies, they work together on the same task.

Distributed teams have been formally defined as “teams whose members are dispersed across distance and time, are linked together by some form of electronic technology, and physically interact with each other rarely” [19].

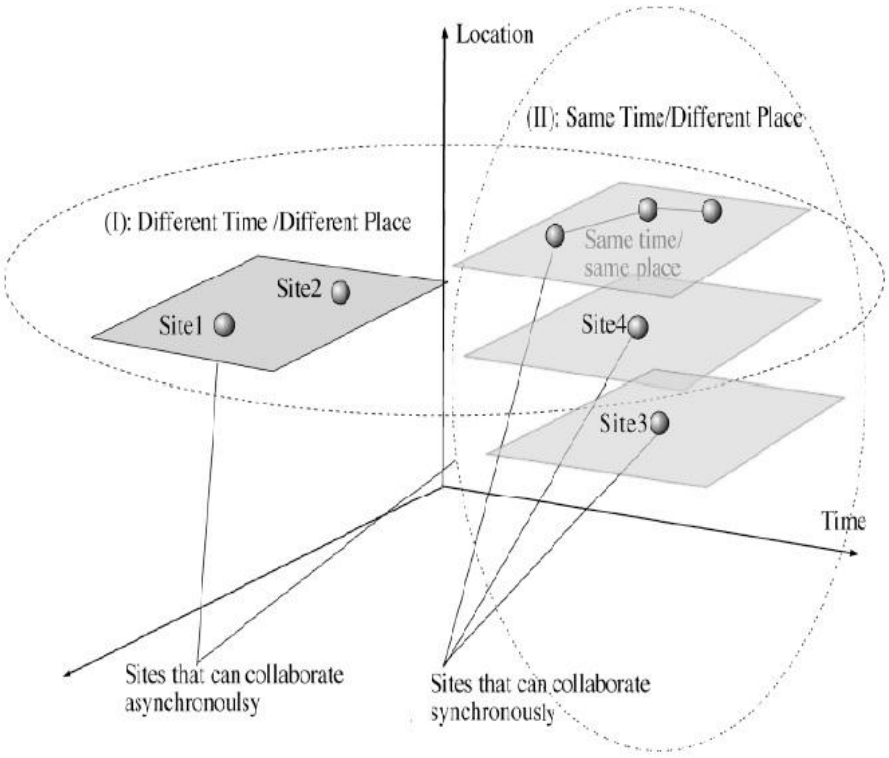


Figure (2-1) Collaboration dimensions of multiple-site development projects. The dashed ellipse (I) shows the Horizontal process model, the dashed ellipse (II) shows the Vertical process model. [18]

Distributed teams differ from traditional face-to-face teams in two main ways. First, Co-located teams typically operate at the same time, same place environment and use face-to-face meetings, whereas distributed teams do not operate in the same place and thus rely heavily on information and communications technologies including video-conference and audio-conference tools, e-mail, telephone, fax, whiteboard and the internet. Second, physical separation and depends on technology, team processes, such as communication and coordination, are more difficult to achieve for distributed teams , and product development teams globally distributed are everywhere in many industries, allowing cooperation between countries, cultures and disciplines , distributed cooperation provides the chance to reduce development and production costs and reduce cycle time, but these gains are not free of potential

differentiation. [19].

Figure 2-2 communication team in Co– located [38]

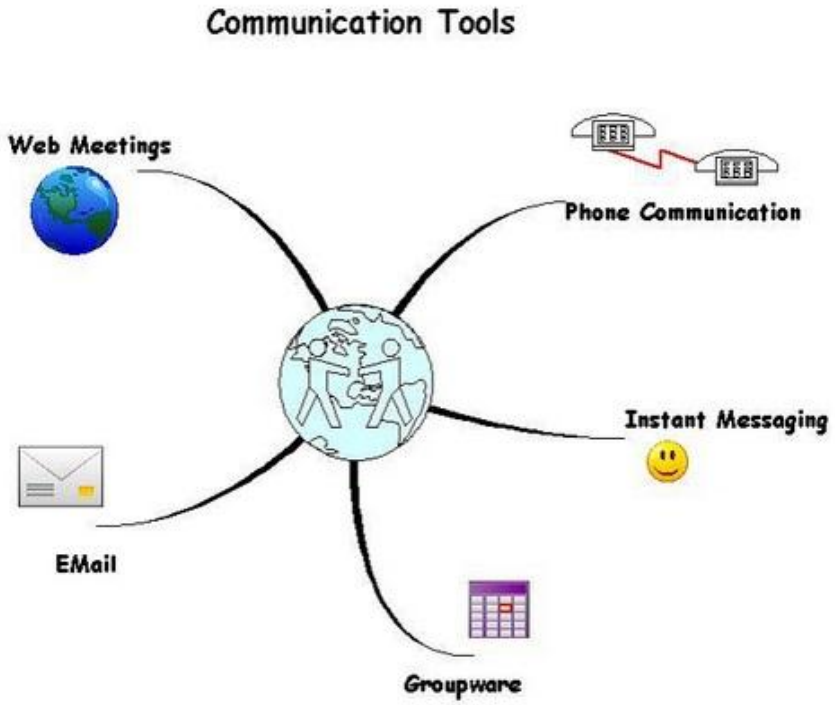


Figure 2-3 Conceptual view of global communication tools [24]

2.2 Distributed Software Engineering problems

Distributed Software Development (DSD) allows the team members to be located in different remote sites throughout the software life cycle, making a network of remote sub-teams, the distance between different teams can differ from a few meters away (when teams work in neighboring buildings) to different continents, and it is called the situation in which teams are distributed outside the boundaries of Nation Global Software Development (GSD), however, there are some problems which caused by the distance that separates the development teams. Coordination and communication become more difficult as the software components are sourced from different places, thus affecting project organization, project control, and product quality.

In this section, we assemble some problems (challenges) that impacted the distributed.

2.2.1 Communication

Distance introduces barriers to informal face-to-face communication, and the stakeholders' communication that is dependent on the quality of using synchronous or asynchronous electronic communication tools, in general, the communication is important for S\W development because communication facilitates knowledge transfer between team members and allow team member to understand the requirement from client. The software life cycle requires a great deal of communication between those members involved in the development who exchange a large amount of information through different tools and different formats without following communication standards, and who thus face misunderstandings and high response times [20]. These drawbacks, combined with the complex infrastructure and the great size of personal networks which change over time, are summarized in a decreasing in communication frequency and quality, which directly affects productivity.

2.2.2 Time Zone (Time Differences)

The large distribution of stakeholders across five continents introduces large time-zone differences and allows little overlap available for synchronous collaboration, the communication in distributed projects is difficult due to the lack of face to face meetings and synchronous communication makes the communication harder and can produce team misunderstanding, and delays in the projects. Conventional wisdom stands for the projects which are more

distributed in countries with different cultures and large time zone differences, these projects are more challenging, the main advantage of time difference time is the systems have become companies that rely on shift work easier due to the adoption of timing differences between regions.

2.2.3 Collaboration

Software development is a collaborative activity in which business analysts, customers, system engineers, architects, and developers interact with the concurrent edition of models and processes requires synchronous collaboration between architects and developers who cannot be present at common location [20].

2.2.4 Cultural

Cultural diversity is undoubtedly inevitable in a global team distribution on smaller scale, there is a work, and the workplace norms that differ greatly between the offices located in various parts of the same company, though, that there are no cultural differences among members team. There is a sharp contrast to Western values and non-Western, for example, the Western individual value, and accomplishment, equality, and informal, and assertiveness, while non -Western value of the collective, and modesty, and hierarchy, formality [21].

Cultural differences imply different terminologies which may cause mistakes in messages and translation errors. Different levels of understanding of problem domain also exist, as do different levels of knowledge, skills, and training between teams. The use of translation processes and codification guidelines is useful [20].

2.2.5 Configuration Management and Coordination

Distributed environments show problems derived from conflicts related to the source code control and the sheer quantity of information about requirements from multiple sources at remote customer sites are not appropriately shared with the developers [20]. Coordination and synchronization become more complex and difficult as the degree of distribution in multisite developments of the team grows, and traceability is a critical factor, as problems derived from communication, lack of group awareness, and the complexity of the organization appear which influence the way the work must be structured and managed and the challenge of the coordinating development projects has not gone unnoticed by the software development researchers and practitioners who have worked primarily on solutions that provide clear mechanisms and visible format. These solutions include activities such as identifying, planning, following the process, design specifications, manage carefully the requirements, measure the characteristics of the process, determine regular meetings to track the progress and implementation of the workflow system [21].

2.3 Team Management and Tools Communication in Distributed Environment

2.3.1 Team Management

Distributed team software development is kind of software development management paradigm that focuses on work cooperation and resource sharing among the geographical distributed team member during the developmental process.

Team development environment needs to support functions such as issue management, construction management, project management, parallel development, the coordination team, and process management. Besides sharing codes and documents, the team must develop environment to enables team members to exchange technical skills with each other [7].

In the [7], we present presented the concept of knowledge flow and relevant management mechanism to achieve the exchange of knowledge and cognitive cooperation in the process of software development team

which are geographically distributed and shed light on some problems such as changing the member of a development team causes different experienced members to cooperate with each other in the same developmental process.

These new challenges require a distributed team developmental environment to support the cognitive cooperation, and the use Internet based of knowledge exchange approaches: E-mail-based approach. This approach enables a team member to communicate with other team members through E-mail.

Figure 2-4 distributed team management [38]

Many are being developed by professionals projects spread over different places. Previously, it is possible to notice a significant increase in this approach, known as Distributed Software Development (DSD). This popularization of DSD is based on the expectation of: achieving lower production costs, gaining economies of scale, accessing specialized resources, reducing time-to-market, achieving higher quality, increasing proximity to customers, and accessing new markets [8].

In [9] software projects are becoming geographically distributed, with limited face-to face interaction between participants. These projects face special challenges that need careful management attention. The reason for distributing software development can be to improve time-to-market by round the- clock development or increase flexibility in capitalizing on merger and acquisition opportunities.

- Challenges that have been met in [9]

1) Geographic dispersion, which may cause coordination difficulties related to time zone differences, locally-situated knowledge, and lack of presence awareness.

2) National diversity, which can cause coordination difficulties related to different communication routines, linguistic differences, and weak interpersonal relationships.

3) Structural dynamism, which may cause coordination difficulties related to problematic task coupling, low task awareness, and problematic inter-functional conflict resolution.

4) Electronic dependence, which can cause coordination difficulties related to limitations of informal Ad-hoc communications and organizational identification.

Generally, many books have been written about team management in general and software team management in particular. For example, Covey et al. (1989), DePree (1989), Guaspari (1991), Harragan (1977), Hill (1992), for general discussions of management, for ways to manage software development.

Many principles, practices ,and advice can apply to distributed software development teams despite it is written with the traditional teams in mind as mentioned earlier, distributed teams have a traditional commonalities that should be exploited to full advantage. In both types of teams, the team leader brings concentration to the group through our creating resolutely mission of the team project.

Team leader directs and requires the use of a written specification for a program with certain protocols for establishing boundaries, this program assign each person to specific tasks in the project [24].

Telecom guidelines were developed by the team leader who promotes collaborative environment and favorable working relationships .Team leader not only manages the team members and the range of work, but also manages the hardware and system environment according to specifications of the project or other factors .In addition, the head of the management team has common tasks, such as project scheduling, labor prediction, expense reporting, and reporting time, the acquisition of the required software or hardware.

These responsibilities and roles can be seen in both traditional and distributed teams. Table 2.1 gives an overview of the similarities and differences between traditional and distributed teams that may influence their management.

Table (2-1) Comparing traits and aspects of the traditional team versus the distributed team[24].

Team manager is capable of strength and advantage in different team members. For example, in some cases, the hardware and software differences between teams and distributed widely specific hardware and operating systems, or software tools required by the project (for example, an application needs to be migrated from the Windows platform to Linux or to the operating system environments Apple OS)[24].

