A Data Warehouse system for Human resource Management in a Distributed Software Development

Ningombam Subadani Devi, Laishram Prabhakar
Manipur Institute of Management Studies,
Manipur University,
Canchipur, Imphal 795003, Manipur, India
Email: nsubadani[@]gmail.com, prabhakarmims[@]gmail.com

Abstract: The study proposed a new concept called as 'Responsibility Index' (RIndex) that can administer all responsibilities of members involved in developing a team in a distributed team. This Index recorded and automatically updated roles changes. This research study uses the concept of a data warehouse system for data management. The objective of this study is to tracking and monitoring the performance of a developer using RIndex. A data warehousing system implemented to store the data require for the management of all the roles require of a project. The usages of this warehouse support the higher management team to manage the team without much technical knowledge because most of the analytical facilities are integrated with this system itself. Several interesting reports are a generated-allocated hour for each task, the status of the percentage of hours utilized so far. From such report, a development team can get some information about the progress of the project per task wise-Analysis, testing, etc. A report shows the group projection of status of the project such as total hours allotted for a task 'Analysis' or 'Design' etc., the percentage of hours allocation as per a task, used hours and its percentage of usages against each task and percentage of the entered projects hours. It represents almost all data appears in WBS and Gantt chart of a project. Based on the data available, the project team or manager can plan further actions.

Keywords: Data Warehouse system, responsibility index, Distributed Software Development, resource management

(Article History: Received 10 September 2016 and accepted 26 December 2016)

1. Introduction

The software can be developed for a variety of purposes as per requirement of the users by an individual or group of the software developer(s). Managing the activities involved in developing such software can be easy if the developer and their team members, the customer who provides requirements reside together in a centralized location. So the software development may be a slow process, expensive and may generate defects. It may give products with a large number of defects which cause serious problems in usability, reliability, and performance. The Chaos Report[1] reported by the Standish group, companies in the United States spent more than $250 billion per year on application development of approximately 175,000 projects. Around 53% exceeded their original budgets for losses of about $59 billion. Uncontrollable and non-repeatable processes cause many problems for software development organization.

It is said that "you can neither predict nor control what you cannot measure [2]." Reliable measurement is a key component in establishing a scientific basis for software engineering. Various measurement programs have been developed to improve software organizations. Helpful measurement programs help software organizations understand their capabilities. Furthermore, the continuous measurement can provide an efficient foundation for managing process improvement activities. The result is that software organizations have controllable and repeatable development processes and possess. Software measurement is the core of software process improvement and assessment programs, such as CMM [3], ISO 9001 [4], SPICE [5], etc.

Distributed Software Development (DSD) means the development of software in distributed environment. In a DSD, the participants are geographically separated most of the time. Normally the teams do not belong to the same area (country), language and culture, etc. One of the advantages is a greater availability of human resources in decentralized zones at (maybe) minimum cost and around the globe. It also makes the customer comfortable to get services anytime. However, some disadvantages are the distance that separates the development teams, problems in coordination and communication. Today's software projects are geographically distributed with the limited face-to-face interaction between participants [6]. The

project manager must use appropriate tools to control, support and manage the entire team.

There are several success factors. Some of the importantly extracted success factors of such distributed software development are.

a. Training of human resources with the tools and processes and adoption of agile methodologies based on incremental integration and frequent deliveries [7].
b. Detail study and intervention of human resources by participating in surveys [8].
c. Acknowledging the activities with information on pending issues, errors, and resources, etc. [9].
d. organization of an efficient communication instrument between the members to show the current status of any assigned task [10].

A variety of frameworks has evolved over the years. New processes, method, and tools are highly necessary to manage such development, team. This research study, therefore, proposed a management strategy to manage the activities involved in development distributed software using the concept of a data warehouse system. This will help to track and monitor the performance of a developer using concept call responsibility index (RIndex). Most important is a timely collection of feedbacks from the customers. A data warehousing system implemented to store the data require for the management of all the roles require in this project. The usages of this warehouse support the higher management team to manage the team without much technical knowledge because most of the analytical facilities are integrated with this system itself.

2. Review of the literature

Distributed Software Development (DSD) has recently become a serious topic in research. The goal of DSD is to make the skilled workforce more readily available [11]. The development of software in a distributing environment is going to continue to increase the time-to-market by around-the-clock. It is to enhance flexibility in merger and acquisition of different business opportunities. Such geographical distribution of task becomes increase with the high transfer of development and maintenance activities from the developed countries to developing one or vice versa [12]. There are several disadvantages of DSD which are caused by the distance that separates the development teams for communication and coordination among them [13][14][15]. Previous studies [16,17] suggest that cross-site communication and coordination issues cause a substantial loss of development speed. It is reported that in a distributed work the items (a design or a code etc.) appear to take a longer time to complete as similar items where all the work is collocated [18].

In the study report of Chaos of Standish group [19], the company in the United States spent more than $250 billion each year on IT application development of roughly 175, 000 projects. Around 31% were canceled before completion giving losses of about $81 billion. Approximately 53% exceeded their original budgets. Those projects that managed to completion delivered an average of 42% of the planned features. Another report [20] from the Software Engineering Institute (SEI) indicated that out of approximately 542 software organizations participating in the CMM maturity evaluation, the conversion from inputs to final software products is uncontrolled. But, there is still no visibility as to how software products are produced, and any trouble to the development team or resources can easily cause project failure. It is said that 87% of the software organizations in the survey were unable to control their development processes. The development process is so reactive that management control is impractical. Rick management studies are also performed based on deliverable of projects. Such a study can report the risk associated with a deliverable to alerts the developer team to minimize the risk using preventive measures [21].

