

# STUDY THE IMPACT OF REQUIREMENTS MANAGEMENT CHARACTERISTICS IN GLOBAL SOFTWARE DEVELOPMENT PROJECTS: AN ONTOLOGY BASED APPROACH

S. Arun Kumar<sup>1</sup> and T.Arun Kumar<sup>2</sup>

<sup>1</sup>Assistant Professor, SCSE, VIT University, Vellore, Tamil Nadu, India  
*sarunkumar@vit.ac.in*

<sup>2</sup>Professor, SCSE, VIT University, Vellore, Tamil nadu, India  
*tarunkumar@vit.ac.in*

## **ABSTRACT**

*Requirements Management is one of the challenging and key tasks in the development of software products in distributed software development environment. One of the key reasons found in our literature survey the failure of software projects due to poor project management and requirement management activity. This main aim of this paper 1. Formulate a framework for the successful and efficient requirements management framework for Global Software Development Projects. (GSD) 2. Design a Mixed organization structure of both traditional approaches and agile approaches, of global software development projects. 3. Apply Ontology based Knowledge Management Systems for both the approaches to achieve requirements issues such as missing, inconsistency of requirements, communication and knowledge management issues and improve the project management activities in a global software development environment. 4. Propose requirements management metrics to measure and manage software process during the development of information systems. The major contribution of this paper is to analyze the requirements management issues and challenges associated with global software development projects. Two hypotheses have been formulated and tested this problem through statistical techniques like correlation and regression analysis and validate the same.*

## **Keywords**

*Requirements Management (RM), Ontology, Requirements Management Metrics, Knowledge Management (KM), Global Software Development (GSD).*

## **1. INTRODUCTION**

In the software engineering literature project management and requirements management activities are crucial in order to achieve the high quality software system. Due to increase the globalization of software development deriving many benefits to the software companies at the same time it will leads to several problems/issues occurring while managing the requirements in global software development projects which includes [2]

1. Understanding the requirements among different groups of people (geographical distributed development teams).
2. Problems in deals with frequently enhancing the requirements.
3. Cross culture relationship, communicational issues, and knowledge management issues.
4. Coordination among employees.

Traditional approach at the global level, be agile approach at the local level in organization i.e., mixed organization structure is proposed to handle these concerns. To apply ontology based KM practices to the proposed organization structure is helps to reduce the various requirements management issues in global software development projects. Requirements engineering activity is broadly divided in to two categories first phase is requirements definition which deals with elicitation, analysis, documentation and review and second phase is requirements management which deals with change management and traceability. Our main aim is to working in the requirements management phase in distributed development projects. Propose requirements management metrics to measure the requirements changes, requirements schedule, and average rate of defects during the development of information systems in a multi site environment. This paper mainly covers the following application areas software process improvement, requirement management, software project management and software measures is clearly mentioned in fig.0

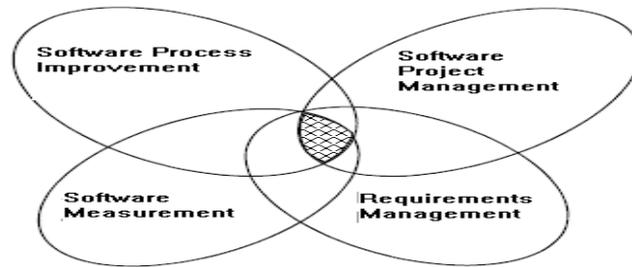


Figure 0: Major Application Areas

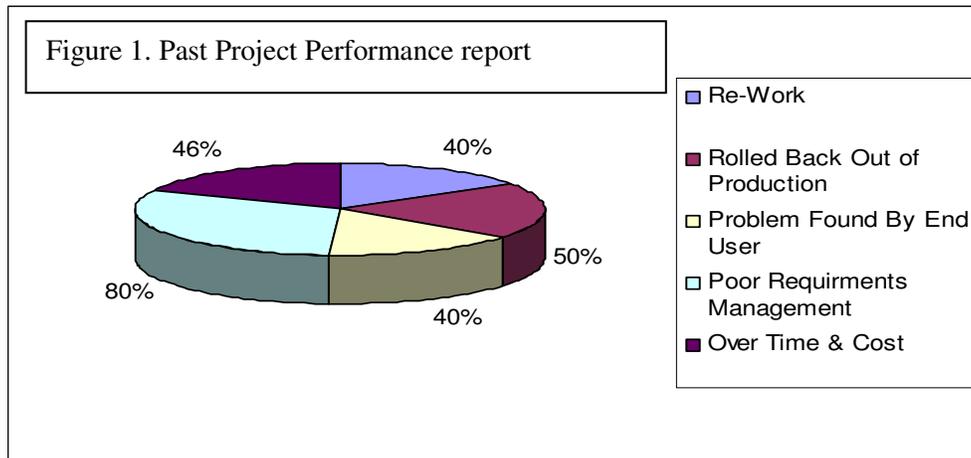
## 2. REQUIREMENTS MANAGEMENT – CURRENT SURVEY

### 2.1 CIO Magazine

Analysts report said 71% of software projects that fail due to **poor requirements management**, making it the single biggest reason for project failure. [16]

### 2.2 Standish Report

The Standish CHAOS Report, which surveyed 9,236 IT projects, found that the top three causes of project failure were **lack of user input, incomplete requirements or changing requirements**. According to the Standish Group International CHAOS Survey – U.S.A past project performance report is shown in fig.1.