2.3.2 Communication Tools in Distributed Environment

Communication tools can be classified into three types: (1) synchronous, (2) asynchronous, (3) knowledge transfer.

Trait/aspect	Traditional team	Distributed team
Location	The traditional team is commonly located in same locale and often within same building.	The distributed team can traverse many different locales، time zones, and countries.
Familiarity with team members	Traditional team members often know one another and may be used to seeing one another in casual or business settings.	Members of distributed teams usually only engage one another for the sake of the project goals and may never meet one another face to face. The only contact they may have with one another is for the sake of the project.
Work environment and communication	Traditional team members work in close proximity to one another allowing for face-to-face communication. E-mails, telephones, and shared project documentation supplement the communication environment.	Face-to-face communication is rare. In-person meetings are replaced with telephone conferences and electronic white board sessions. E-mails and phone calls are likely more frequent, and project documentation can be more crucial in defining goals for the project.
Development (hardware, software, network) environment	The development environment is often set by the organization and is rigid and often cannot be easily adapted for software projects out of the ordinary.	The development environment may be as diverse as its team members. Members may be responsible for their own hardware, software, and network setup, requiring creative solutions for integration.
Project management	The manager of traditional teams has the ability to visually oversee work as it is being conducted by team members.	The manager of distributed teams must rely on review of deliverables, status reports, and other non-visual communication to oversee projects. A greater degree of trust of team members is essential.

Synchronous tools are vital or "real-time. Asynchronous tools that allow information are to be transferred or received over a period of time, which does not require synchronization.

A knowledge transfer tool can be collection of information or tool that helps in using a set of information. As referred to in the proposed is general framework of GSD mentioned in Chapter four precisely in productivity tool modules.

2.3.2.1 Synchronous Communication Tools

Synchronous communication tools for software developers are greatly the same as those for any other type of remote working group. Some forms of communication take place with all participants at the same time, such as a telephone conversation or other real interactions .This tools call this form of communication simultaneously. Here we describe how these tools can help teams to share their thinking processes and thereby increasing the quality and productivity of the program team.

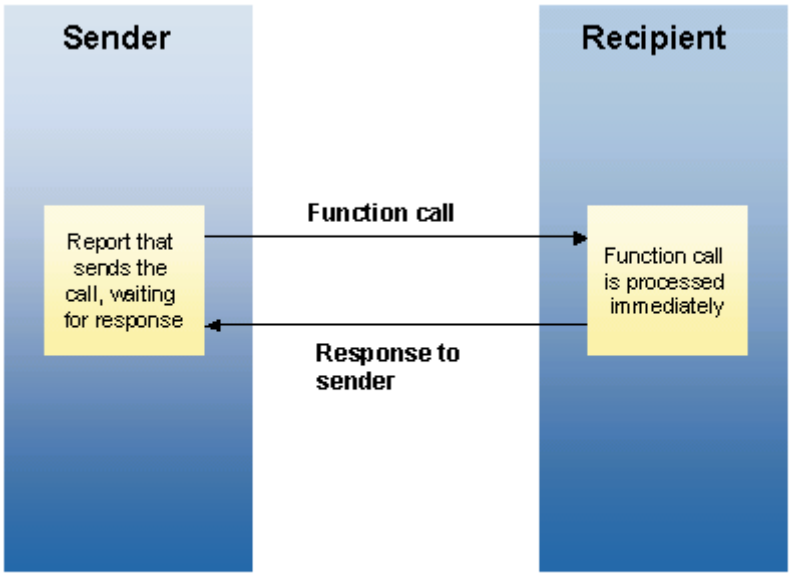


Figure 2-5 Synchronous Communication [38]

Table 2.2 provides examples of the types of synchronous tools that may be used by to such teams.

Table 2.2 Communication tools: synchronous [24]

Tool	Examples	Features	Primary advantage
Telephone	"Plain Old Telephone Service" (POTS), Voice over Internet Protocol (VoIP)	Direct calls, conference calls	Familiar to everyone, provides instant interaction
Instant messaging and chat, video chat	Yahoo Messenger, MSN Messenger, AOL Instant Messenger, Internet Relay Chat, Skype	Instant interaction, less intrusive than a phone call	Provides instant interaction, can be used in asynchronous mode if all parties remain connected, with video chat conveying facial expressions for richer communication
Web casts	NetMeeting, WebEx, Citrix, Go To Meeting, ATT Connect Meeting	Live audio, dynamic video, whiteboard, application sharing	Real-time interaction, augments speech with images and live action displays
Online translators	Google, Yahoo, various other free and commercial varieties available	Instant translation of words, paragraphs, and entire documents	Real-time translation

Some of these tools are physical hardware, but most are software. Also, most of who rely on an Internet connection. Global distributed teams are highly dependent on the Internet; actually, progress has been full of growth for globalization of software development is possible through the presence of the Internet [24].

Many teams still rely on telephone to conduct immediate direct talks. Internet has brought about how advances in distributed teams can get benefits from a different kind of phone Voice over Internet Protocol (VoIP) phone. This technology makes the call possible for team members in any location you have a number in the same area

code and the other team members, and the reduction of long-distance or international call costs. VoIP can also be used to put the team members on the same phone network, this means that one of the members of the team in a different location has the possibility of reaching by writing the extension for entering into the entire sequence of the area code, and so on. Simple things like being on the same phone network can make the difference feel more cohesive [24].

Instant Messaging (IM) can be a quick real-time way to get an answer from a teammate without the disruption of a phone call. Most chat software allows each person to set or develop his position to availability. If a person is not available, then IM can be considered asynchronous similar function to send E-mail. If he/she is available and communicate with the disruption shall be a little quicker.

"Webcast or webinar, what is the difference?" The difference varies from person who uses this term, but in most cases, news refer to a presentation via Internet, where the sound in one direction only. Webinar online seminars generally allow two-way or multi-directional communications.

Online machine translation services and translation software installed locally to provide some of the information in time for the team because the distributed services are readily available and can be accessed when needed. Global teams sometimes need to work out differences between languages. It can also be used to translate a piece of a message or document jointed by the communications team, summary or informal.

Many teams relies distributed synchronous tools, but there are teams in situations where real-time communication does not work well. We have global or distributed teams to deal with different time zones and different work schedules, and so not all the team can be available at the same time.

2.3.2.2 Asynchronous Communication Tools

In a globalized workforce, one big problem is time zone differences because is difficult for team members to find a common time of the day, which are all available for meetings or other forms of communication. So far, it can be the time zone differences to benefit the software development team productivity because they let the team work together around the clock. Therefore asynchronous tools are an asset to distributed teams because they facilitate communication without respecting time.[24, 8].Table 2.3 show example of asynchronous Communication tools

Table 2.3 Communication tools: asynchronous [24]

Tools familiar to traditional teams, such as e-mail and calendars on the Internet can help to fill the need of communications management across the team of fundamental differences in the time zone tables. Transfering files keep track of the case and put the information in the available records to other team members when needed. If the team is distributed, or at the same place communication and information exchange needs to happen for the team is to be successful. Asynchronous communication tools allow the team to choose the right time to access the information, and minimize disruption of the schedule by any individual.

E-mail is well known enough now, that almost everyone is aware of in 2002, the number of e-mail users worldwide was around 890 million and by 2009, it approximately reached 1.4 billion [24]. It has become a standard form of communication within teams and hard to be replaced. With the flexibility of e-mail, a team member or leader used to send messages to one person or the entire groups that may include file attachments if necessary. Beyond the basics of sending and receiving, E-mail software can be used to organize and archive information for ready access.

Groupware or share calendars are useful tools to keep rapid with the available and obligations of team members. It keeps each member with his or her own calendar, and sets limits on who has access to entries so the other team members can know who is available, who is on vacation, or when it may be available to meet.

In many cases, it may happen online group meeting or conference call in time that is inconvenient for one or more employees. When this happens, or when important information is

Exchanged, team manager may choose to record the call, create archives record that can be referenced later. The recordings can be considered by the same way that webcasts, but often broadcast live to provide information flowing. By recording broadcast on the Internet or calls, information can be grouped with training materials via the Internet or text-based minutes to review asynchronously.

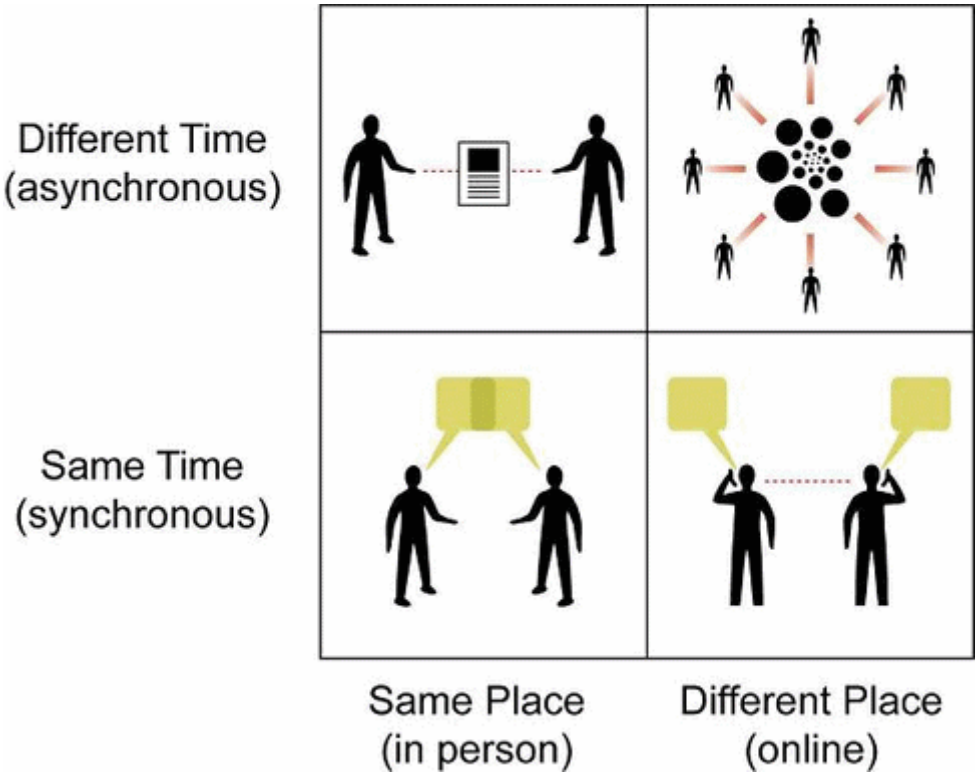


Figure 2-6 Synchronous VS Asynchronous Communication

2.3.2.3 Knowledge Transfer Tools

The knowledge transfer is a specialized form of communication; it can be hard to do at a distance. For global software development teams, which need to be closely coordinate their activities and to ensure that their products integrate well members, the transfer of knowledge can be a challenge. What is the transfer of knowledge?

On any team, and some people have more experience in one area than another country. For example, in the software development team, there are roles for specializes in gathering requirements, database design, and determine the best user interface, or evaluating the appropriate platform [24].

What are the types of tools that are available to support the communication? Table 2.4 shows examples of the many tools that can be used for the exchange of knowledge in an informal manner or for routine training operations are more controlled and planned.

Tool	Examples	Features	Primary advantage
Wiki	WikiPedia, PBWiki, TikiWiki, DocuWiki, MediaWiki	Information in many languages on many topics, constantly updated by contributors, easy to create a custom wiki for a team	Open sharing of information, anytime
Electronic library, institutional repositories	BRICKS, Fedora, Greenstone, Invenio, Refbase, and numerous others	Immediate access to library-type information, books, journals, graphics	Open sharing of information, anytime

Table (2-4) Communication tools: knowledge sharing and training [24]

Search tools	Google, Yahoo, Bing, Ask.com	Specialized search libraries, such as Google Scholar; access to worldwide extensive information sources	Open sharing of information, anytime
e-learning	Articulate Online, Adobe Elearning Suite, Moodle, SnagIt	Audio/video capture, develop tutorials, learning management systems, image capture (classroom)	Create once, and reuse any time, web-based presentation for easy access

What is a wiki? The wiki is a Web site that anyone can contribute to using the built-in tools directly to the editing Web pages. Wikipedia, the free encyclopedia, is a famous example offered in many languages (http://www.wikipedia.org/) can be distribute teams to create and use a wiki site to share information on the new tool, the package they are developing, or anything they want to cooperate in list can be endless [24].