In a distributed development environment, the coordination and communication works are complex. Managing of such projects becomes more challenging and troublesome. Distributed software development has many complexity and challenge. Making distributed project teams' work efficiently and delivering better outcomes on time and within budget is a severe test of industries. To answer these challenges, experts and researchers or practitioners are continuously finding and developing vast amounts of frameworks, guidelines, tools, methodologies, and tips [22, 23]. Performance analysis of any organization can be prepared on the factors such as efficiency, innovativeness, etc.[24].

a. Trends in Distributed Software Development

The majority of the primary studies analyzed on specific domain related case studies. Non-experimental studies and surveys in which members involved in the development take part in outlining their difficulties have a significant representation. A good number of the studies are centered on the organizational processes [25]. Members of a virtual team tend to be less productive due to feelings of isolation and indifference. Literature deals with the poor

socialization and socio-cultural differences which cause a lack of trust. Numerous changes in processes, lack of continuity in communications, and lack of collaborative tool integration cause remote groups to be unaware of what is important because they do not know what other people are working on.

Storey et al. [26] recommend a framework for the comparison and understanding of visualization tools that provides awareness of software development activities, giving a solid grounding to the existing theoretical foundation of the field. A visualization tool supports DSD processes by creating visual representations of both software artifacts and software development actions, thus allowing developers to explore the relationships between them. Herbsleb et al. [27] developed a tool that provides a visualization of the changing management process. Thus, it makes easy to discover who has experience in working on which parts of the code or design. In the identical line, R. Holmes and R. J. Walker [28] present the YooHoo alertness system to help developers to keep apprised of code changes, providing notifications in a flexible manner. The distribution environment also causes the problem to share information in Knowledge Management system. M. A. Babar [29] proposes the application of an electronic workspace paradigm to capture and share knowledge to support the software architecture processes. H. Zhuge [30] presents an approach that works with a knowledge repository in which information related to each project is saved by using an internet-based communication software, thus enabling a new team member to become rapidly skilled by learning the knowledge stored.

b. Team Management in Distributed Environments

The Distributed Software Project Management Tool (DSPM tool) is an integration of tools opening the realms of software project management to users distributed across the world [31]. The first prototype presented the core of a software repository and configuration management. The DSPM tool presents the second prototype and introduces new concepts of task and team management to improve the quality of software projects [32]. The resources such as repository and files are exposed to the entire team. The eGroups [33], iTeamWork [34] and MS Project [35] are online services. The risk of using an online project management is that all resources and files are exposed to a member of the project. MS Project provides an excellent environment to plan and monitor the progress of tasks. MS Project is a standalone application, in which recipients must purchase and install their own copy of MS Project.

Global software development (GSD) model are used for development of projects in distributed environment. GSD involves the three major types of detachment: geographical, temporal, and socio-cultural [36]. though, GSD is fraught with difficulties arising from geographical, temporal and socio-cultural distances. Vast geographical distances imply the difficulty of relocating to another of the company's sites. The temporal detachment across multiple time zones reduces the number of overlapping working hours. Socio-cultural distance arises from the different national and organizational backgrounds of the people involved [37]. The communication and control problems are recognized as being the most troublesome and pervasive in software development. The main challenge for GSD teams is the lack of casual communication which has been found to be an essential method in traditionally co-located development [38] [39].

However, in most of such study, the proper framework for the organization of the responsibilities involved in distributed software projects are not passed out using the concept of Data warehousing. Some tools are required for the Project manager that help to monitor and managed the responsibilities and roles of the team members dynamically.

3. Brief theory of Layer architecture and the responsibility index

Many members are involved in the development of the Distributed Software Projects. Each member needs to know the responsibilities of each other for appropriate management of the activities to produce coherent outcomes. Several software management tools do not fully help the managers to manage the member's responsibilities, but the layered architecture can assist to do so. This style gives the layer level description of the activity involved and defines and directs the group of the workforce. The groups of the workforce, the development team, and the customer can know the activity and assigned member involved in working on that specific task. The layered architecture is useful to the development team and stakeholders. This study extended the new approach of layer pattern name as 'Responsibility Index.'

a. What is a Responsibility Index?

Responsibility Index (RIndex) is the value given to a possible role or responsibility to a person for developing a project. Let's assume that a project has several tasks such as Analysis, Design, Coding, Data Entry, temporary free, etc. all these tasks can be considered as the possible role

that can be given to a person. Each of such tasks can be given an Index value, for example, 1: Analysis, 2: Design, 3: Coding, etc. This Index will help to manage the possible roles of a development team. A person can have several roles in an organization. The status of the RIndex will be automatically updated wherever there come any changes in roles or component of work.

The 'Responsibility Index' helps in managing all the responsibilities of a large distributed project team. This index will help in Tracking and monitor the performance of a developer. This Index will show the time taken by a person on specific task. In needed the assistance can be provided to an individual who is looking for help, in case they are unable to finish the work in given time. It also will contribute to judge the performance of a person in time of performance appraisal and choosing new members for new projects. The RIndex will be automatically updated when a role changes and a team member will know the role of each other. It will assist the development team, stakeholder and customer to come closer bringing all involving members closer reducing the communication gap. By using the facilities of Data Mining, a Project Manager can plan, monitor the progress of the projects, and also help a needy member in time of urgency [40].

b. The benefit of 'Responsibility Index.'

1. Responsibility Index is linked to all possible roles of a person for developing a project. It helps in managing all the responsibilities of a large distributed project team.
2. The status of the RIndex is automatically updated on changes in roles helping in continuous Tracking and monitoring the performance of a developer anytime.
3. The Index shows the time taken by a person on a task.
4. In needed the assistance can be provided to an individual who is looking for help, in case they are unable to finish the task in given time.
5. It also will help to judge the performance of a person in time of performance appraisal and also choosing new members for new projects.
6. It helps the team, stockholder, and customer to come closer by knowing each other's roles. Customer will are aware of their designated developer, in the case of rising ticket (complaints). It helps to bring all involving members closer reducing the communication gap.

c. The overview of the Layer architecture

Architectural Styles are the high-level set of policy that constrains the architecture for certain context and describe to describe solutions to the problems. Architecture is considered to consist of several components and the connectors (interactions) between them. A software component can be a software module, a service, or a resource that encapsulates a set of related functions. Design explicitly addresses functional requirements while architecture explicitly addresses functional and non-functional requirements. The goal of software architecture is to enable the construction of enormous system architectures [41]. Layer pattern is better suited to systems that require a dependable operation because it is easier to implement error management. Structurally, each layer provides a related set of services. Dynamically, each layer may only use the layers below it. Layer A may use layer B because it depends on something B does for example data are written to the database by B to be used by A. A evokes B says A passes manage or data or both directly to B.