According to the Standish Group International CHAOS Survey – U.S.A the Factors for Project Failure Deal with Requirements is shown in fig.2.

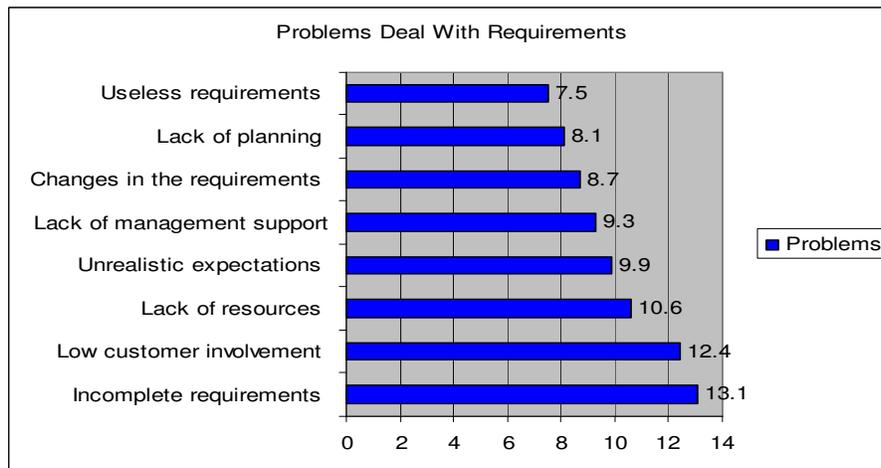


Figure 2: Project Failure Due to Requirements Factor

### 2.3 Survey of European Software Organizations

A recent survey of European software organizations identified that more than 40% perceived that they had major problems in managing customer requirements.

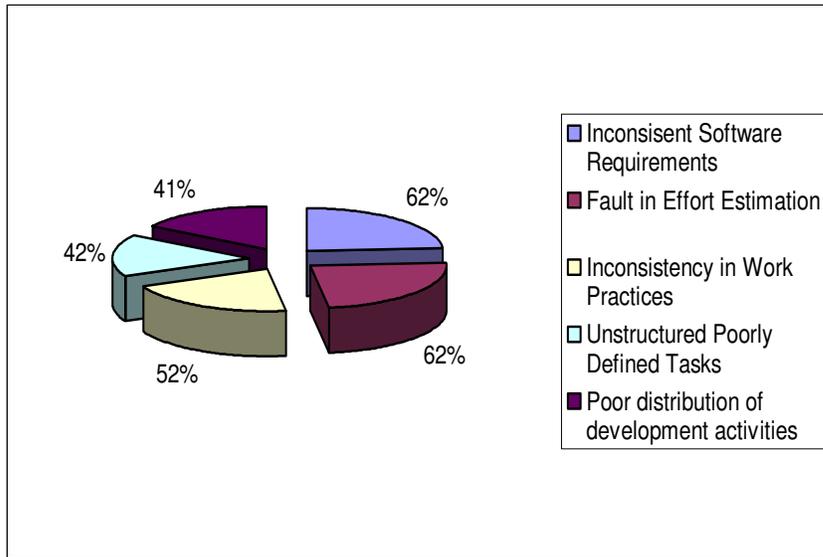


Figure 3: TOP 5 Threads in RM [15]

### 3. REQUIREMENTS MANAGEMENT IN GSD ISSUES

The survey said there number of problems arising in requirements management in global software development projects is shown in Fig.1. The surveys gives 44% to 80% of all defects are found in the requirements phase [7]. The software projects regularly fail due to problems with requirements. Frequent changes in requirements may result in incomplete, wrong, ambiguous requirements. The poor requirements management is leads to increase overall cost, decrease quality of the system or fail altogether.

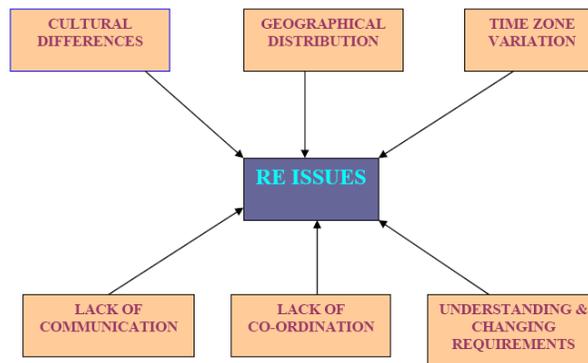


Figure 4: RM Issues in GSD Projects

### **3.1 ISSUES IN GLOBAL SOFTWARE DEVELOPMENT PROJECTS**

There are number of issues identified in GSD through our literature review

- Different conception and terminology is used among the team members about the technical as well as managerial aspects of software engineering principles.
- Difficult to ensure common understanding of problem domain among the development teams.
- Different level training, knowledge and skills that varies among the development teams.
- Delays in getting status reports and concern problem raised not solved on time.
- All these issues occurred in GSD due to one of the main root cause: poor communication – thus, poor sharing and transfer of knowledge among the development teams.