Electronic libraries have many names, including the digital library and virtual information services, institutional depots, and many other variations, but they all serve the same purpose these libraries are electronic files of reference materials in the preparation of documents and regulatory materials, specifications and documentation software life cycle, and other sorts of written information E-learning tools provide a way to provide custom-made, and standardized training for the team, without having to travel. E-learning is any kind of learning,

which uses some combination of a computer and / or Internet access. Generally, it includes learning on the Internet, and computer-based learning, or arranges virtual classrooms. Often the information is delivered via the Internet through the packet or recorded video, or from a CD-ROM or DVD [24].

2.4 Requirement Management In global Software development

Requirements engineering is a difficult task when done locally, but it is even more difficult when cross-functional stakeholder groups specify requirements across cultural, language and time zone boundaries [11]. The goal was to examine RE practice in global software development, to formulate recommendations for improvement as well as to provide directions for future research on methods and tools. Based on the empirical evidence, we have constructed model of remote communication and knowledge management, cultural diversity and time differences negatively impact on requirements gathering, negotiation and specification. A field investigation of RE in a multi-site developmental organization is described to illustrate the effects of stakeholders' geographical distribution depends on requirements collaborative activities as well as specific challenges which placed in conducting this research. Finding emphasize major problems in communication, is a critical ingredient of collaborative activities in software development. Furthermore, geographical distribution brings about cultural diversity and contribution to ineffective knowledge management, subsequently affecting global software development [12]

Requirements engineering (RE) is a necessary part of the software development process, as it helps customers and designers in identifying the necessary system requirements .If these stakeholders are separated by distance, we can argue that a distributed groupware environment which supports a cooperative requirements engineering process must be supplied allowing them to negotiate software requirements. Such a groupware environment must support aspects of joint work relevant to requirements negotiation: synchronous and asynchronous collaboration, telepresence, and tele data. It should also add explicit support for a structured RE process, which includes the team ability to discuss multiple perspectives during requirements acquisition and traceability [13].

Distributed software development presents several characteristics that are different from co-located software development. Currently, one of the main difficulties imposed by geographically distributed software teams is the requirements engineering process. Traditional approaches to the requirements process don't cover the fundamental difficulties of dispersion, like communication and coordination. The objective of this study is to present a proposal to adapt the specification process to address the main challenges found in distributed software development environments [14].

Software development teams are often geographically distributed from their customers and end users. This creates significant communication and coordination challenges that impact the effectiveness of requirements engineering. Travel costs, and the local availability of quality technical staff increase the demand for effective distributed software development teams the engineers conducted a requirements analysis and produced a Software Requirements Specification (SRS) document through distributed interaction with the remote customers [16].

The requirements management in geographically distributed environments, identifying the main challenges, the results suggests the necessity to adapt the requirements management to the distributed software development environment, addressing the main existing challenges.

As a consequence, the communication between the project team, users and customers occurs in a geographically distributed way. In this case, the requirements management seems to be an even more critical activity. Normally, the requirements' gathering occurs in meetings having all the participants (project team, users and customers) in the same place.

In distributed software development environments the challenges become even more significant [15].

2.5 Requirement Challenges In global Software development

With global distribution has caused problems for globally distributed software teams, such as the breakdown, and described in the form of a model of RE challenges due to geographical distribution

of stakeholders, illustrated.

• Communication is not sufficient. Distance enters the barriers of communication and informal face to face, and stakeholder communications depend on the quality of the use of electronic communication tools synchronously or asynchronously. In this study, interest groups (customers, business management, and developers) are not effectively communicates all sought to exercise power and influence on each other's.

• Knowledge Management. Do not participate in the vast amount of information about the requirements from multiple sources in remote customer sites properly with developers. Moreover, through the guidance of information about the requirements of the business strategy for developers through the key stakeholders, development manager take the advantage of distance to promote certain positions of authority in the organization [11].

In [11, 12] mentioned more challenges such as.

- Diversity in customer culture and business
- Achieving appropriate participation of system users and field personnel
- Lack of informal communication and diminished awareness of local working context
- Informal communication within the development group
- Reduced level of trust
- Difficulty in managing conflict and having open discussions of interests Difficulty in achieving common understanding of requirements
- Ineffective decision-making meetings
- Delay

In [13] the problem is that stakeholder's software can operate in environments that reflect organizational and national boundaries, where you may find team members and sub-groups who are isolated from each other by distance and time. One solution to this problem is to have a team networks which use groupware specifically designed to support distributed RE process.

Requirements engineering is a critical phase of software development, which represents several new challenge and exacerbates its fundamental ones when it is in distributed environments, the requirements engineering seems to be an even more critical phase. Normally, the requirements gathering and specification occurs in meetings having all the participants (project team, users and customers) at the same place. The quality and the capacity of analyzing and managing the requirements of software projects not only affect the final product quality, but also the time necessary to satisfy the requirements. Badly identified requirement can compromise the project success, generating delays or project cancellation [14].

2.6 Application Lifecycle Management

Application Lifecycle Management (ALM) coordinates the life-cycle activities of software development, including architecture management , requirements management, change and software configuration management, build and deploy management, and quality management for the concept of Application Lifecycle Management , on the other hand, it indicates the coordination of activities and the management of artifacts

(e.g., requirements, source code, test cases) during the Software (SW) product's lifecycle .ALM as a concept is completely new , most of which have been discussed in professional articles , has been discussing the concept of ALM from various points of view for example:

1) Model-driven development.
2) Complex systems development.
3) Technology and ALM tools.
4) Only treated in a cursory way.

According to [25] companies are aware of the ALM as a concept which is unclean tool to provide their own definitions of ALM that reflect their backgrounds and marketing strategies. However, it is hard to find articles that discuss the concept what constitutes ALM? This may affect the explanation of the whole concept of ALM which is not clear and driven by tool vendors. It argues that a wide range of tools described as ALM tools because of the scope of the loose definition of ALM.

Most of the discussions on the ALM concept are professional publications: books and professional articles. (Schwaber, 2006) She defines the three pillars of ALM to be traceable, process automation and reporting. An important viewpoint on ALM is that it does not focus on any specific lifecycle activity, but keeps all the activities synchronized. Therefore ALM is a thread that ties the development lifecycle together from business needs to operations. It also has been identified to provide support for project management.

According to Doyle (2007), and the appropriate ALM tool provides strong support for project management, for example, providing an objective means to monitor the activities of the project and generate reports in real-time project data. States that is the ALM does not necessarily require tools. Traditionally, it has been dealing with the life cycle activities partly manual processes.

Doyle and Lloyd (2007) for example, states that the management of the important requirements for ALM, and presents the idea of that concepts, such as tracking, process automation, reporting and integration related to ALM tool. Moreover, the basis of ALM comes from the Software Configuration Management (SCM), ALM tools have their roots in the formation and management of Integrated Development Environments (IDEs), the current SCM tools as usual basis of ALM infrastructure [25].

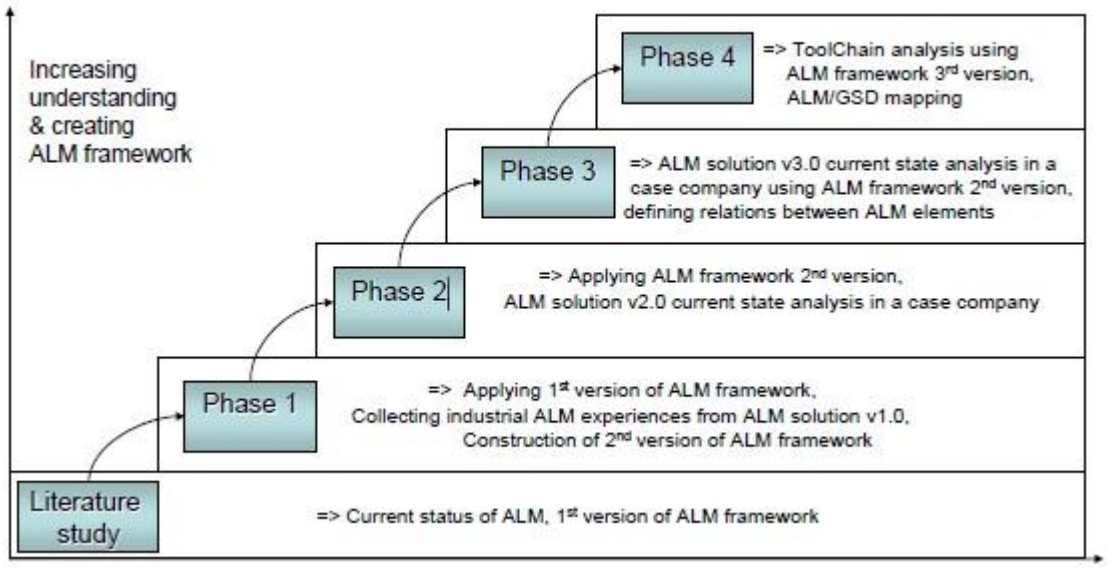


Figure 2-7 Phases of the ALM framework construction and demonstration.[25]

Chapter Three

Global Software Development Model and Tools

3.1 virtual teams

1- A virtual team - also known as Geographically Distributed Team (GDT) is a group of individuals who work across time and space, and organizational boundaries with links strengthened communication technology networks (e-mail, video-conferencing, telephone, etc.).

2- A virtual team does not always mean teleworkers. Teleworkers are defined as individuals who work from home. Many virtual teams in today's organizations consist of employees both working at home and small groups in the office but in different geographic locations.

3- A virtual teams represent a growing response to the need of fast time-to-market; low-cost and rapid solutions of complex organizational problems, virtual teams enable organizations to gather the talents and expertise of employees and non-employees by eliminating time and space barriers.

4- A virtual teams are often formed to overcome geographical or temporal separations. The term "virtual team" is used to cover a wide range of activities and forms for technology-supported working.

5- A virtual teams are comprised of members who are located in more than one physical location, this team trait has fostered extensive use of a variety of forms for computer-mediated communication that enable geographically dispersed members to coordinate their individual efforts and inputs.

(Gassmann and Von Zedtwitz ,et al ,2003) defined “virtual team as a group of people and sub-teams who interact through interdependent tasks guided by common purpose and work across links strengthened by information, communication, and transport technologies.

Lurey and Raisinghani (2001) defined virtual teams - groups of people who work together although they are often dispersed across space, time, and/or organizational boundaries.

(S. Chinbat 2010) defined virtual team is a building block of global software development. There are many different names and definitions used for global virtual team in different literatures such as virtual team, globally distributed team, international virtual team etc. Each of these names refers to slightly different meaning.

The degree of geographic dispersion within a virtual team can vary widely from having one member located in a different location than the rest of the team while having each member located in a different country [37].

3.2 Characteristics of virtual team

- Geographically dispersed (over different time zones)
- group of people working on same software project
- Best employees may be located anywhere in the world.
- work across different geographic locations (particularly different countries)
- Driven by common purpose (guided by a common purpose)
- Workers demand personal flexibility.
- team members have culturally (language, custom, behavioral norms etc) diverse background

- Enabled by communication technologies.
- Workers demand increasing technological sophistication.
- Involved in cross-boundary collaboration
- A flexible organization is more competitive and responsive to the marketplace.
- It is not a permanent team
- Workers tend to be more productive – less commuting and travel time.
- Small team size
- The increasing globalization of trade and corporate activity.
- Team members may belong to different companies
- The global virtual team workday is 24 vs. 8 hours.