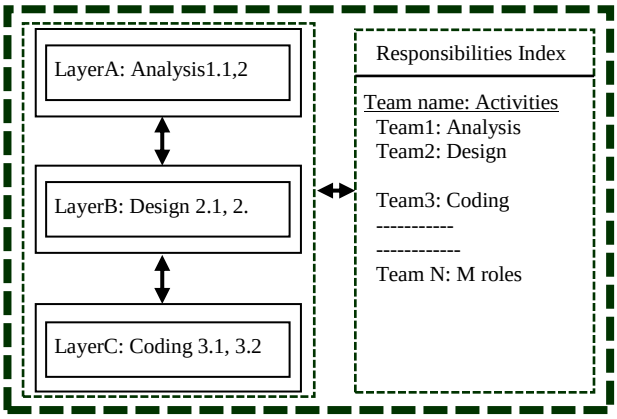

Figure 1. Enhanced version of the General Layer Architecture

A sample sketch of the enhanced version of the Layer Architecture is given in Figure 1. A development team may have several team groups such as team1, 2, 3, etc. Suppose that project has numerous phases such as Analysis, Designing, and Coding, etc. Different teams were assigned to different works. Let's assume that TeamA is responsible for designing a component of the project. TeamA has three members who work in various roles. A sample of the situation is shown in Table1. Initially (T1), the member 'Mr.X' has works of designing 'Design Table T1' which Rindex2 as per this scenario. Likewise member 'Mr.Y' has worked off 'Testing Table T1', 'Mr.Z' has worked off 'Entering data to Table T1' respectively.

After 8 hours (T2) (say a day), the role of 'Mr.X' is changed to 'Refresh of Table T1' which was finish product from another member. The Index value of 'Mr.X' is changed to 5 from the previous Rvalue2. Likewise, the role of 'Mr.Y' change to 7 meaning 'free,' as no work was assigned, 'Mr.Z' continues with the 'design of form1'.

The 'Responsibility index' was automatically updated wherever there happen any changes in roles or component of work. From this current RIndex value, the evolving team members will come to know about the component handled or handling by the team members. If the type of the project is customer service, the index value can be shared with the customer and customer can cooperate with the corresponding responsible person in times of any issues. In the current customer service system, the ticket raised by the customers are received by a central team and distributed to the responsible member. Occasionally the ticket goes around, and services are delayed, annoying the customers. Management of activity will be much earlier with the help of database which contains the roles. By adapting this Index, any changes in the tasks will be automatically updated. This option does not be available in traditional layered architecture.

Table1. Sample of role change in new layer architecture

	T1: Assigning time (Old)	T2: After 8hours (New)
Member	RIndex and Activity	
Mr.X	2.Design T1	5.Refresh T1
Mr.Y	3.Testing T1	7.Free (no work)
Mr.Z	4.Data Entry T1	6.Codding for T1

d. A case study of 'Computerized Medical Clinic.'

We have considered a scenario to develop a project name as 'Computerized Medical Clinic.' This project adopts the development process of DSD process. Some of the core modules are

- Patient Service,
- Security Service
- User's Application
- Application Services
- Image Databases

The Layer architecture shows only the components of the possible software. But it does not demonstrate that which components are going to developed together or what sequences of the components are. The layered architecture can be applied to resolve key distributed system design challenges. It will help to resolve the following common design challenges like separating concerns between tiers, improving performance, enabling client extensibility, ensuring platform-neutral, network-transparent communication, decoupling suppliers and consumers, locating and creating components scalable, minimizing resource utilization, etc.

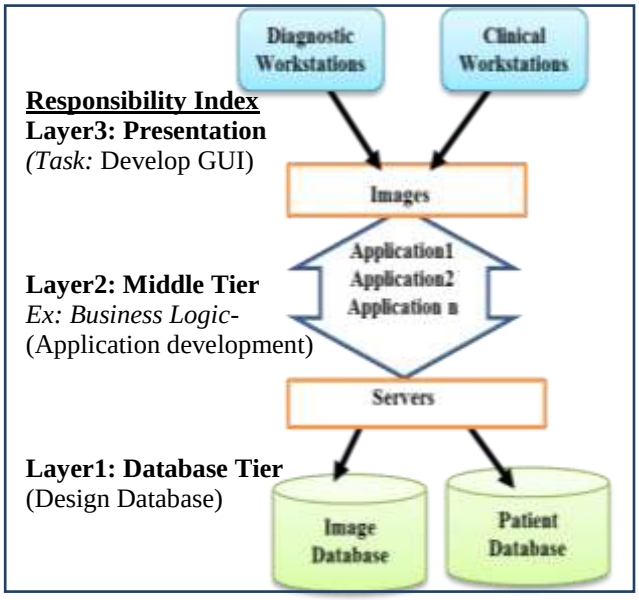

Figure2. Medical Imaging Systems-(with Layer architecture and Responsibility Index (RIndex))

e. The project details

Consider that the following Table2 gives some detail of work breakdown structure (WBS) of the above case study. Table3 gives the sample of projects allotted to an organization which has a several projects, team. Table4 gives the sample of employees and their details of a project or an organization. Table5 gives sample predefined values of Responsibility Index (Rindex), also known as Task name. This task name or Rindex has subtask number and names also for easy management. Table6 gives the details of Task and subtask of a project. Table7 gives the details of Task assignment to a member of the team. Table8 gives the Task progress of a team of a project.