## **4. RESEARCH QUESTIONS AND METHODOLOGY**

The research question that motivated through our study was R1: “What are the basic problems and challenges for managing requirements in the distributed setting?” our main aim is to realize how requirements are managed in global software development projects; to identify the challenges occurred, policies and technologies are used to overcome these challenges. R2: “What is the ontology based Requirements management system in Global Software Development (GSD) environment?” The software engineering ontology’s assist communication and knowledge management issues like knowledge sharing and knowledge transfer among the development team in GSD. R3: “What are the main reasons that brought up the Ontology Driven Development in GSD?” Ontology driven development can help to improve the way in which software development projects are organized in distributed setting. Ontological domain models can refine classical life cycle phase such as requirements, design and implementation. R4: “Testing of ontology based Requirements management system in multi-site software development environment.” ontology based experimental model as part of the communication and knowledge management frame work to facilitate performance and other related issues to develop high quality software project under GSD. Our proposed research design and methodology is shown in fig.5.

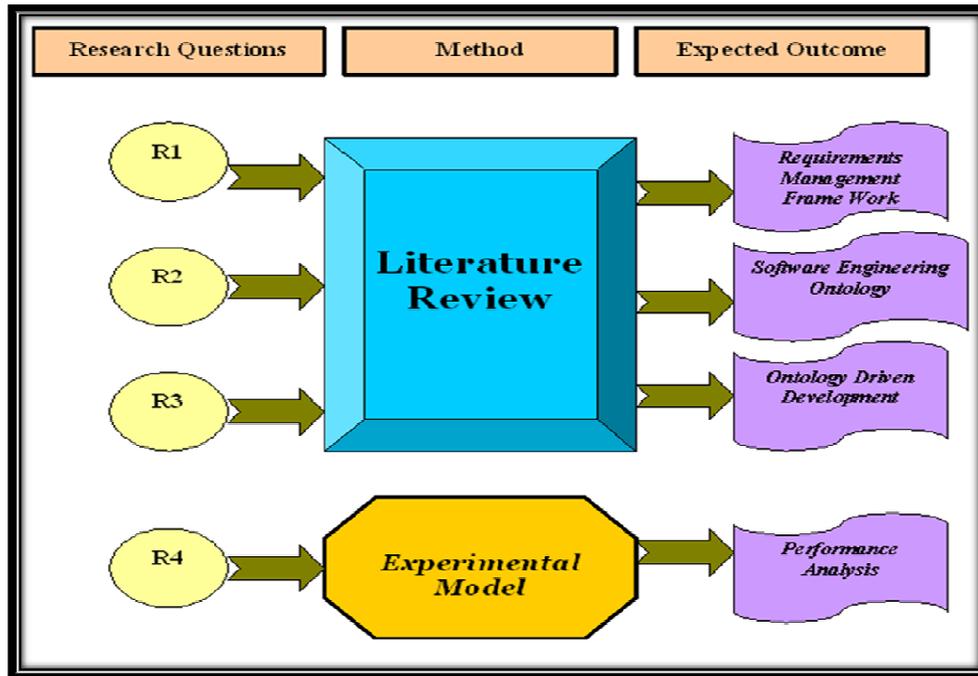


Figure 5: Research Design and Methodology

## 5. REQUIREMENTS MANAGEMENT FRAME WORK

The researcher will collect the materials related to the key areas of requirements management with respect to process. Through comparative study will be carried out to achieve an integrated approach for the software development organization. Diagrammatic representation of the proposed organization structure is shown Fig.6.

Our proposed work mainly focus on mainly focus on [2]

Phase I Analyze the performance issues in traditional approach and agile approach in global software development.

Phase II Design an optimized framework model to integrate traditional approach at the global level and agile approach at the local level of a organization structure which is arrived at phase-I. To solve the issues are mainly frequently to enhance the requirements, missing requirements, and inconsistency among requirements in GSD projects.