3.3 Barriers and Challenges for Virtual Team

Virtual teams face particular challenges involving ***trust*** which is a key element to build up successful interactions and to overcome selfish interests, ***effective communication*** that is even more critical for success in the virtual setting, ***deadlines***, and ***team cohesiveness.*** While there are great advantages that come with the adoption of the virtual teams, new challenges rise with them.

Declared there are five main disadvantages of a virtual team: *lack of physical interaction*, *loss of face-to-face synergies*, *lack of trust*, *greater concern with predictability and reliability*, and *lack of social interaction*. In building a virtual team, all of these issues must be at least implicitly addressed in order to have an effective virtual team.

Virtual teams are challenged because they are virtually, they exist through computer mediated communication technology rather than face-to-face interactions, it refers to some other of the difficulties and barriers such as [37]:

- Knowledge transfer (specially tacit knowledge) becomes difficult
- Remote communication problems: ambiguity in communication, less communication richness
- Difficulties coordination of team members efforts
- Cultural issues (language barrier)
- Reduced opportunity for building personal relationships
- Low level of team spirit
- Low level of trust between team members
- Lack of common standard for process, activities and terms (software configuration, difficulties in establishing shared understanding)

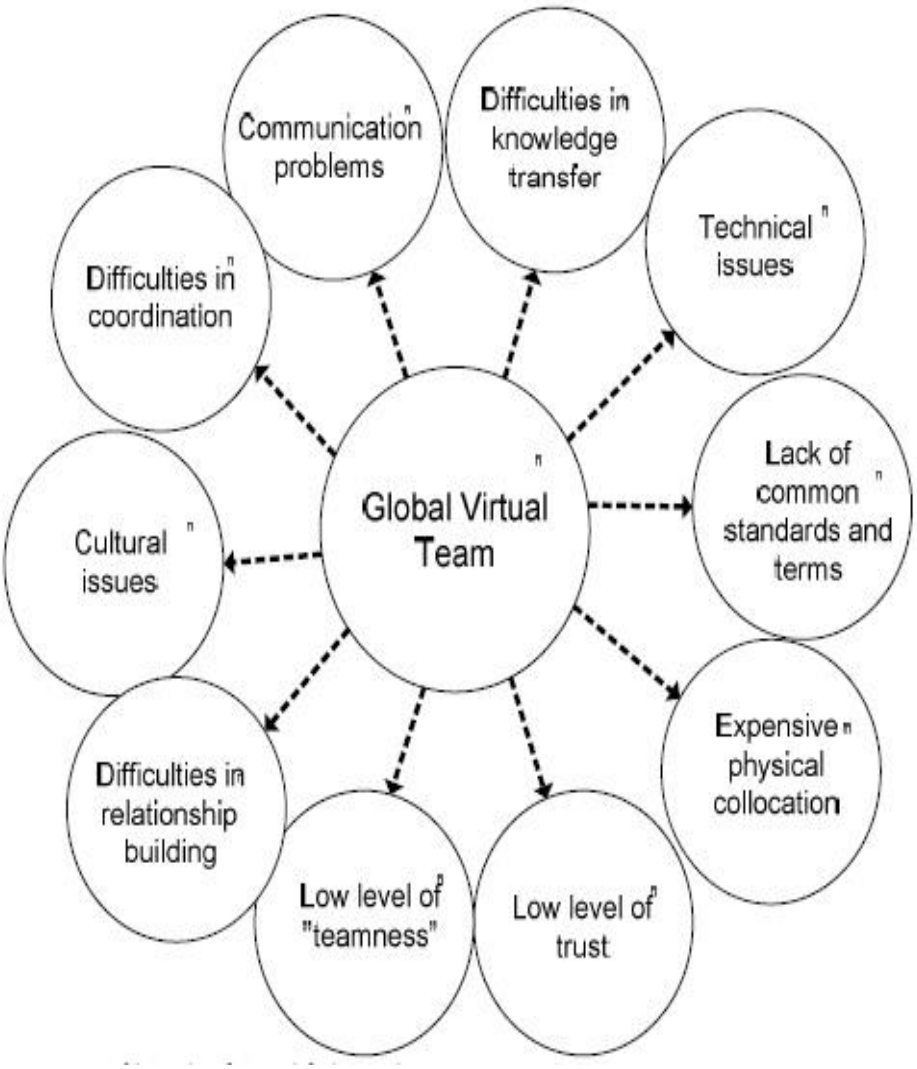


Figure 3.1 Problems that draws global virtual team apart [37]

3.4 Benefits and Draw Back of Virtual Team

The flexible and configurable based on infrastructure is one of the main advantages of agile virtual teams. It indicates that the effective use of communication, particularly during the early stages of development, plays an equally important role in the acquisition and maintaining confidence. Virtual teams that do not work at the same time or place often face tight schedules and the need to quickly start and perform immediately.

Virtual team may allow people to collaborate more productivity at a distance. As a drawback, virtual teams are particularly vulnerable to mistrust, communication break downs, conflicts, and power struggles. On the other hand, virtual teams reduce time-to-market. Lead time or time to market has been generally admitted to be one of the most important keys for success s in manufacturing companies. Table 3.1 summarizes some of the main advantages and Table 3.2 some of the main disadvantages associated with virtual teaming [37].

Table 3.1 some of the main advantages associated with virtual teaming [37]

Advantages	Reference
Reduction time and costs relocation, and travel costs (Virtual teams overcome the limitations of time), space,	(McDonough et al., 2001, Rice et al., 2007, Bergiel et al., 2008
Reducing time-to-market	(Lipnack and Stamps, 2000, May and Carter, 2001, Sorliet a l ., 2006,
Able to digitally or electronically unite experts in highly specialized fields working at great distances from each other	(Rosen et al., 2007)
More effective R&D continuation decisions	(Cummings and T eng, 2003, Schmidt et al., 2001)
Most effective and rapid in making decisions	(Hossain and Wigand, 2004, Paul et al., 2004b, Bal and Gundry, 1999)
Able to tap selectively into center of excellence, using the best talent regardless of location	Criscuolo, 2005, Cascio, 2000, Samarah et al., 2007,
Greater degree of freedom to individuals involved with the development project	Ojasalo, 2008, Badrinarayanan and Arnett, 2008
Greater productivity, shorter development times	(McDonough et al., 2001, Mulebeke and Zheng, 2006)
Producing better outcomes and attract better employees, Generate the greatest competitive advantage from limited resources.	(Martins et al., 2004, Rice et al., 2007, Chen et al., 2008b)
Useful for projects that require cross-functional or cross boundary skilled inputs	(Lee-Kelley and Sankey, 2008)

On time implementation of the tasks assigned, Less resistant to change	(Precup et al., 2006)
Integrating talent in newly industrialized Facilitating transnational innovation processes	(Gassmann and Von Zedtwitz, 2003b, Prasad and Akhilesh, 2002)
Higher degree of cohesion (Teams can be organized whether or not members are in proximity to one another)	(Kratzer et al., 2005, Cascio, 2000, Gaudes et al., 2007)

Table 3.2 some of the main disadvantages associated with virtual teaming [37]

Disadvantages	Reference
Sometimes requires complex technological applications	(Bergiel *et al*., 2008, Badrinarayanan and Arnett, 2008)
Face-to-Face collaboration (FFC) appears to be better developing a conceptual understanding of a problem lack of physical interaction	(Cascio, 2000, Hossain and Wigand, 2004, Kan k an h al l i *et a l* . , 2006)
Decrease monitoring and control of activities	(Pawar and Sharifi, 1997)
Everything to be reinforced in a much more structured, formal process	(Lurey and Raisinghani, 2001).
Vulnerable to mistrust, communication break downs, conflicts, ، and power struggles	(Rosen et al., 2007, Cascio, 2000, Ki rkman et al., 2002, T aifi, 2007)
Challenges of project management are more related to the distance between team members than to their cultural, or language differences	(Wong and Burton, 2000, Martinez-Sanchez *et al*., 2006,
Challenges of determining the appropriate task technology fit	Qureshi and Vog el , 2001, Ocker and Fjermestad, 2008, Griffith et al., 2003, Badrinarayanan and Arnett, 2008, Bell and Kozlowski)
Challenges of managing conflict	(McDonough et al., 2001, Mulebeke and Zheng, 2006)
Team members need special training and encouragement	(Lee-Kelley and Sankey, 2008)

3.5 G.S.D Global Software Development

Globally Software Development (GSD) are projects that consist of two or more teams working together to accomplish project goals from different geographical locations. In addition to geographical dispersion, globally distributed teams face time-zone and cultural differences that may include but they are not limited to different language, national traditions, values and norms of behavior [6].

Globally Distributed Software Development Team (GDSDT) refers to the process of development of information system in global context of location around the world .Technological improvements in the fields of communication and networking are making distributed group work more feasible. Increasing quality and availability of this technology makes distributed software engineering efforts more viable option for organizations. Software development projects, which utilize distributed teams, are members of the project group which they are geographically distributed among different physical sites.

In Software Development Virtual Team the authors described the (SDVT) as a "group of people who work independently with shared purpose across space, time, and organization boundaries, using technology to communicate and collaborate." As such, virtual team allows organization together people to gather with the best expertise, regardless where they live.

3.6 Characteristics of Global Software Development

The Global Software Development (GSD) collaboration and coordination is important in software development because it leads to benefits such as cost saving , shorter development cycles , and better-integrated products but this perspective has only been investigated in real-time collocated tasks and we know little about which types of team knowledge best help coordination in the most geographically distributed software work and the main reason for distributing software development is to improve time-to-market by round the- clock development or to increase flexibility and take advantage on merger and acquisition opportunities. Other reasons include access to cheaper labor, increasing knowledge of customers and local conditions by market, or take advantage on the global talent pool [5]. In fact, shortage of highly skilled science and talent engineering needs for the access of qualified personnel which are important explanatory factors for off-shoring innovation decisions. In line with these trends, distributed software development is now no longer only an option for most enterprises; rather, it is a business necessity [40].

3.7 Barriers and Challenges for GSD

Today's software projects are often distributed and development across multi geographic locations because of the globally distributed software development has become very important issue for organizations at the present time in the climate of increasing the tendency towards globalization and global outsourcing. This distribution poses new challenges produced by the cooperation across different countries, times zones, and cultures, languages, trust factors, communication across temporal, spatial distances, lack awareness of contextual shared, project management, risk management and coordination.

3.7.1 Technical issues

- Communication across temporal: Inadequate communication distance introduces barriers to informal and face-to-face communication, and the stakeholders' communication is depends on the quality of using synchronous or asynchronous electronic communication tools and knowledge transfer. As referred to in the proposed general framework of GSD mentioned in Chapter four precisely in productivity tool module.

  Synchronous tool: such as Telephone, Instant messaging and chat, Video conference, Web casts, and online translators

  Asynchronous tools: E-mail, Groupware/ shared services, and Recordings.

Knowledge Transfer: such as plat form, PB works, Smart Cloud IBM and Id space.

- Knowledge management: The sheer quantity of information about requirements from multiple sources at remote customer sites was not appropriately shared with the developers.
- Time difference. The large distribution of stakeholders across five continents introduced large time-zone differences and allowed little overlap available for synchronous collaboration. Hence asynchronous channels were predominant in the communication, complemented by teleconferencing calls. Synchronous meetings across continents are always un suitable for at least one site – either too early or too late in the day, and involve someone having to compromise on their work schedule.

3.7.2 Non-Technical issues

- Cultural: differences imply different terminologies which may cause mistakes in messages and translation errors. Different levels of understanding the problem domain also exist, as do different levels of knowledge, skills, and training between teams
- Reduced level of trust factor
- Languages

3.8 Overview of idSpace Platform

The idSpace platform supports online distributed collaborative product innovation. It differentiates between user innovators and a moderator, who sets up and guides innovation sessions. The platform is supported by a wiki which functions as an online help for the users [27, 29]. IdSpace, an internet-based‹ creativity support platform offers its users a variety of creativity techniques, pedagogical approaches and context aware use of available information on projects, people and techniques. The idSpace project focuses on research and development platform for distributed collaborative product innovation and aimed at resolving number of deficiencies in the now available alternatives. Topic maps is a significant factor in solving a number of issues, such as the concepts of participation, re-use ideas and knowledge and to create an integrated ontology connects all aspects of the creative process together [26] . The project provides with a way to help the user to establish a fruitful cooperation with the other team members who work in different locations.