Table2: Sample of task assign to team and member name (WBS)

Task & hours	Sub task No	Sub-task name	Hours	Assign to	Team name
Analysis (35 hours)	1	Analyze the requirements	10	Mr.X	TeamA
	2	Users problem & solution direction	10	Mr.Y	TeamA
	3	Find the risk factors	5	Mr.Z	TeamA
	4	Draft project analysis report	10	Mr.X	TeamA
Design (30)	5	Design table1:employee	3	Mr.X	TeamA
	6	Design table2:patient	3	Mr.Y	TeamA
	7	Design tableT3	3	Mr.Z	TeamA
	8	Design TableT4	3	...	TeamA
	9	GUI1: Image acquisition system	4	...	...
	10	GUI2: Preprocessing the images	4	...	...
	11	GUI3: Detection system of diseases	5	...	...
	12	GUI4: Reporting system of diseases	5	...	...
Coding (55)	13	Code GUI1: Acquisition unit	15	...	...
	14	Code GUI2: Preprocessing unit	15	...	...
	15	Code GUI3: Detection Unit	15	...	....
	16	Code GUI4: Reporting unit	10	....	....
Testing (60)	17	Testing Table T1	2	Mr.X	
	18	Testing Table T2	2	Mr.Y	
	19	Testing Table T3	2	Mr.Z	
	20	Testing Table T4	2		
	21	Writing test scripts-p1	4		
	22	Writing test scripts-p2	4		
	23	Writing test scripts-p3	4		
	24	Writing test scripts-p4	4		
	25	Testing the program1	2		
	26	Testing the program2	2		
	27	Testing the program3	2		
	28	Testing the program4	2		
	29	Testing the interfaces GUI1	4		
	30	Testing the interfaces GUI2	4		
	31	Testing the interfaces GUI3	4		
	32	Testing the interfaces GUI4	6		
	33	Reporting the testing outcomes	10		
Installa-tion(20)	34	Installation of the software	10		
	35	Preparation of 'Help document.'	10		

Table 3: Sample of projects allotted to an organization

Pno	client	Cost($)	Others information
1	ABC Clinic	1,00,000	A medical project
2	ABC College	20,000	College Website
3	Technodata	2,00,000	Progress Management of Civil works

Table 4: Employees Table of a project

No.Ename	Address	Qualification	Skillset
1Mr. X	Guwahati	BTech	Mysql, Php
2Mr. Y	Imphal	BTech	Mysql, Php
3Mr. Z	Shillong	MCA	Mysql, Php
4Mr. Prof	Guwahati	Ph.D	Mysql, Php

Table 5 Responsibility Index of task

RIndex value	Task name
1	Analysis
2	Design
3	Coding
4	Data entry
5	Refresh
6	Testing
7	Installation
8	Others
9	Free

Table 6. Table of Task and subtask of project

Sub task no	Task Name	Sub Task Name	Hrs	Start date	End date
1	Analysis	Analyze the requirements	10	6/6/2015	6/16/2015
2	Analysis	Users problem & solution direction	10	6/7/2015	6/17/2015
3	Analysis	Find the risk factors	5	6/7/2015	6/17/2015
4	Analysis	Draft project analysis report	10	6/7/2015	6/17/2015
5	Design	Design table1:employee	3	6/7/2015	6/12/2015
6	Design	Design table2:patient	3	7/1/2015	7/6/2015
7	Design	Design tableT3	3	7/2/2015	7/7/2015
8	Design	Design Table T4	3	7/3/2015	7/8/2015
9	Design	GUI1: Image acquisition system	4	7/5/2015	7/10/2015
10	Design	GUI2: Preprocessing the images	4	7/6/2015	7/11/2015
11	Design	GUI3: Detection system of diseases	5	7/7/2015	7/12/2015
12	Design	GUI4: Reporting system of diseases	5	7/8/2015	7/13/2015
13	Coding	Code GUI1: Acquisition unit	15	7/9/2015	7/19/2015
14	Coding	Code GUI2: Preprocessing unit	15	7/10/2015	7/20/2015
15	Coding	Code GUI3: Detection Unit	15	7/11/2015	7/21/2015
16	Coding	Code GUI4: Reporting unit	10	7/12/2015	7/22/2015
17	Testing	Testing Table T1	2	7/4/2015	7/9/2015
18	Testing	Testing Table T2	2	7/4/2015	7/9/2015
19	Testing	Testing Table T3	2	7/4/2015	7/9/2015
20	Testing	Testing Table T4	2	7/4/2015	7/9/2015
21	Testing	Writing test scripts-p1	4	8/1/2015	8/6/2015
22	Testing	Writing test scripts-p2	4	8/2/2015	8/7/2015
23	Testing	Writing test scripts-p3	4	8/3/2015	8/8/2015
24	Testing	Writing test scripts-p4	4	8/4/2015	8/9/2015

25	Testing	Testing the program1	2	8/6/2015	8/11/2015
26	Testing	Testing the program2	2	8/7/2015	8/12/2015
27	Testing	Testing the program3	2	8/8/2015	8/13/2015
28	Testing	Testing the program4	2	8/9/2015	8/14/2015
29	Testing	Testing the interfaces GUI1	4	8/11/2015	8/16/2015
30	Testing	Testing the interfaces GUI2	4	8/12/2015	8/17/2015
31	Testing	Testing the interfaces GUI3	4	8/13/2015	8/18/2015
32	Testing	Testing the interfaces GUI4	6	8/14/2015	8/19/2015
33	Testing	Reporting the testing outcomes	10	8/15/2015	8/25/2015
34	Installation	Installation of the software	10	9/1/2015	9/11/2015
35	Installation	Preparation of 'Help document.'	10	9/2/2015	9/12/2015