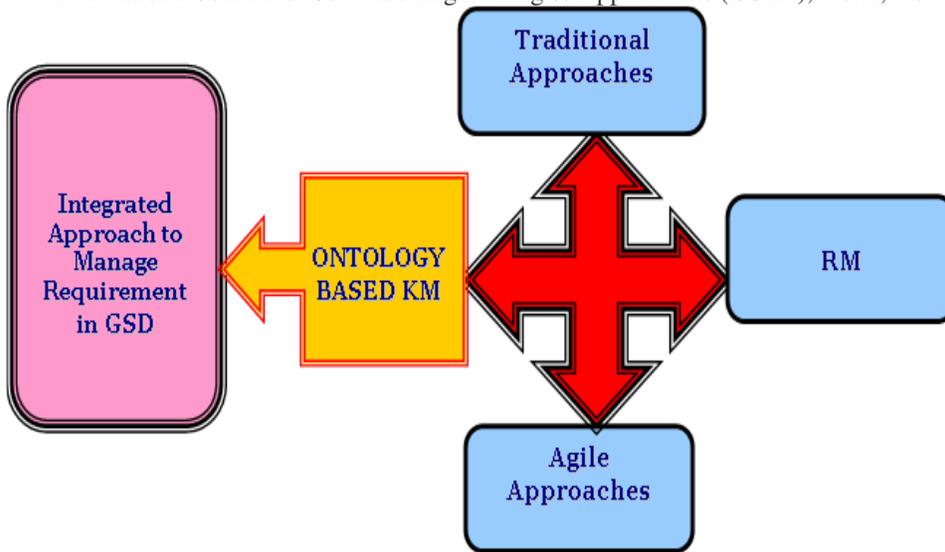


Figure.6 Requirements Management Framework

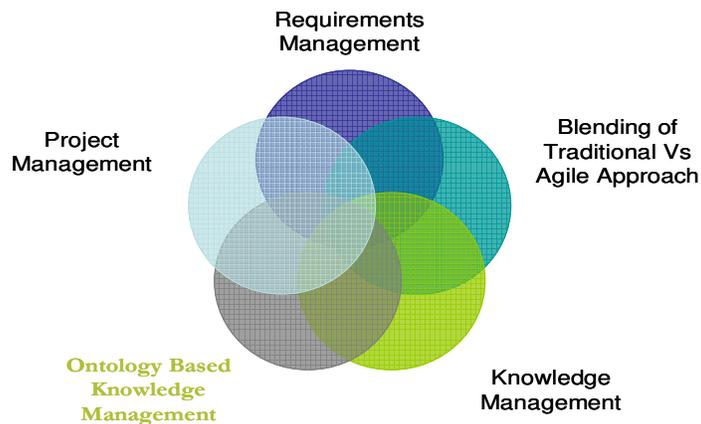


Figure 7: Research Approach

Phase III Design Ontology based knowledge management frame work is formally representing the application domain knowledge and formal representation of requirements in GSD.

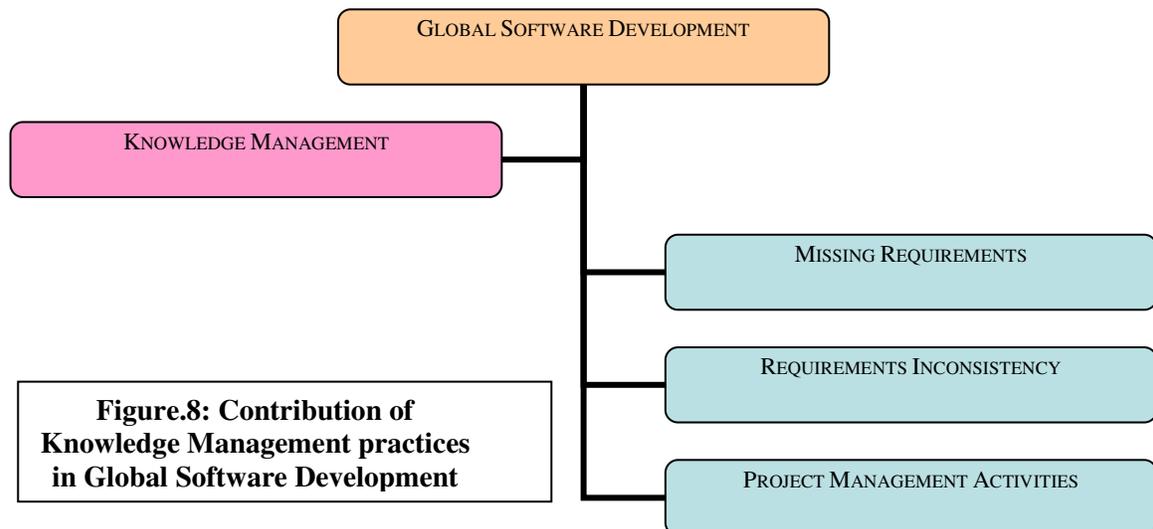
Phase IV Propose requirements metrics to measure the requirements changes, average rate of defects and requirements schedule in GSD projects.

Phase V Formulate the hypotheses and tested the requirements management metrics using statistical techniques (correlation and Regression Analysis) and validate the same.

## 6. KNOWLEDGE MANAGEMENT ISSUES IN GLOBAL SOFTWARE DEVELOPMENT

Knowledge management is all about the people to test how the people are working towards in order to achieve business and organizational objectives. Knowledge management is defined as “delivering the right knowledge to the right persons at the right time.” To apply ontology based knowledge management system to the proposed organization structure the global software development deriving many benefits this includes: [2]

- KM helps to ensure the right information and to make right decision.
- On time to delivery of the system
- Increase the profitability
- To improve the project management and communication and knowledge among team members.
- KM helps to reduce overall development time and cost.



Global software development project really requires effective knowledge management techniques to fulfill the user expectations. Ontology based knowledge management techniques are taking major initiative for the success of the geographically distributed software development projects. Ontology Knowledge management mainly focusing on knowledge sharing of globally distributed development teams, cross culture issues, communication issues (meet time difference time zone and distance) and technical issues such as lack of synchronization.

Our main aim is successfully manage and resolve these issues in distributed software development environment. Our proposed model is framing an effective organization structure in a current software industry practice.

## **7. ONTOLOGY BASED KNOWLEDGE MANAGEMENT SYSTEM**

Global Software Development projects are facing number of challenges like time differences, language diversity, cultural issues communication and knowledge management issues. Knowledge transfer and sharing is not easy when team members are working in different development sites. The common understanding of problem domain is crucial when the team members do not have the common language. In multi-site software development the people sharing the knowledge and communication in common language (English) but stake holders are come from different country and regions idiomatic difference one of the major challenge for communication takes place in GSD. For instance, people from United States and England share English as their own native language but the way of pronouncing the terms may varies from one to another and many words have different meanings. So this is critical to find the common understanding of the problem domain. So our main aim is to overcome the knowledge management and communication issues to minimize the idiomatic differences we have use ontology based knowledge management system. When the team member is come from different parts of country to share common terminology and concept about the problem domain is very difficult for that our aim is to use domain ontology to solve these communication and knowledge management issues. Our proposed Ontology based knowledge management system is giving clear picture about the problem domain and its associated concepts and terms related to the problem.