The ultimate goal of the idSpace project is to build, in prototypical form, the idSpace environment that should come to the aid of distributed teams of innovators who want to collaborate on product design, thereby making use of earlier results by themselves or even others.

3.8.1 The IdSpace Environment:

Contains a set of integrated tools help to track and store semantic relations between conceptual models, which are used to describe the ideas and objectives, benefits and values. Such as extended exhibitions, educational curricula to connect informal and paid specific uses of the platform [26]. A flexible, context aware, web-based platform, which forms the substrate for communities of practice to grow, thrive and learn the idSpace environment features an integrative toolset that will allow

- A distributed team of innovators
- To create new ideas,
- To contribute to and preserve existing ideas
- To learn about them.

3.8.2 Approach Chosen for the IdSpace Project

Has been set for the project idSpace with very high ambitions. Formulated as the following objectives of the project [26]:

- Must platform idSpace support a range of educational strategies and several types of creativity techniques.

- The platform should facilitate distributed collaboration and support users who are not experts either in the field of creativity or in the field of semantics .The platform should enable collaboration among users across different sites.

The way to do this is through the use of Internet technology as a basis and combination of visual as well as text based editing aids. All of this should be designed in a way that requires no knowledge of semantics. The user also doesn't have to be familiar with the structure and principles of creative processes, with the exception of the moderator who defines the parameters of a creativity project.

- The platform should be building on semantic principles, specifically Topic Maps, and should include an ontology which integrates all elements of a creative process.

- The tools help to track and store semantic relationships among conceptual models, which will be used to describe ideas, goals, features and values.

3.8.3 Use IdSpace for Evaluation Plan & Planning

The main objective of rating idSpace is analysis of the feasibility and impact of advanced technology, showing the efficiency and relevance, significance and usefulness idSpace platform as a tool to support innovation and creativity. Plan the Evaluation Activities: For each phase of the development process of the idSpace platform (planning, development and demonstration) concrete evaluation activities and tasks will be performed regarding the Methodology Evaluation, the evaluation will provide evidence for the efficiency, appropriateness, meaningfulness and usefulness of the idSpace platform as a tool for supporting innovation and creativity[28].

3.9 Overview of Wiki

The term “Wiki” derives from the Hawaiian phrase, “wiki-wiki,” which means fast or quick, Wikis are informational websites which can be edited by any user. All users can contribute to the webpage; Wikis are ideal educational tools to facilitate not only the communication of information, but also the sharing and collaborative growth of knowledge. They are typically organized by content rather than chronology, Wikis are lightweight technology to support the establishment and maintenance of web pages easy, flexible structuring of the contents is a special feature which makes wikis sufficient and popular as knowledge repositories. The wiki can be used to save the shared information resources of a group such as an organizational unit or the Internet community allowing the members add and modify contents, The contents of a wiki may be shared with the world or limited to a group or community .Also wiki is defined as "a scalable freely from web pages interconnected system of hypertext to store and modify information - a database where each page easily by any user for editing with Web Forms able to client browser [30].

Lightweight Project Management (PM) solutions are often sought when ‘heavy’ PM is not feasible, as seen in open source software development.

It refer to the potential advantages of the wiki to support Software Engineering (SE): Versions comprehensive change control, and provide a shared workspace for the production and storage of project documents, and access to input from many of the participants in more effective manner, than through e-mail.

That Wikis are particularly useful media discussion, the advantages of more public discussion lists as ease of use, and the fact that the elements of the discussion can be placed in its proper context. Moreover, it is referring to the Democratic side: wiki project gives a voice to all in the project [31].

Presents survey findings of innovative forms of wiki use in student SE teams the project wikis contain or provide links to various project artifacts, convey team-internal as well as information for external stakeholders and generally seem to reflect a mixture of administrative, development related and socio-emotional aspects of project work .For example, the work of Chao on the use of wiki projects SE defines the different types of use, but only at the level overview based [32]. Louridas addresses the potential of the wiki in SE projects without providing experimental results on the actual use.

3.9.1 Wikis and Collaborative Learning

Perhaps the most promising aspect of the Wiki is its ability to enable collaborative learning. In collaborative learning, students work together to support each others' individual learning .Collaborative learning enhances students' interdependence within their work group as well as their individual accountability for completing their work. Collaborative learning is also essential for facilitating the distribution and sharing of knowledge and expertise among a group of individuals. Wikis are also promising because they involve the incorporation of technology that has already been adopted by students in their everyday lives [33].

3.9.2 How Wikis Can Assist In the Software Development Process

This section discusses the areas of software development and how structured wikis help to write documents such as software requirement specifications in a traditional environment or user stories in an Agile environment, store test plans, and incubate and produce user documentation in online and offline formats.

Software requirement specifications, which list the features and designs required to be part of a complete release of a software component, are replaced with user stories in Agile that are thinly sliced representations of what a user can do with a software feature [34].

Since the completion of a certain number of user stories usually through repetition, it is important for the user story to contain only the amount of code that can be written and tested (and probably documented) in a certain amount of time for recurrence. By storing these stories used in wiki, you offer the latest update on the Internet at all times, and you give colleagues the ability to discuss user story on a wiki for everyone to see.

Wiki pages that contain user stories also give a picture of the dashboard of progress on each feature or user story. A lot of discussion about the user story may indicate that the design is not settled yet.

Quality engineers must devise a document describing the scope of the test, and focus, and the approach they plan to take and ensure the feature of the program is tested to the limits and also indicate

whether the passes specific thresholds to ensure that the program is useful. By storing test plans on the wiki, using writing the appropriate topic such as the concept and mission, and the team which can review test plans, and search for omissions or errors, put the outer limits of the test, and to discuss the approach, scope and focus of the test[34]. Unit tests written by quality engineers and developers can also be stored on wiki pages for reuse across team members and as examples for others to follow.

3.9.3 Key Wiki Features For Software Engineering

Wikis have dozens of features, and not all of them are critical to software engineering processes. The following list is not offered as a complete list, but rather, a summary of some key features that facilitate information exchange and publishing processes [34].

- Paste screen shots from the clipboard
- Embed recordings for in-page viewing
- Transclusion of content from other pages (where the source is editable and transcluding instances read-only)
- Filter content based on audience needs (summaries versus details)
- Built-in workflow
- Integration with issue tracking systems
- Plug-in architecture
- Standard APIs for integration and content extraction

3.9.4 Advantages of Wiki

Because wiki gives online groups space for storages, sharing documents, information exchange, and work collaboratively as many contacts you do with the technical co-authored works, the wiki can be a great boon to virtual teams. Other tools are available to support the distribution of sharing and exchanging information (such as e-mail, shared server space, and content management systems), but it can be clumsy in some cases [35].

Materials on the wiki are available to team members from any location at any time of day. Team members can access materials from any computer whether they are in the office, traveling, or working from home. In most cases, team members receive an email notification when changes are made to the wiki, facilitating communication among group members even when members are working on different schedules and different locations.

In large projects and multi-author documents, files via e-mail back and forth quickly take account space as well as create mass confusion on the currency of the draft. Shared server space alleviates many exchange problems, but version control is still an issue when multiple authors are attempting to edit a file. Wiki, on the other hand, has the advantage of being live, shared space, where all team members are in writing and editing privileges, availability and ease of use from anywhere can make all the team members involved in the production. Wiki is a simple way of technologically to grant access editing to the Web site for many people without the need for team members to direct

requests editorial by administrators and without fears that Web author novice will destroy the site by deleting files accidentally. It is to reduce the time it takes to update the wiki, and it is likely to become a useful space and use it regularly. Ease of cooperation in the wiki can make it a powerful tool for project management and collaborative writing. Centralized wiki can be useful for team members who are traveling, and probably did not have access to a specific word processing programs. Wikis public that provide information to a wider audience has the advantage to attract any shareholder who will add or edit content. In addition to the productivity advantages, we can use wiki shared social space for team members who work remotely [35].

Many programs available for developing wikis, such as PBworks <www.pbworks.com> or Wikispaces <www.wikispaces.com>, provide page templates to help build and develop the space to suit the needs of a team. For example, the organizer can develop a team member page with contact information or a calendar with pertinent scheduling information.

3.9.5 Disadvantages of Wikis

Despite their benefits, wikis also have some disadvantages. Basically, they require users to learn wiki syntax in order to maximize take advantage of the capabilities of a wiki format. Adding plain text on a page is simple, but coordination addresses, lists or tables requires the use of a somewhat ambiguous system of punctuation. Edit pages through the web browser usually do not allow users to spell check, or have the same sophisticated editing function of a word processing program.

Wiki editing can also intimidate users new to the collaborative environment. If writers and editors accustomed to collaborative visual cues provided by the Microsoft Word because they modify documents by using track and comment boxes, the vague nature of these activities in the wiki editing may be disturbing. The 24-hour rule can be very helpful with a public site such as Wikipedia but it could cause problems with a small group that is actively editing a time-sensitive document [35].

Another disadvantage of the wiki is that the basic design can be primitive appearance, without graphics or exciting colors, unless you upgrade the core template team. Moreover, navigation systems are not built automatically in the wiki packages. That must be diligent about the team members who create and update navigational links and menus to match the growth of the site.

The wiki can also become resource intensive, consuming storage space for revision history and hogging processor time to calculate differences between edits. In addition, the wiki requires computing resources vital for the renewal of pages per visit rather than static content. Intensive editing, or even watch, can overload rapidly weakened the server.

Finally, the collaborative nature of the wiki, like other interactive programs, requires a group to buy and adopt collective in order to be successful. A lone devotee cannot take advantage of the many features of the wiki, the group as a whole must be committed to using the site. In a small wiki supporting a cooperative group, a hierarchy may develop, then one or two users may become invisible autocrats of the wiki [35].

3.9.6 PB Works: Online Collaboration Space

PBworks is a cloud-based group workspace that is widely used in business and education. It's one of the largest online collaboration services, with over 3 million active users around the world working on upwards of 15 million pages and files. Designated Software-as-a-Service (SaaS), PBworks is completely web-based and can be accessed from any computer with an Internet connection. This is especially useful for working groups that are dispersed over a wide geographic area, such as distance learners.

Core wiki functions such as editing, annotating, and publishing pages are supported, as multimedia features as uploading and sharing files. Features that assist in managing workflow include email change notifications, revision history, IM/chat, and live edit sharing. Security settings also enable administrators to set user privileges at the workspace, folder, and page levels, as well as restrict unauthorized access. Workspaces can also be branded by changing the color and adding a library logo.

Academic librarians can use a PBworks account to support technology classes, assist research groups, provide library use instruction, and build a library intranet page. Basic workspaces are available free for higher education and include privileges for up to 20 users, 5 workspaces, and 20MB of storage per workspace. Premium workspaces with advanced security and access controls are available for a $99 annual fee. Campus Edition, with access for up to 1,000 users, 2GB of storage per workspace, and an account management dashboard, is available for $799 per year [36].

3.10 Overview of IBM Smart Cloud

IBM Smart Cloud involve combines e-mail at the enterprise level, and instant messaging, document editing online, web conferencing, file sharing and social business services in an easy to deploy, simplified package. With Smart Cloud, you can take advantage of the tools essential and effective cooperation in the security-rich environment that helps to simplify and improve daily business interactions with customers, partners and colleagues. Smart Cloud is a force that has been designed to give you the ability to engage the appropriate resources in a timely manner by allowing you to collaborate and work with your external network. As a service to the cloud-based, private collaboration and messaging infrastructure needs are managed by IBM that allows you to focus on your core business [36]. As it refers to in the proposed general framework of GSD mentioned in Chapter four precisely in development environment module.