Table 7 Data contents of Table of Task assignment

id	Subtaskname	STask no	E no	Emp Name	task	Hrs
1	Analyze the requirements	1	1	Mr.X	Analysis	10
2	Users problem & solution direction	2	2	Mr.Y	Analysis	10
3	Find the risk factors	3	3	Mr.Z	Analysis	5
4	Draft project analysis report	4	1	Mr.X	Analysis	10
5	Design table1:employee	5	1	Mr.X	Design	3
6	Design table2:patient	6	2	Mr.Y	Design	3
7	Design tableT3	7	3	Mr.Z	Design	3
8	Testing Table T1	17	1	Mr.X	Testing	2
9	Testing Table T2	18	2	Mr.Y	Testing	2
10	Testing Table T3	19	3	Mr.Z	Testing	2

Table 8 Data contents of Table of Task progress

Em Id	ST No	R Ind	Update status	Time taken	Earn time	Emp Name	Subtaskname
1	1	1	Finish	9	1	Mr. X	Analyze the reqd
2	2	1	Finish	10	0	Mr. Y	Analysis-Users prob
3	3	1	Finish	5	0	Mr.Z	Analysis-Find the risk factors
1	4	1	Not finish	5	0	Mr.X	Draft project report
1	5	2	Finish	2	1	Mr. X	Design table1
2	6	2	Finish	3	0	Mr.Y	Design table2
3	7	2	Finish	3	0	Mr.Z	Design tableT3
1	17	6	Finish	1	1	Mr. X	Testing Table T1
2	18	6	Not finish	1	0	Mr.Y	Testing Table T2
3	19	6	Finish	2	0	Mr.Z	Testing Table T3

Table 9: Sample Responsibility Index (with time)

Name: TeamA,		Component: Layer1(Design database Tier)			
Name (team)	Task	Subtask	Current Index value		
			Start T	End T	RIValue
Mr. X(A)	--	--	8.00	00	9
Y(A)	--	--	8.00	00	9
Z(A)	--	--	8.00	00	9
X(A)	Design	Table1: Employee	09.00	11.00	2
Y(A)	Design	Table2	09.00	12.00	2
Z(A)	Design	Table3	09.00	12.00	2
Y(A)	--	--	12.00		9
X(A)	Testing	Table1	11.00	12.00	6
X(A)			12.00		9
Z(A)	--	--	12.00		9

Table 9 gives a sample of data on Responsibility Index (with time). In particular a day, initially Mr.X,Y and Z of team A has no work, and its index value is 9. Let consider some example. Mr.X was assigned to 'Design Table1:Employee' within given 3hours. An Index value of Design; that is 2 is shown again RIndex of Mr.X (A) with respect to time. Mr.X executed the first task 'Design table1' and completed with 2hrs (9 am to 11 am) by earning 1hour. Mr.X executed the second task 'Test table1' at 11 am and got free (RIndex is 9) after 12 am. Anyone access this RIndex values will come to know about the roles and works executed by Mr.X(A). Consider that Mr.Y has given work of 'Design Table2' within given 3hours at 9 am and finish the task in given time(3hours). S/he does not earn any extra point, which links to incentives or future assessment. The other task is also executed in different time, and RIndex values are updated as and when the development member updated their progress. Doing so we can calculate the 'percentage %' of hours allotted and 'used hours' against each phase of a project. The following figures are provided for the purpose. The study used some concept of Data Mining process and prediction for analyzing the different scenario of the projects.

4. The Development of Data Warehouse system

The data warehouse of the study is implemented using different phases. The following steps are followed to implement the data warehouse system.

1. Loading
2. Extraction and transformation
3. Mining or prediction

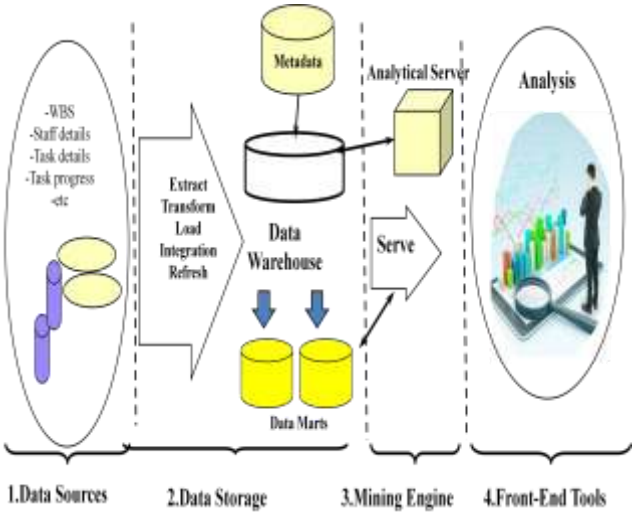

Figure3. Sample of Data warehouse system

i). Loading

The source databases are integrated into a single Data Warehouse and formed a **Data Management System**. The data from the following table are collected from different data sources. Some of the required table for the chosen case study are given below

Table2: Task assign to team and member name (WBS)
Table3: Sample of projects allotted to an organization
Table4: Sample Employees detail of projects
Table5: Responsibility Index of task
Table6: Data of Table of Task and Subtask of project
Table7: Data of Table of Task assignment

The concept of Multi-dimensional schemas is used for this study in which we adopt the Star schema. Star schema consists of a fact table with a single table for every dimension. The sample of the star schema is given below.

a. The Dimension table.

These tables are used store the details of most details data such as details of staff, WBS, task, subtask, index values and most important data the progress status of the task which are updated by the evolving member of a development project.

b. Fact table

This Fact table contains some measured or observed variable (s) such as total time taken to a task and identifies its pointers to dimension tables. The fact table contains the data, and the dimensions to identify each tuple in the data.