### **7.1 Benefits on the Use of Ontology in GSD [8]**

- Ontology's provide a representation vocabulary specialized for the software process, eliminating conceptual and terminological mismatches.
- The use of ontology's and alignment techniques allows solving compatibility problems without having to change existing models.
- Ontology's might help to develop benchmarks of software process by collecting data on the Internet and the use of the Semantic Web.
- Ontology does allow both to transfer knowledge and to simplify the development cycle from project to project.
- Ontology's promote common understanding among software developers, as well as being used as domain models.
- Ontology allow for an easier knowledge acquisition process, by sharing a same conceptualization for different software applications.
- Ontology's allow reducing terminological and conceptual mismatches, by forcing to share understanding and communications among different users during the ontological analysis.
- Ontology's also providing for a refined communication between tools forming part of an environment.
- Ontology's, when as machine-understandable representations, help in the development of tools for software engineering activities.

## 8. TRADITIONAL APPROACH VERSUS AGILE APPROACH

The survey said 1,027 projects in the United Kingdom 13% of projects did not fail, and waterfall-model was the one of "major contributing factor for failure, problem being addressed in 82% of the projects. The United States Defense Department projects concluded that "46% of the systems really did not meet the real needs as well as user specified requirements, that they were never successfully used, and another 20% of projects are required for extensive rework" to be usable. The Analysis Center for Software [12] and software development researchers in Israel [13] said agile methods are 29% better cost, 91% better schedule, 97% better productivity, 50% better quality, 400% better satisfaction, and 470% better return on investment compare than traditional approaches. The survey said traditional approaches improve the quality and productivity of the software system incase of agile approach satisfy the real needs of the customers and project success. The requirements are keep on changing while the project is progress. The requirements changes due to improve the effectiveness of the system to satisfy the real needs of the users. It is difficult to respond the customer changes once the requirements have been accepted in the traditional approaches but agile approach is adoption of changing requirements dynamically even the project is progress. Our proposed model is creating mixed organization structure blending of both traditional approaches at the global level to finalize the requirements for system development and be agile approach at the local level to elicit, analyze, and review the software requirements dynamically. The reason beyond to go for this mixed organization structure is now a days traditional way of software development has faced several challenges/issues like understanding ,changing requirements in a multi site environment , the agile approach successfully to manage the challenges in a global software development environment. To implement both traditional and agile method in an organization, to take the benefits of these two methods to improve accuracy, productivity, and quality of global software development projects. To apply Ontology based KM practices for both the approaches to achieve requirements management issues such as missing requirements and inconsistency, communication and knowledge management issues and improve the project management activities in a global software development projects. We propose requirements management metrics to measure the requirements changes, average rage of defects and requirements schedule in order to improve effectiveness of the software development process.

The survey said agile software development deriving many benefits to the software companies which includes: [2]

- Early and continuous delivery of software system.
- Dynamically keep on changes in the requirements even progress in the development.
- More on customer and end users involvement towards system development.
- Return on investments in earliest manner.
- Meet the stakeholder's expectations.
- Simplicity and visibility of actual progress of the project is available.
- Face-to-Face conversation.
- Risk reduction in the overall development of the software system.

## 9. REQUIREMENTS MANAGEMENT MEASURES

Requirement is need of customer. Requirements are capabilities and intention to any product or service must conform which is common to all development of the system and other engineering activities. The term “engineering” also suggests that the outputs of a requirement engineering process need to be carefully engineered, where those “outputs” are usually understood to be detailed specifications. [9].

Requirements engineering is defined as follows: *“Requirement engineering (RE) is a set of activities concerned with Identifying and communicating the purpose of a software-intensive system, and the context in which it will be used. Hence, RE acts as the bridges between the real world needs of user, customer, and other constituencies affected by a software system, and the capabilities and opportunities affected by software-intensive technologies”*. [9]

Requirements management is defined as follows: *“Requirement management is a set of activities that helps the project team to identify, control, and track requirements and changes to requirements at any time as project proceeds”*. [10]

Software metrics used in requirement management process to provide the information, needed to take key project decisions and to take suitable actions. Metrics are used for continuous project improvement and control in requirement management process. The general and comprehensive set of requirement management measures is defined for implementation of the goals of the requirement management KPA (Key Process Area) within the SW-CMM level2 is mentioned in table 1. [11]

<b>Reference</b>	<b>Type of Measures</b>	<b>Attributes</b>	<b>Entity</b>	<b>Theoretical /Empirical Valid</b>	<b>Environment</b>
Loconsole and Böstler [3,6]	Total number of requirements, number of Initial, current, and final requirements, status of requirements, number of changes per requirement, status, type, reason, and cost of change to requirements	Size and status of requirements specifications, size and status of changes to Requirements	Requirements management	Kitchenham representational theory	Academic case study
Loconsole and Böstler [4]	Num lines, words, actors, use cases, subjective volatility Volatility (num changes, major, moderate, minor, revisions)	Requirements specifications	-	Spearman correlation	Industrial software
Ambrisola and Gervasi [5]	Measures of stability (amount of information contained in requirements at time t) and efficiency	Amount of rework, perceived work efficiency	Requirements analysis process	Plot of data	Academic experiment
Our proposed Methodology[2]	Number of Requirement Changes, Average Rate of Defects, Requirements Schedule	Changes to requirements, Schedule Slippage, and rate of defects	Requirements Management	Pearson Correlation coefficient and Regression Analysis	Web Based Project
Table 2. Measures Defined and Validated this Paper					