3.10.1 Characteristic of IBM Smart Cloud

- Powerful business-class email
- Calendaring and scheduling
- Instant messaging
- Web meetings
- Socially enabled online office productivity suite
- Business social networking capabilities
- File storing and sharing and file viewers
- Contact management
- Project management
- Easy-to-use administration tools

- Communities
- Community based Wikis, Blogs and ideation
- Global services offered in 22 languages
- Choose the optional Smart Cloud Traveler for notes to access email from various mobile devices including apple iPhone, apple iPad, Windows Mobile, Google Android and Nokia Symbian mobile devices.
- Choose the optional Smart Cloud Notes Hosted BlackBerry Service to access email, calendar and contacts from your BlackBerry device
- Utilize the optional Smart Cloud Archive Essentials email archive solution to capture and retain the content of users email messages for later legal discovery
- Support for IMAP clients including Microsoft Outlook 2003 and 2007, Mozilla Thunderbird 9.0.1, Apple Mail 4 and 5
- API extension to manage Notes Traveler with your existing Mobile
- Device Management (MDM) tools

Chapter Four

Proposed GSD Framework

4.1 Introduction:

This chapter dedicated for wrap up all observations, examination and testing, reviewing of the literatures, studying of several initiatives and projects, tools and methods, interviewing with experts, practicing several tools, studying several implemented approaches and methods, analysis of results in order to propose a comprehensive framework for GSD,

The key parameters as shown and deducted throughout this work can be presented as Global Software Development Framework (GSDF) consisted of four interrelation component or modules, each one of these modules has models and attributes as shown hereunder:

1- ***Agile Methodology***: The suitable software requirements engineering model methodology may be used in GSD is the Agile methodology and this opinion is made based on Agile strong advantages such as a dynamic(defied and prioritized sets of feature) and incremental (delivering smaller pieces , one by one and with shorter development periods) framework for software engineering project development that maximize team's ability to deliver quickly and respond to evolving and changing the requirements, the capability of using what so called Virtual Scrum Team (VST) cross-functional team in deferent geographical locations instead of Co-located team,

2- Introduced to accommodate rapid product changes ,Frequent releases in short development cycles (Sprints as a basic unit of development in SCRUM ,Time boxing (restricted to specific duration of constant length 2-4 weeks per sprint,), Minimal requirement specification and test documentation (if at all), and last but not least often no formal test cases.

2- Cloud Computation Model: Proposed developmental environment in response to problems of GSD related to accessibility and availability of technical resources which one needed in one place and not in the others so as an integral part of the proposed framework we suggest Cloud Computation Model to be used as an infrastructure and platform for the GSD.

3- ***Productivity Tools and Techniques:*** during the analysis of a range of tools and techniques have been used in facilitating the mutual and effective communication between the global team during all phases of software development we concluded that most of these tools such as free and public, privet media or social network, e-mails accounts , teleconferences , chat application, all provide an adequate and satisfactory means to facilitate the team communication.

4- GSD Performance Assessment Model (GSDPAM):

The main contribution of the work is to come with new framework model for assessing the performance of GSD as a result of critics and evaluate the two previous models proposed by

other researchers [41] the new model proposed by this work incorporates nine rather than seven Independent (predictor) variables and five rather than four Dependent (outcome) variables by adding the following three models or attributes to the IV:

- Cooperation, collaboration and coordination model (CCCM)
- Satisfaction, recognition and reward model (SRRM)

also by adding new important factor in software to the DV:

- Value creation model (VCM)

Fig 4-1 shows the Schematic representation of the general framework of components GSD while Fig 4-2 shows the schematic representation of the original framework for GSD Performance assessment Model (GSDPAM):

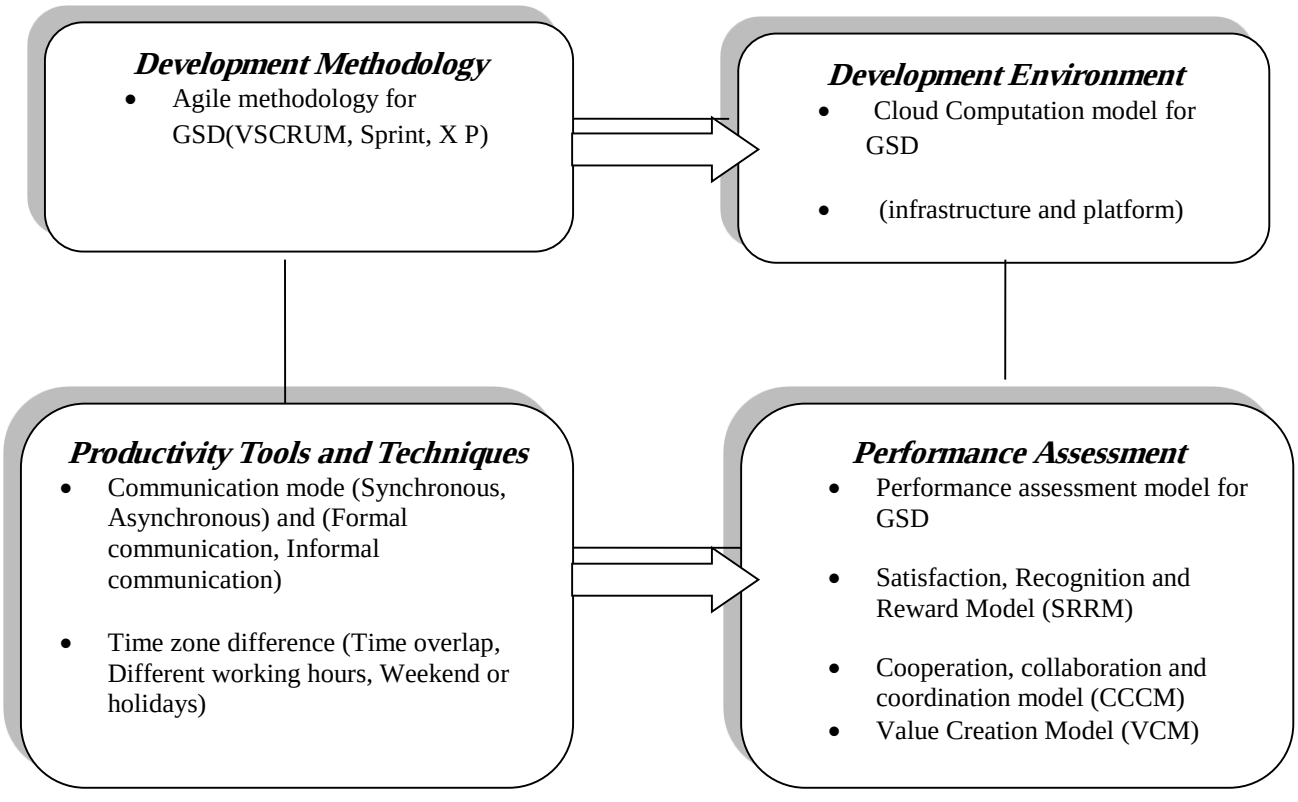


Figure 4-1 Schematic representation of the general framework of components GSD

4-2 Original Model for Assessing the Performance of GSD:

This model contains seven variables as Independent Variables (IV) or predictors and four variables as Dependent Variables (DV) or outcome.

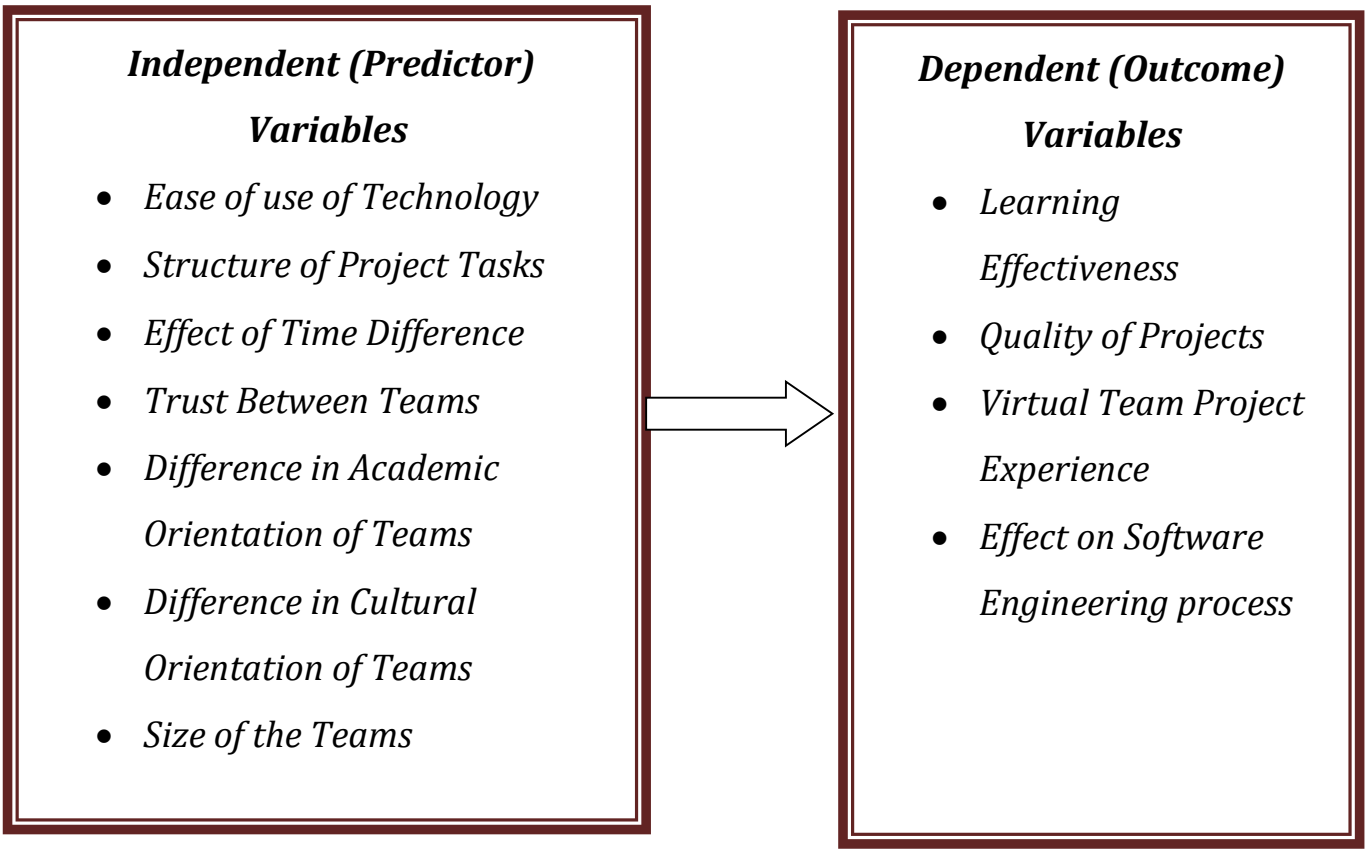


Figure.4-2 Original model for assessing the performance of GSD [41]

4-2-1 Predictors variable:

1- Ease of use of Technology: This variable measures the impact of communication technologies. The importance of communication for the performance of global software development is to overcome distance related problems. The team members were encouraged to use both synchronous and asynchronous technology communication such as chat, document sharing and email. However the Internet bandwidth availability and the reliability of Internet services are different in countries.

2- Structure of Project Tasks: The critical of this variable is to cohesion in task distribution where the work is split up according to feature content in cited as a critical activity for better performance of global teams.

3- Effect of Time Difference: using of this variable is to affirmative effects of the time difference on software project management to introduce “software shift work”. The analyses show the effect of time difference between the multi locations geographically in doing the project using the (Effect of Time Difference). While it difficult to hypothesize on the relationship between the predictor and outcome variables due to lack of previous research in this area.