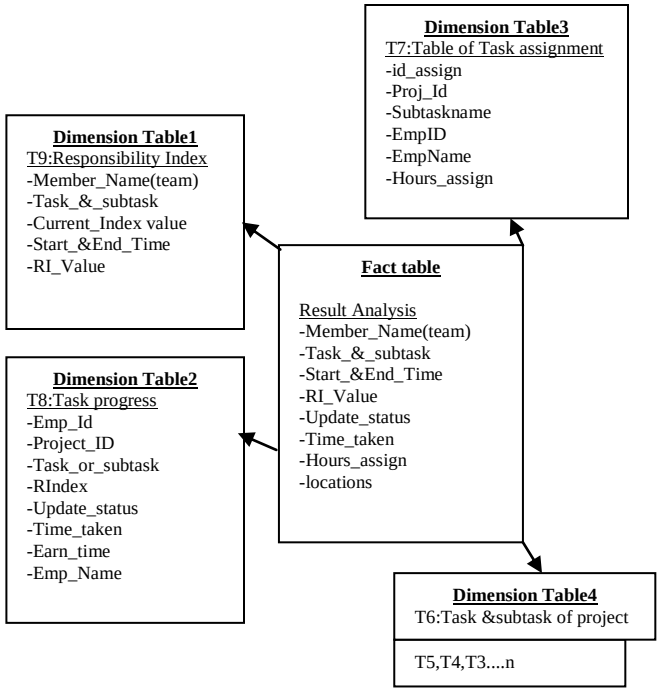

Figure4. Sample of Star schema (relation of fact table and dimension tables)

ii). Extraction and transformation

The loaded data are extracted, and necessary preprocessing process such as cleaning, transformation process are performed. During the integration process, the data from different sources are integrated into the single data set. For example the list of employees involved in project1, project2, etc. are integrated into single data set of 'employees of projects.' In our case study the **Table8 contents the data of progress of Tasks** assigned to all members in different projects. The progress of the task is integrated to single task data set. If needed the Data mart are implemented for storing the the subset of the data such list of employees of same projects or, similar skillsets, etc.

iii). Mining or prediction operations

The Mining Engine or the analytical server help to provide some basic data warehousing functionalities that will provide to the senior manager to manage the data of the projects.

a. Roll-up operation:

The details are summarized for prediction based on some business requirement. It helps to zoom out stepwise.

b. Drill-Down operation:

In this, the increasing levels of detail are revealed. Drill-down is performed to view the data at a level of increased detail. It is used to zoom into more detailed data by Changing Dimensions.

c. Pivoting operation:

Cross tabulation is performed. Data pivoting is a process of rotating the view of data. A pivot table helps the problem of displaying the high-level volume of data by means of automated calculations which are defined in a separate column side by side with data column of the requested data observation. Pivoting summarizes the data contained in a long list into a compact format allowing automatic sort, count, and total, etc.

d. Slicing and dicing operation:

Performing projection operations on the dimensions. The slice and dice are used to break a detail of information down into smaller parts or to study it from different viewpoints so that it can understand better. In a slice, detail information is sliced into small into small cubes. In dicing the sliced information is again decomposed into smaller parts or views that will yield more information.

5. Reporting from the Data warehouse system

The following table10 gives the hours allotted and used against each phase of a project. The similar data can be visualized as provided in figure5 and 6. This type of report will show different scenarios of the projects such as 'Allotted hours,' 'Percentage as per allotted hours,' 'Progress (used hours)' and Percentage of used hours, etc. It used some concept of Data Mining and prediction for analyzing the status of different scenarios of the projects.

Table 10: Percentage (%) of hours allotted and used in each phase

Task	Allotted Hours		Used hours		
	Allotted Hours	%	Used hours	% Individual	% Entire project
Analysis	35	17.5	35	100	17.5
Design	30	15	9	30	4.5
Coding	55	27.5	0	0	0
Testing	60	30	6	10	3
Installation	20	10	0	0	0
Total	200	100	50		25

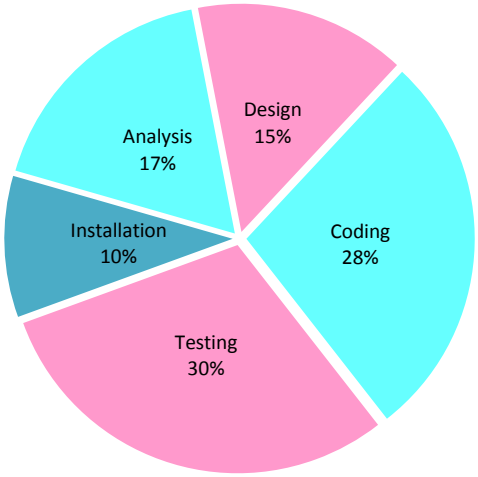

Figure5. Percentage (%) of hours allotted to each project phase

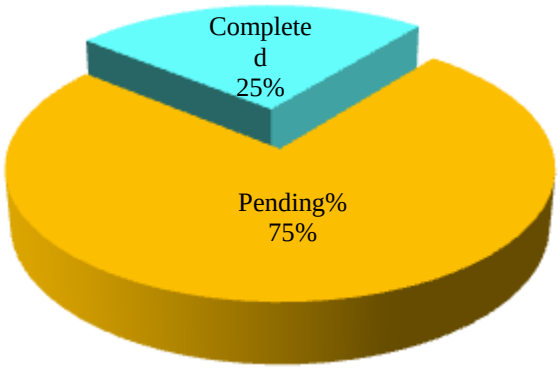

Figure 6. Shows the status of percentage (%)-Completed Vs pending

This figure shows the status of percentage (%) of hours utilized so far. The development team can get some information about the progress of the project per Task wise-Analysis, testing, etc. A report shows the group projection of status of the project such as total hours allotted for a task 'Analysis' or 'Design' etc., the percentage of hours allocation as per a task, used hours and its percentage of usages against each task and percentage of the entered projects hours. It represents almost all data appears in WBS and Gantt chart of a project. Based on the data available, the project team or manager can plan further actions.