Table 1. Requirements Management Measures [11]

S.No	Measurement Type	Measure	Entity
1.	Direct	Requirements affected by Change	Requirement management process
2.	Direct	Inconsistent requirements	Requirement management process
3.	Direct	Missing requirements	Requirement management process
4.	Direct	Incomplete requirements	Requirement management process
5.	Direct	Initial requirements	Requirement management process
6.	Direct	Final requirements	Requirement management process
7.	Indirect	Requirement schedule	Requirement management process

## 10. REQUIREMENTS MANAGEMENT METRICS

We propose a set of metrics for requirements phase. The purpose of this metrics is in order to improve the effectiveness of the software development process. [2]

**Actual Requirements Count (ARC):** The numerical count of total number of original requirements is present in the project.

**Number of New Requirements (NNR):** The numerical count of the actual number of requirements in the project, compared to the count of the number of new requirements introduced in the project cycle: analysis, design, code, test, and implementation.

$$\text{NNR} = \text{Number of Original requirements} + \text{newly added requirements} - \text{Deleted requirements}$$

**Number of Requirements Changes (NRC):** To calculate total number of requirements change requests (insert/delete/update) during the project cycle.

$$\text{NRC} = \text{newly added requirements} + \text{deleted requirements} + \text{revised requirements.} \\ (\text{ARC} - \text{Number of Requirements changes (insert/delete/update)})$$

**Average Rate of Requirements (ARR):** The average number of requirements change requests required in the entire life cycle of the project.

$$\text{Number of Requirements changes (NRC) / Actual Requirements count (ARC) * 100 per phase.}$$

**Average Rate of Defects (ARD):** The total number of defects is detected in each phase of the project per month.

**Requirements Schedule (RS):** The numerical count total effort is required to complete the task.

Actual completion date- Estimate completion date / actual effort

## 11. METHODOLOGY AND DATA ANALYSIS

In this paper a set of requirements management measures has been presented and validated based on table 1 and table 2.

Our observation is that the majority of existing measures are neither theoretically nor empirically validated for different academic case studies and industry case studies. Table 3 provides the data were collected based on web based project according to this data we have measured and validate via the statistical analysis and find correlation coefficient and forming regression model of NRC, ARD and RS.

The main aim of this study is to establish the importance of requirements management metrics for global software development projects. We use correlation and multiple correlations, regression analysis to test the two hypotheses formulated above to study the impact of one metric over another.

Table 3

Project	NRC	ARD	RS(Days)
A	600	792	210
B	750	862	265
C	816	937	350

We have formulated the following two hypotheses focusing our study and to assist the statistical analysis: [2]

**Hypothesis 1:** The metric NRC influences the metric average rate of Defects (ARD)

**Hypothesis 2:** The metrics NRC, ARD decide the requirements schedule (RS) in person-hours.

Correlation is a statistical measurement of the relationship among two variables. The correlation ranges from +1 to -1. There is no relationship among the variables then the correlation is said to be zero correlation. If one variable goes up, and another variable goes down then the correlation is said to be perfect negative correlation(-1). A correlation of +1 denotes a perfect positive correlation; it means the both variables progress mutually in the same direction.

Regression analysis is a statistical method where the mean of one or more random variables is predicted based on other measured random variables. Regression analysis is mainly used to understand how the actual value of the dependent variable changes when any one of the independent variables is varied. Here RS metric is dependent variable of NRC metric and ARD metric.

We have used IBM SPSS analytical software to find Pearson coefficient of correlation and regression analysis of above requirements metrics. [2]

Based on the available data shown in Table 3 provides the NRC, ARD and RS values from a web based projects. The coefficient of multiple correlations 'R' between NRC ( $r_1$ ) versus ARD ( $r_2$ ), NRC ( $r_1$ ) versus RS ( $r_3$ ), and ARD ( $r_2$ ) versus RS ( $r_3$ ) is computed below

- $R_{12} = 0.971$
- $R_{13} = 0.941$
- $R_{23} = 0.995$

The value of 'R' is greater than zero is denotes a perfect positive correlation between this three variables. Here the relationship between two variables (NRC metric and ARD metric) change in one variable result in a positive or negative change in the other variable and also greater change in NRC metric result in corresponding greater change in other metric ARD and RS.

Say  $NRC = x_1$ ,  $ARD = x_2$ ,  $RS = x_3$  when RS is dependent on NRC, ARD then the regression model will be

$RS_{\text{predict}} = -7.1429 + 1.376x_2 - 0.275x_1$
---

The RS metric is predicted based on NRC metric and ARD metric which is clearly stated in the above regression model. The regression  $R^2 = 1$  correlation coefficient determination = 1 so this regression model is perfect fit for the above requirements metrics. Through this data analysis gives a great deal of support to our two hypotheses formulated in this study. The clearest result observed from the value  $x_1$ ,  $x_2$  and  $x_3$  is the strong relationship between NRC metric influence ARD metric and NRC and ARD metric decide the RS metric.