4- Trust between Teams: Where the parties involved in a business partnership do not see each other. Trust has been cited as one of most important factor especially in the context. There is fortune of research, which systematically examines the effect of trust in the context of electronic commerce.

However, the existing literature lacks to analyses the effect of trust on the effectiveness of global software development teams. Therefore it included trust as one of the predictor variables to measures the extent of which the teams member trusted their capabilities at the beginning and during the course of the project.

5- Difference in Cultural Orientation of Teams: measures dimensions of national culture on the global software development area. The cultural differences include work ethic, work hours, preferred method of communication, revering hierarchy, individualism versus collectivism and concern for quality. Therefore it measures how the peer teams found culture to affect their virtual team project exercise by using the variable (Cultural Orientation of Teams).

6- Difference in Academic Orientation of Teams: measures the effect of differences in academic orientation on the global software development team exercises. So it can expect difference in academic orientation of distributed team's member to have positive correlations with virtual team project experience and quality of projects.

7- Size of the Teams: The effect of teams size on doing the project, large teams require coordination and control, while smaller teams may find it difficult to comp

4-2-2 outcome variable:

1- Quality of Projects: The quality of the requirements defined by the virtual team's member who measured to given the projects.

2- Learning Effectiveness: the team's member will not be effective if the team members themselves are not satisfied with the functions way in virtual environment. Therefore it should determine the effectiveness of virtual team exercise by how the team members felt about the learning process in virtual teams as compared to real co-located teams.

3- Virtual Team Project Experience: The project experience is a measurement of satisfaction or un-satisfaction of team members, who are writhen the virtual team exercise.

4- Effect on Software Engineering process: This variable can be used to determine whether the virtual team exercise has brought any efficiency gain or loss in the requirements definition stage of the software development process. So the fourth variable used to measure the impact of virtual team members exercise on the software engineering process in global distributed. Therefore it is reasonable to expect that the learning effectiveness may be correlated positively with project experience and the software engineering process improvement correlated positively with the quality of the projects.

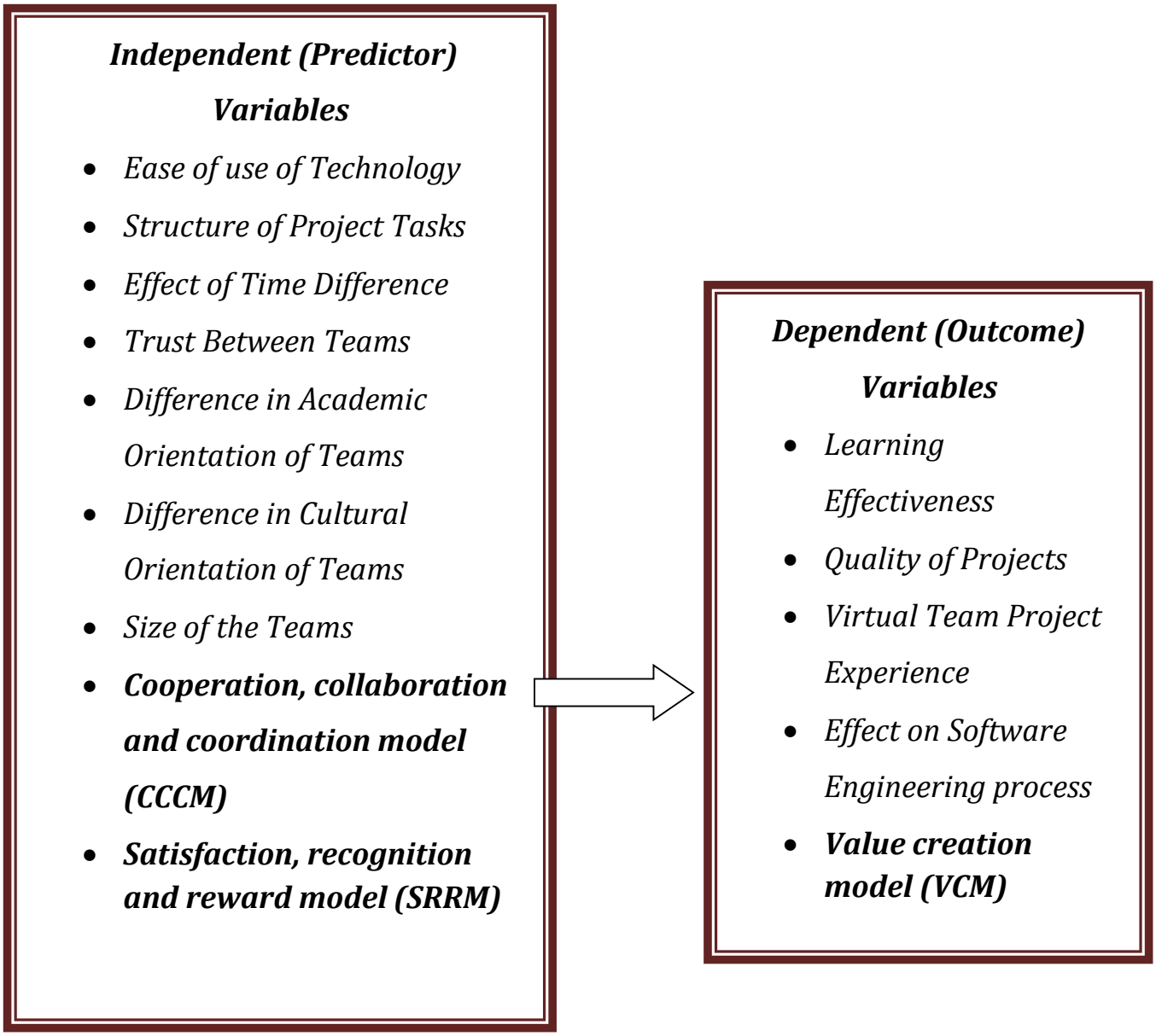


Fig 4-3 shows the schematic representation of the proposed framework for GSD Performance assessment Model (GSDPAM): After modification

Figure.4-3 Proposed Model for assessing the performance of GSD

4.3 Cooperation, collaboration and coordination model (CCCM)

During the investigation of the Model for assessing the performance of Global Virtual Team Projects suggested by Edwards H. k. 2002[41] and based on the literature review made and studied the factors effecting the both the quality and value creation as a competitive advantage for any software products, the level of cooperation with or willingly in assisting the others to act together or in compliance for mutual benefit, also on the level of collaboration ;to work together jointly in an intellectual endeavor ,and the third parameter in that direction is the coordination which enables the virtual team working in harmonious functioning of parts for effective results, so we suggest the Cooperation, Collaboration and Coordination Model (CCCM) to be added on as another predictor in Independent (Predictor) Variable of the original model. (First modification to the model) the CCCM depicted in Fig 4-4.

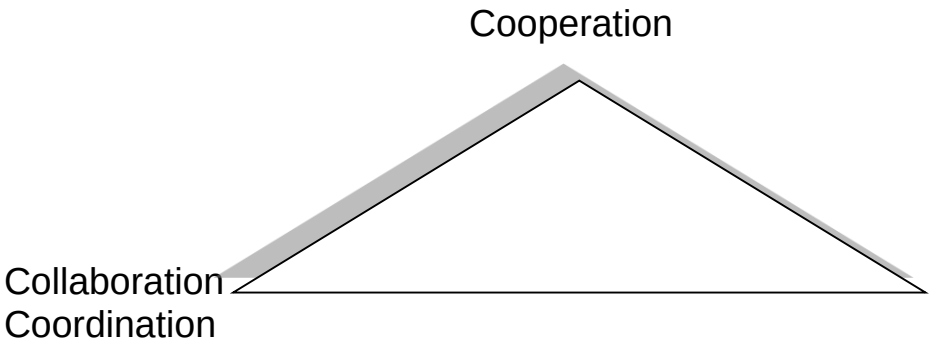


Figure 4-4 Cooperation, collaboration and coordination model (CCCM)

4.4 Satisfaction, Recognition and Reward Model (SRRM)

Another modification suggested by this work through adding an empower and motivation factor working with the (CCCM) model in order to maintain the good attitude of Virtual Team Members (VTM). The suggested new Independent (Predictor) Variable are Satisfaction, Recognition and Reward Model (SRRM) .The management is responsible for keeping the balances between the attributes to this model for the benefit of all stockholder of the software engineering project, the Satisfaction, Recognition and Reward model (SRRM) depicted in Fig 4-5.

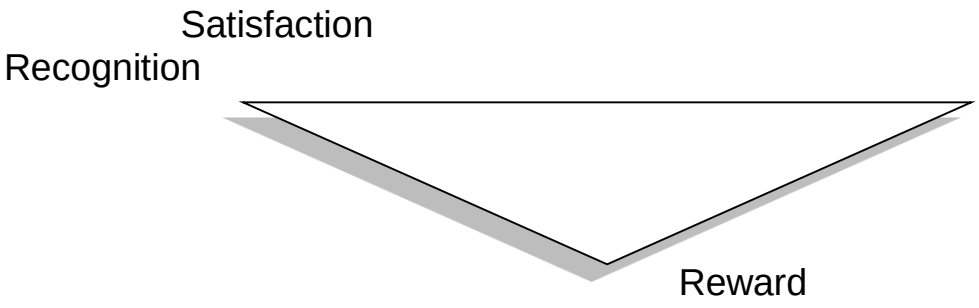


Figure 4-5 Satisfaction, recognition and reward model (SRRM)

4.5 Value Creation Model (VCM): Through Disruptive Technology (Innovation)

Software industry depends on the mental power, the core of the intellectual capital, one fuel of creating new innovative idea which may be transformed in to profitable and high quality software product, The two models have been added as new Independent (Predictor) Variables to the original assessment model for GSD which will have great impact on the outcomes of the project and on team mutual

Cooperation and specially on the quality and value creation of the software product what so called disruptive innovation.

In order to facilitate measurements of both quality and creation value of the software product, we suggest new attribute to the Dependent (Outcome) Variables in the original model, the new attribute is the Value Creation Model (VCM) which link the strategy of building the software product with innovation through cohesive relationship between three parameters Understand Opportunities (product strategy), Develop Technological Capabilities and Integrate People and Process, the Value creation model (VCM) depicted in Fig 4-6.