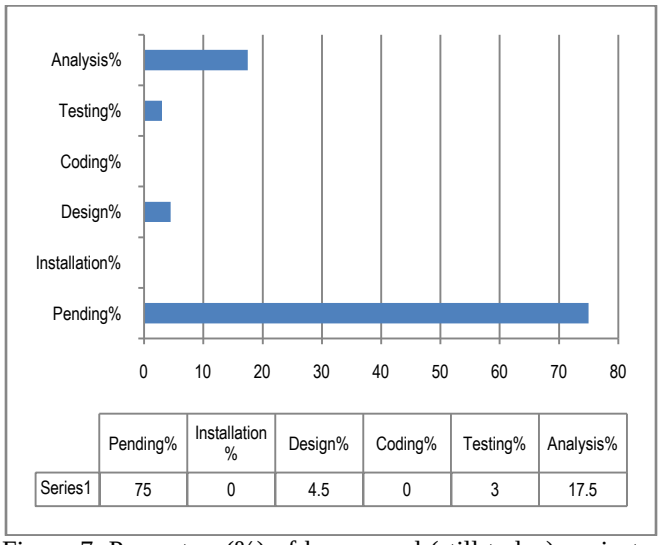

Figure 7. Percentage(%) of hours used (still today) against each phase of a project

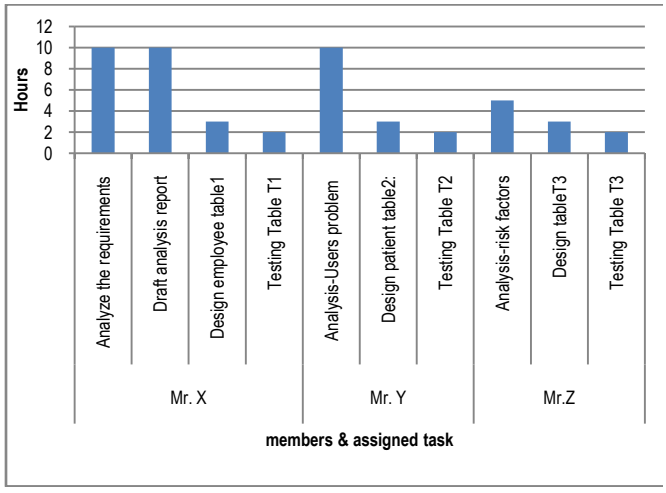

Figure 8: Team member and allotted task of a project

This figure will help to show the list of task and hours assigned to the particular member.

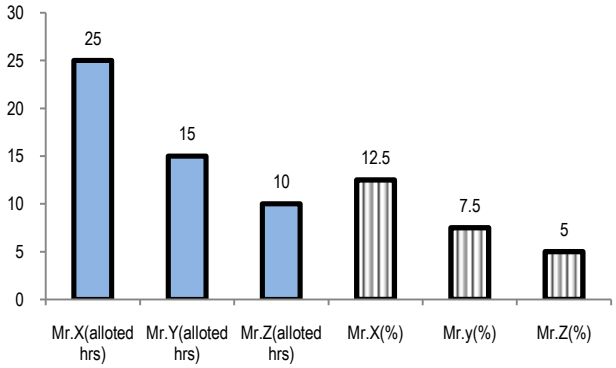

Figure 9. Total hours allotted and Percentage (%) of each Team member

This Employee Utilization report will show the hours allocation per a team member of a project. The data are provided to compare the allotted hours among some group of member of a project. This figure help to project the most and least assigned member in a team.

7. Conclusion

The study proposed a new concept called as 'Responsibility Index' to manage all the responsibilities of a large distributed project team. The new Index has the capacity to manage all the roles and responsibilities of members involved in developing a team. In such distributed team, the projects are conducted jointly from various parts of the globe with several roles changes. This Index recorded and automatically updated roles changes. It helps all team members to know the role of each other. Thus, it helps the development team, stockholder, and customer to come closer. Customer will know their designated developer, in the case of complains. This helps to manage development activities, and member's responsibilities are improving the relation with customer and stakeholder. It also helps timely collection of feedbacks from the customers helping involving members closer reducing the communication gap. A sample 'Application' of this Responsibility Index (RIndex) has been developed and demonstrated.

The Mining Engine or the analytical server help to provide some basic data warehousing functionalities that will provide to the senior manager to manage the projects. A report shows different scenarios of the projects such as 'Allotted hours', 'Percentage as per allotted hours', 'Progress(used hours)' and Percentage of used hours, etc. A report shows the group projection of status of the project such as total hours allotted for a task, the percentage of hours allocation as per a task, used hours. It represents the data appears in WBS and Gantt chart of a project. Based such data project manager can plan further actions. Such study of implementing a Data Warehouse system for Human resource management in a Distributed Software Development may contribute a lot in managing resources in distributed projects.

References

[1] Standish. The Chaos Report. http://www.standishgroup.com/sample research/PDF pages/-chaos1994.pdf

[2] T. DeMarco. Controlling Software Projects. Yourdon Press, New York, 1982.

[3] W. S. Humphrey. Managing the Software Engineering. Addison-Wesley, 1989.

[4] ISO. ISO 9001 quality systems - model for quality assurance in design/development, production, installation, and servicing. ISO, 1987

[5] ISO. ISO: 1998 information technology - software process assessment. ISO,1998.

[6] M Jiménez, M Piattini, and A Vizcaíno, "Challenges and Improvements in Distributed Software Development: A Systematic Review," Advances in Software Engineering, vol. 2009, Article ID 710971, 2009. doi:10.1155/2009/710971

[7] K. Crowston, Q. Li, K. Wei, U. Y. Eseryel, and J. Howison, "Self- organization of teams for free/libre open source software development," Information and Software Technology, vol. 49, no. 6, pp. 564–575, 2007.

[8] F. J. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," Software Quality Journal, vol.16(2), pp. 237– 261, 2008.