## 12. CASE STUDY: INDIAN RAILWAY SYSTEM

The main purpose of this case study we find the benefit of using ontology in development of information system. The main reasons are to use ontology in our Railway Information System (RIS) is to retrieve the information and compare with traditional development of railway information system. Ontology is to analyze the domain knowledge, to enable the reuse of domain knowledge, to make the domain assumptions explicitly, to separate the domain knowledge from the operational knowledge and to share the common understanding of the structure of information among the people or the software agents.

Indian Railway's network is the second largest railway network in the world. For easy operation and management, Indian Railway is divided into fifteen zones and each zone manages its own local affairs. Southern railway is one among the fifteen zones of Indian railway. If a person wants to travel from one place to another place by train, he or she has to plan for the shortest route, minimum traveling time or cost effective route and the details of the trains operating in the identified route for the comfort of his journey.

A new form of web content that is meaningful to computers will unleash a revolution of new abilities. The ontology's which contain terms and relationships between terms that have been agreed upon by members of a certain domain.

We have find ontology has created a systematic way for railway operation management and reduction of human intervention or controlling activities and performances. Railway information systems have emerged in the last decade as an essential tool for urban and resource planning and management.

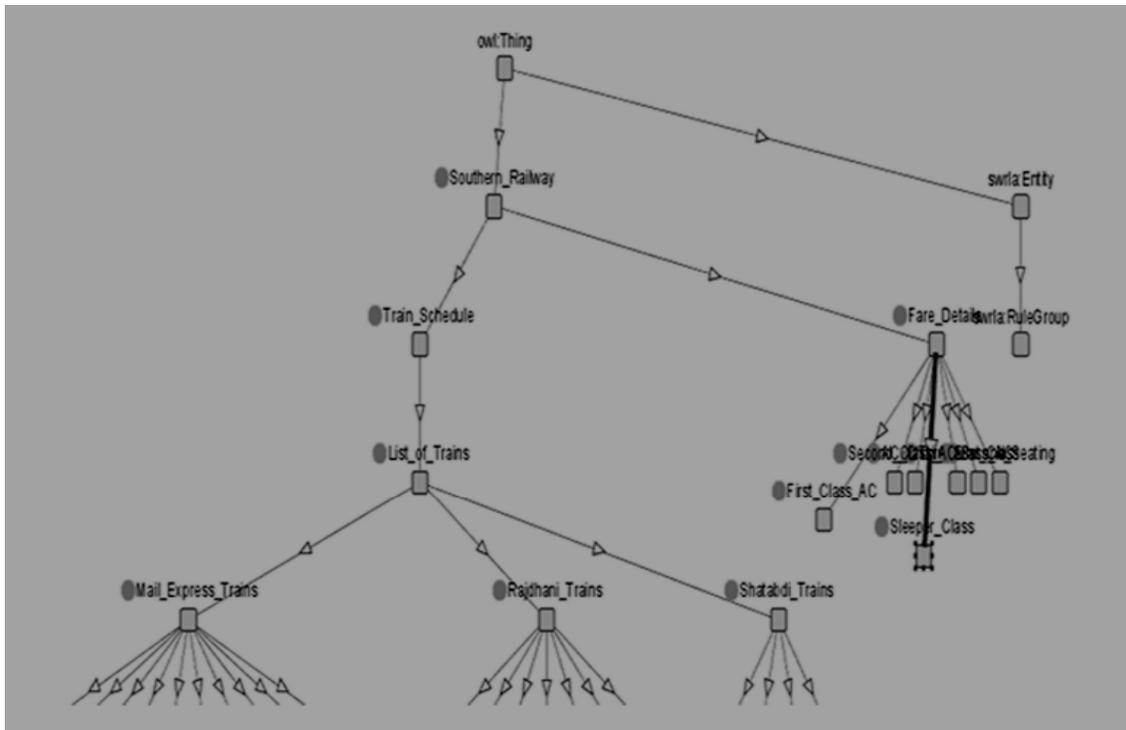
Their capacity to store, retrieve, analyze, model and map large areas with huge volumes of spatial data has led to an extraordinary proliferation of applications. RIS allows the creation of maps through automated mapping, data capture and surveying analysis tool. It can be used to find out what's occurring within a set distance of a feature.

Finally, creating maps include precise details about train crossing and train transfers After creating maps by RIS, then link these maps to railways Database. RIS benefits organizations of all sizes and in almost every industry. There is a growing interest in and awareness of the economic and strategic value of RIS, in part because of more standards based technology and greater awareness of the benefits demonstrated by RIS users. The number of RIS enterprise solutions and IT strategies that include RIS are growing rapidly.

We have used the ISO standard for business processes as the fundamental agreed knowledge, known as Business Ontology, for the development teams. The class tree representation of ontology based RIS in shown in fig.9. The class tree representation is help to standardize a) the domain knowledge about the business operations and workflow, b) vocabulary and concepts used and definition of them across the cultures. [1]

Through this case study the outcomes of ontology based RIS generally fall into six basic categories

1. Cost savings resulting from greater efficiency.
2. Better decision making.
3. Improved communication.
4. Better geographic information record keeping.
5. Managing geographically.Optimized query.



6.

Figure 9: Class Tree Representation of RIS Ontology

We have analyzed the performance issues of traditional railway information system and compare with ontology based railway information system which is clearly mentioned in table 4.