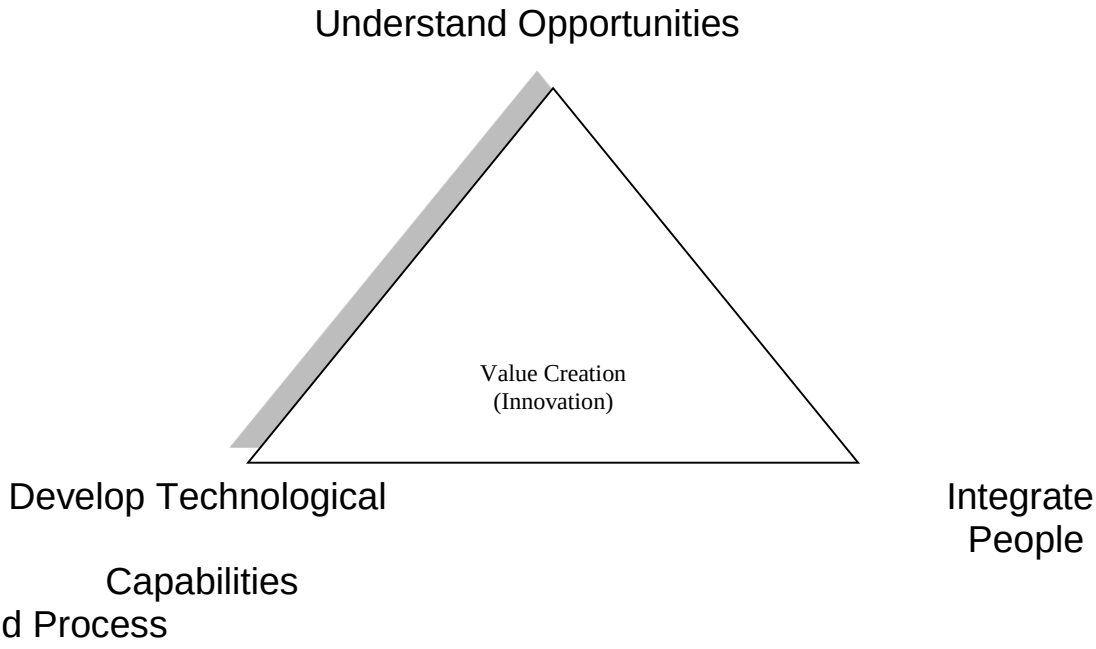


Figure 4-6 Value Creation Model (VCM): through Disruptive Technology (Innovation)

Table 4-1 shows the three models that are added, as propose to the original model and explain the advantages that can be added to the assessing the performance of GSD

Table 4-1: description the proposed models and explain the advantages that can be added to the assessing the performance of GSD

Model Name	Type of Variable	Names of attribute	Advantages for GSD Performance Assessment Model (GSDPAM)
CCCM	IV (Predictor)	1- Cooperation. 2- Collaboration. 3- Coordination.	1- The level of cooperation and collaboration make team member working together jointly in an intellectual endeavor and compliance for mutual benefit . 2- The third parameter coordination enables the virtual team working in harmonious functioning of parts for effective results
SRRM	IV (Predictor)	1- Satisfaction. 2- Recognition . 3- Reward .	1- New modification suggested by this work through adding an empower and motivation factor working with the (CCCM) model. 2- In order to maintain the good attitude of Virtual Team Members. 3- restorability of the management to keep the balances between the attributes to this model for the benefit of all stockholder .
VCM	DV (outcome)	1- Understand opportunities . 2- Develop Technological Capabilities . 3- Integrate People and Process.	1- New innovative idea which may transformed in to profitable and high quality software product. 2- Facilitate the measure both the quality and value creation of the software product. 3- link the strategy of building the software product with innovation through cohesive relationship between three parameters .

Chapter five

Conclusions and future work

5.1 Introduction

This chapter is the final chapter in this thesis which will be dedicated for the conclusions about the particular results with this work gain and a general observations and interpretation of deep analysis of several projects in various practices and solutions addressed by different GSD team then we present the future work as percepts and suggested by the researcher. The three pillars of effective infrastructure management for GSD as concluded by this work are (1) communication and collaboration; (2) methodology and productivity tools and finally (3) software performance assessing and management.

5.2 Conclusion

1. GSD is inherently difficult. Problems with coordination, motivation, and aligning of technologies, infrastructure, and processes often cause projects to grind to a halt. GSD is, however, a reality that is here to stay and so methodologies must be evaluated and new tools must obtained knowledge and experience to be gained and available applied to batter execution of such projects efficiently and successfully by geographically distributed teams. The work represents some of GSD challenges, and introduces the need for managing its complexities in an efficient in and structured manner to obtain better rewards inherent in GSD.

2. This work identified issues of concern in GSD projects and provided factors that are critical to the success of such projects. Then management of framework for assessing and evaluating current projects was discussed and modify the existing model by proposes two additional predictor models. And outcome model, the final framework presented.

3. Importance of coordination: are many issues experienced in GSD Difficulties that is related to communication and coordination. In GSD the teams separated by geographic distances that make Ad-hoc communication very difficult and costly. It is not often recognized how vital role informal communication (e.g., around lunch table or in hallway) plays in a software development effort, so a new independent (predictor) variable proposed under the name Cooperation, collaboration and coordination model (CCCM).

4. Importance of getting everyone involved and satisfied: the level of participation and understanding of all team members on a remote team and motivated by the GSD model play critical role in quality and value creation of the software product, so a new independent (predictor) variable proposed under the name of Satisfaction, recognition and reward model (SRRM)

5. Disruptive Technology (Innovation), Quality assurance and value creation practices are critical for the successful outcome of a global development project and affected directly by the two added predictors to the original model and in order to measure the impacts of such new variables, the work propose a new model for value creation under the name Value Creation Model (VCM).

6. Productivity Tools and Techniques: by examine several tools and techniques used for Communication either Synchronous or Asynchronous, Formal or Informal also Time zone difference problem available solutions withier a Time overlap, Different working hours, Weekend or holidays, we discover that this issue is not as crucial as the other issues just because most of the technological tools available part for free others with marginal cost for all GSD team members.

5.3 Future Work

1. The theoretical proposed framework need to be implemented and experienced with actual projects to measure the impact of the predictors on the outcome variables, and mainly for additional attributes proposed by this work.
2. The proposed models may be examined as standalone framework to measure the significance degree of dependency between independent and dependent variables by real data collected from chooses software

3. project developed globally.
4. In addition, there is a need for more research utilizing formal analytical methods on how work is managed and coordinated by using agile software development methodology, e.g. teams using virtual Scrum, and common sprint process.

5. Cloud computing is a very promising infrastructure and platform for GSD project due to the potential common capabilities provided by this model for the virtual teams development and management purpose, so farther researches and investigations on that direction.

References

[1] Jackson, M., 1997 The meaning of requirements, the annals of software engineering, special issue on requirements engineering, 3:5-21, http://d**on**key**onthe**edge.com/i/geoloc_s**of**tware_engineering.pdf (visited 7/3/2013) , web

[2] Sawyer, P., Sommerville, I., and Viller, S. (Pisa, Italy, July 1998) , Improving the requirements process, Proceedings of the 4th International workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'98),. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22. (visited 10/3/2013), paper

[3] Pressman , R . S., 2001 Software Engineering : A Practitioner's Approach 5th Edition http://www.huson.edu.jodr_mjab/SD/MC.pdf , book

[4] Sommervill, I., Sawyer , P., 1997. Requirement Engineering – a good practice guide . http://eu.wiley.com.

[5] Herbsleb, J.D.and Moitra, D. (2001).Global Software Development. IEEE Software، paper

[6] Carmel, E. (1999). Global software teams., http://dl.acm.org/citation.cfm?id=293968 (visited 10/3/2013) , book

[7] Hai Zhuge , 22 November 2001, Knowledge flow management for distributed team software development ,Elsevier, paper.

[8] Catarina Costa, Camila Cunha, Rodrigo Rocha, A. César C. França, Fabio Q. B da Silva and Rafael Prikladnicki , 13 April 2010 , Models and Tools for Managing Distributed Software Development: A Systematic Literature Review, http://ewic.bcs.org , paper .

[9] John Stouby Persson (August 2009).Managing Distributed Software Projects, http://www.academia.edu (visited 19/3/2013), phd thesis .

[10] Julia Kotlarsky (16 June 2005), Management of Globally Distributed
Component-Based Software Development Projects , http://www.erim.eur.nl , (visited 2/3/2013), thesis .

[11] Daniela E. Damian , Didar Zowghi , (2003), Requirements Engineering challenges in multi-site software development organizations1 , Requirements Engineering Journal, 8, pp. 149-160, 2003, paper

[12] Daniela E. Damian University of Technology, , NSW 2007, Australia . The study of requirements engineering in global software development: http://citeseerx.ist.psu.edu (visited 28/2/2013) , paper .

[13] Daniela Herlea and Saul Greenberg (1998), Using a Groupware Space for Distributed Requirements Engineering , IEEE. Paper

[14] Leandro Lopes, Rafael Prikladnicki, Jorge Audy , Requirements Specification in Distributed Software Development – A Process Proposal , School of Computer Science – PUCRS http://www.**inf**.**pucrs**.br/munddos/docs/HICSS2005_**leandro**.pd , paper

[15] Rafael Prikladnicki, Jorge Audy, Roberto Evaristo . (2003) Requirements Management in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context, http://citeseerx.ist.psu.edu/viewdoc (visited 10/3/2013), paper .

[16] Wesley James Lloyd , Mary Beth Rosson , James D. Arthur . (2002), Effectiveness of Elicitation Techniques in Distributed Requirements Engineering , IEEE, paper .

[17] Manning S, Lewin AY. 2008. A dynamic perspective on next-generation offshoring: the global sourcing of science & engineering skills. Academy of Management Perspectives 22: 35-54 http://papers.ssrn.com . paper

[18] Naoufel ben Ahmed Boulila , 10.10.2005 , A Framework for Distributed Collaborative Software Design Meetings , Dissertation.

[19] Paul Salmon , Neville Stanton and Robert Houghton , 2/ 08 September 2008 , Developing Guidelines for Distributed Teamwork: Review of the Literature and the HFI DTC's Distributed Teamwork Studies , Paper .

[20] Miguel Jim´enez, Mario Piattini, and Aurora Vizcaino , 6 March 2009 , Challenges and Improvements in Distributed Software Development: A Systematic Review , Publishing Corporation Advances in Software Engineering Volume 2009, Article ID 710971, paper .

[21] Michele Durham , June 15,2007 project management and distributed team , research paper .

[22] Maria Paasivaara Nico Hiort af Ornäs Peitsa Hynninen Casper Lassenius Tuomas Niinimäki and Arttu Piri , 2010 , Practical Guide to Managing Distributed Software Development Projects , paper .

[23] Linda Rising and Norman S. Janoff , July 2000 , The Scrum Software
Development Process for Small Teams , IEEE, Paper .

[24] M. Rita Thissen, Susan K. Myers, and R. Nathan Sikes , 2011, Globalization of Software Development Teams, paper .

[25] Jukka K , 2011 , Towards an Application Lifecycle Management Framework , (URL: http://www.vtt.fi/publications/index.jsp) , Dissertation

[26] Roger M. G. Dols and Quintin Siebers , 2009 , idSpace: Distributed Collaborative Product Innovation. Paper .

[27] Peter Van Rosmalen, Jo Boon, Marlies Bitter-Rijpkema, and Rory Sie, The idSpace creativity support platform: evaluating for cocreating , paper

[28] Thomas Heider (LiNK), Rani Pinchuk (SAS), Lynn De Proft (SAS),
Marlies Bitter (OUNL), Jo Boon (OUNL), 31-12-2008, Evaluation plan & planning , Tooling of and training for collaborative, distributed product
Innovation idSpace , paper .

[29] MORPH , 11-11-2009 , idSpace platform & user guide , Tooling of and training for collaborative, distributed product innovation, paper.

[30] Birgit R. Krogstie, The wiki as an integrative tool in project work, paper

[31] Louridas, P. 2006., Using Wikis in Software Development. IEEE Software, paper .

[32] Chao, J. 2007. Student Project Collaboration using Wikis. in 20th Conference on Software Engineering & Training (CSEET'07). Paper

[33] Slotter, E. B. (2010). Using Wiki contributions to induce collaborative learning in a psychology course. International Journal of Technology in Teaching and Learning, paper.

[34] Lisa Dyer and Anne Gentle , Using Structured Wikis in Software Engineering , paper .

[35] Carolyn Wei, Brandon Maust,and Jennifer Barrick, 8-11-2005 , Wikis for Supporting Distributed Collaborative Writing , STC , paper

[36] IBM Corporation 2012 , IBM SmartCloud Engage Advanced , Integrated email, collaboration and online document editing from the IBM cloud , paper .

[37] Nader Ale Ebrahim, Shamsuddin Ahmed and Zahari Taha , 2009 , Virtual Teams: a Literature Review, Australian Journal of Basic and Applied Sciences , paper .

[38]https://www.google.com/search?hl=en&site=imghp&tbm=isch&source=hp&biw=1241&bih=593&q=communication+team+in+Co+located&oq=communication+team (visited 13/7/2013)

[39] Brian Nicholson ,and Sundeep Sahay , 2004 , Embedded knowledge and offshore software development , paper , Elsevier .

[40] http://www.bristol.ac.uk/expsych/people/markus-f-damian/index.html (visited 20/7/2013)

[41] H. Keith Edwards and Varadharajan Sridhar , 2002, Analysis of the Effectiveness of Global Virtual Teams in Software Engineering Projects ,paper , IEEE.