[9] J. Greenfield, K. Short, S. Cook, S. Kent, and J. Crupi, Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools, John Wiley & Sons, New York, NY, USA, 2004.

[10] K. Narayanaswamy and N. M. Goldman, "A flexible framework for cooperative distributed software development," The Journal of Systems and Software, vol. 16, no. 2, pp. 97–105, 1991.

[11] W. Aspray, F. Mayadas, and M. Y. Vardi, "Globalization and offshoring of software," Report of the ACM Job Migration Task Force, Association for Computing Machinery, New York, NY, USA, 2006

[12] B. Meyer, "The unspoken revolution in software engineering," IEEE Computer,2006, 39(1): 121-123

[13] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University & Durham University Joint Report, Staffordshire, UK, 2007

[14] F.P. Brooks Jr.,The Mythical Man-Month: Essays on Software Engineering.Addison Wesley, 1995.

[15] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems,"Comm. ACM,vol. 31, (11), pp. 1268-1287, 1988

[16]J.D. Herbsleb and R.E. Grinter, "Splitting the Organization and Integrating the Code: Conway's Law Revisited,"Proc. Int'l Conf.Software Eng.,pp. 85-95, 1999.

[17] J.D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter, "An Empirical Study of Global Software Development: Distance and Speed,"Proc. Int'l Conf. Software Eng., pp. 81-90, 2001

[18] JD Herbsleb, A Mockus, "An empirical study of speed and communication in globally distributed software development." Software Engineering, IEEE Transactions on 29.6 (2003): 481-494.

[19] Standish. The Chaos Report. www.standishgroup.com /sample research/PDFpages/-chaos1994.pdf,1994.

[20] W. C.Peterson "SEI's software process program-presentation to the board of visitors", Software Engineering Institute, Carnegie Mellon University.1997

[21] Y. Subadani Devi and L. Prabhakar. "A Preventive Risk Analysis for Managing Distributed Software Projects based on Deliverable." International Journal of Computer Applications 68.1 (2013): 27-31.

[22] J.Espinosa, DeLone, W., and Lee, G. "Global boundaries, task processes and IS project success: a field study," Information Technology and People, 2006

[23] Y. Subadani Devi, Y. Jayanta Singh, Laishram Prabhakar. "Grooming a New Team with Potential Roles using the Scrum Practices."Proc. Int. Multi Conf. of Engineers & Computer Scientists. Hong Kong, 2012.

[24] ANH Zaied. "An integrated knowledge management capabilities framework for assessing organizational performance."Int. Journal of Information Technology and Computer Science(IJITCS), 2012.

[25] M Jiménez, M Piattini, A Vizcaíno "Challenges and improvements in distributed software development: A systematic review." Advances in Software Engineering,3, 2009

[26] M.A. D. Storey, D. Čubranić, and D. M. German, "On the use of visualization to support awareness of human activities in software development: a survey and a framework," in Proceedings of the ACM Symposium on

Software Visualization (SoftVis '05), pp. 193–202, St. Louis, Mo, USA, May 2005
[27]J. D. Herbsleb and D. Moitra, "Global software development," IEEE Software, vol. 18, no. 2, pp. 16–20, 2001
[28] R. Holmes and R. J. Walker, "Promoting developer-specific awareness," in Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '08), Leipzig, Germany, May 2008
[29]M. Ali-Babar, "The application of knowledge-sharing workspace paradigm for software architecture processes," in Proceedings of the 3rd International Workshop on Sharing and Reusing Architectural Knowledge (SHARK '08), Leipzig, Germany, May 2008
[30] H. Zhuge, "Knowledge flow management for distributed team software development," Knowledge-Based Systems, vol. 15, no. 8, pp. 465–471, 2002
[31] R. Siirjaputra and P. Maheshwari, A distributed software project management tool, Proc. IEEE 8th int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, USA, 1999
[32] HE Lam, P Maheshwari, "Task and team management in the distributed software project management tool." Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International. IEEE, 2001
[33]eGroups, www.eGroups.com/
[34] iTeamWork, www.iteamwork.com/
[35] Microsoft Project www.microsoft.com/office/project
[36] P.J.Agerfalk, B. Fitzgerald, H.H. Holmström, B.Lings, B. Lundell, E.O,Conchuir, "A framework for considering opportunities and threats in distributed software development". International Workshop on Distributed Software Development, France, Austrian Computer Society, 2005
[37]PJ Ågerfalk, B Fitzgerald, HH Olsson, Eoin Ó Conchúir, "Benefits of global software development: the known and unknown." Making Globally Distributed Software Development a Success Story. Springer Berlin Heidelberg, 2008
[38]B.Curtis, H. Krasner, N.Iscoe, "A field study of the software design process forlarge systems. Communications of the ACM 31(11), 1268–1287 (1988)
[39]E.Carmel, " Global Software Teams: Collaborating Across Borders and Time Zones", 1st edn. Prentice-Hall, Upper Saddle River (1999
[40] Yumnam Subadani Devi, and Laishram Prabhakar. "Management of Possible Roles for Distributed Software Projects Using Layer Architecture." International Journal of Information Technology and Computer Science (IJITCS) 7.7 (2015): 57.
[41]M. Shaw, P. Clements. "A Field Boxology: Preliminary Classification of Architectural Styles for Software Systems", 1996.

Author's Profile

Dr. Ningombam Subadani Devi holds a Ph.D from Manipur Institute of Management Studies (MIMS), Manipur University,India. She also holds MBA specialization in Information Technology from Sikim Manipal University. Her interest areas are Software Engineering, Management Information Systems and Database Management System.

Laishram Prabhakar, is an Associate Professor in Manipur Institute of Management Studies (MIMS), Manipur University, India. He holds a Ph.D degree(Information System Management. Presently working as Coordinator of Community College, Manipur University. He has over two decades of experience in the field of Information System Management. His area of interest is in the area of developing strategies for innovation and knowledge management. He provides concepts of knowledge management in Business and Technology.