SI No.	Traditional Railway Information System	Ontology based Railway Information System
1.	In the traditional railway information system, the users have to create many queries to retrieve train information of just one specific train.	A new form of web content that is meaningful to computers will unleash a revolution of new abilities making information retrieval lot easier.
2.	The results of the queries are based on the syntactic nature of data.	The results of the queries are based on the semantic nature of data.
3.	A semi-illiterate human being finds difficulty in reading the railway time table	This problem can be effectively handled through semantic technology.
4.	The traditional railway information system is a weak integrated system.	But it can be improved by using ontology as, a good robust integrated system is developed, which is supported by a middleware infrastructure.
5.	Maintenance was straightforward but inefficient.	Ontology Provides easy maintenance as railway system is divided into sub classes.
6.	Risk Analysis excluded.	Ontology Based Railway Information System predicts faults and failures.
7.	Component Separation is not visible.	Communication between components is analyzed.

Table 4. Traditional Versus Ontology Based RIS

### 13. CONCLUSION

Though there are number of requirements engineering models being practiced in most of the organizations, no unified model, organization structure exists currently for implementation in managing requirements while developing a software product in distributed setting. Ontology based KM methods; tools are required to achieve better requirements management practice in global software development projects. The proposed requirements management framework would facilitate the organizations to effectively manage requirements engineering issues into various levels that will result in forming a benchmarked approach.

An integrated approach and a validated model for implementing the requirements management framework involve managing the requirements in GSD in an effective manner. To analyze some of the issues related to this problem currently arising in global software development project having some remarks: reduce cost, increase productivity manage, measure the requirement changes, average rate of defects and requirements schedule , knowledge and communication related problems and improve product and services.

The conclusion of this paper based on statistical analysis of requirements metrics and benefit of proposed organization structure in a multi-site environment. In our future research is based on development of software metrics to reflect on the quality factors to improve the software development process and we are planning to implement ontology based knowledge management

International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011 system for various application domain like e-health and e-learning system, and to measure various factors (knowledge sharing, trust, team work and requirements flow down) among offshore and onsite team members related to the GSD projects [17] in our future research and analyze the performance issues.

## 14. REFERENCES

- [1] P. Wongthongtham , E. Chang a, T.S. Dillon , I. Sommerville (2006). “Ontology-based multi-site software development methodology and tools”, Journal of Systems Architecture 640–653.
- [2] S.Arun Kumar, T.Arun Kumar, (2011) “Characterization and Validation of Requirements Management Measures using Correlation and Regression Model”, International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 2, April 2011.
- [3] Loconsole, A. and Borstler, J. 2005. An Industrial Case Study on Requirements Volatility Measures. In Proceedings of the 12th Asia-Pacific Software Engineering Conference (Dec. 15–17, 2005). IEEE Computer Society, Washington, DC, 249–256.
- [4] Loconsole, A. and Borstler, J. 2005. An Industrial Case Study on Requirements Volatility Measures. In Proceedings of the 12th Asia-Pacific Software Engineering Conference (Dec. 15–17, 2005). IEEE Computer Society, Washington, DC, 249–256.
- [5] Ambriola, V. and Gervasi, V. 2000. Process Metrics for Requirements Analysis. In Proceedings of the 7th European Workshop on Software Process Technology (Feb. 21–25, 2000). R. Conradi, Ed. Lecture Notes In Computer Science, 1780. Springer- Verlag, London, 90–95.
- [6] Loconsole, A. and Börstler J., (2007) A Correlational Study on Four Size Measures as Predictors of Requirements Volatility, Submitted to Journal of Software Measurement.
- [7] Nafon Assawamekin.(2010) An Ontology-Based Approach for Multiperspective Requirements Traceability between Analysis Models, 9th IEEE/ACIS International Conference on Computer and Information Science.
- [8] Ontology-based Software Engineering ASLab-ICEA-R-2006-016 v 0.1 Draft of 2006-11-15 <http://www.aslab.org/documents/ASLab-ICEA-2006-016.pdf>
- [9] Steve Easterbrook, “What are Requirements”,2004. <http://jasonnolan.net/kmd1002/easterbrook.pdf>.
- [10] Pressman S Rogers, (2001) “Software Engineering, A Practitioner’s Approach”, Fifth Edition, McGraw Hill Higher Education.
- [11] Loconsole Annabella, (2001)“Measuring the Requirements Management Key Process Area”, Proceeding of ESCOM – European Software Control and Metrics Conference, London, UK.
- [12] Viktor Clerc (2005). “Towards architectural knowledge management practices for global software development” ACM, ISBN: 978-1-60558-038-8.
- [13] Berenbach.B, “Global Software Development Handbook”, [www.ism-journal.com/ITToday/AU9384\\_C002.pdf](http://www.ism-journal.com/ITToday/AU9384_C002.pdf).

[14] Loconsole, A. and Borstler, J. (2003). "Theoretical Validation and Case Study of Requirements Management Measures", Department of Computing Science, Umea University, Technical Report UMINF 03.02.

[15] Seija Komi-Sirvio and Maarit Tihinen (2005). "Lessons Learned by Participants of Distributed Software Development Knowledge and Process Management", Volume 12 Number 2 pp 108–122.

[16] Christopher Lindquist (2006). "Fixing the Requirements Mess", CIO magazine, URL: <http://www.cio.in/article/fixing-requirements-mess>.

[17] S.Arun Kumar, and T.Arun Kumar (2011). "State of software metrics to forecast the variety of elements in software development process", PDCTA 2011, Springer CCIS 203, pp. 561-569.