

A Comparison Based Framework for Reuse Adoption in an Organization

¹Anshul Kalia

Ph.D. Scholar, Dept. of Research Innovation & Consultancy,
IKG-Punjab Technical University, Kapurthala-144601 (India)
ansh.kamal007@gmail.com

²Sumesh Sood

HOD, Dept. of Computer Applications,
IKG-Punjab Technical University Campus Dinanagar, Dinanagar (India)
sumesh64@gmail.com

Abstract - Organizational issues are the first step towards the implementation of reuse program. Resolving these issues is a complex task due to the entanglement of various factors such as human, technological, environmental etc. In a present course of study number of previous methods have been identified and studied. It is observed that the reuse of software components is practiced on ad-hoc basis or at individual level for personal benefits. The study proposes a new organizational framework which is dedicated to reuse of software components. It stresses that the reuse should be followed with a systematic approach. To fulfill this objective the framework serves as a central area that addresses all the issues concerning reuse but the execution and decision making has been decentralized within the framework. The proposed framework consists of a hierarchical structure, roles & responsibilities of personnel, and the operations to be performed. The framework has been measured using a comparison based approach with the RiSE – Reference Model. During the course of comparison, the maturity level and the capability of the processes of the proposed framework has been determined.

Keywords—Organizational issues, software reuse division, influencing factors, maturity level, and process capability.

1. Introduction

Software reuse is not only concerned with the technical aspects rather it is concerned with the non-technical aspects as well. It has been pointed by reuse researchers [16] that one needs to resolve the non-technical aspects in order to get an effective reuse program in place [6]. It requires incorporating the reuse tools and techniques formally into the software development life cycle by integrating it with the reuse processes and activities [6]. The other non-technical aspects that should be addressed may refer to the change in organizational structure, work practices, infrastructure requirements, economic issues, selling and buying of reusable components, payoffs and incentives, legal issues, intellectual rights issues [6][8].

The reason for continuous failure of software reuse over the years has been stated as the organizational, administrative, economic, and psychological impediments [21]. Although these organizational issues cannot be resolved just in once, but it requires a thorough and continuous analysis of newly adopted practices and making appropriate changes for improvement in the practices to seek better results [8][9].

The changes in the organizational structure will define new roles, responsibilities and work practices among the personnel that will create a better understanding in implementation of software reuse. Such changes will reduce the risk of adoption of reuse program and there remains an opportunity to always improve the reuse program by taking it on an incremental path. Not every source code, every software component can be reused [13]. It requires a proper analysis before making any component reusable. There exist some of the modern tools such as Gilligan tool suite that can transform the source code from its originating system and integrate into the newer system. Incorporating such tools in work practices will improve the reusability by reducing the time that is needed to transform the source code [13].

2. Problem Statement

The main problem in implementing reuse is not the technical rather it is the administrative problem. It requires changing the traditional work practices of the organization for the development of software. An implementation experience of Apte et al. (1990) reflected operational, strategic and monetary benefits when applying the reuse concepts for designing and implementing a bank information system. But this implementation experience highlighted the need of effective reuse strategy [6]. Prieto-Diaz (1991) pointed towards defining the necessary organizational structure for obtaining success in reuse program. It highlighted the need for teams with specific roles such as asset management, asset identification, maintenance, and development of assets [6]. A survey conducted by Lucredio (2008) on Brazilian software companies showed that organizational factor is one of the

important factors that affect the success of software reuse [16]. Garcia (2010) presented a detailed study on reuse adoption by software companies. It stated the deficiencies in previously implemented reuse adoption programs and specified the requirements that should be addressed [6]. The problems stated by the above mentioned researchers in their respective research studies related to reuse can be summarized as:

- Lack of formal processes or methods for software reuse.
- Practicing software reuse on ad-hoc basis.
- Lack of knowledge about introducing reuse program in an organization.
- Roles and activities are not clearly defined with respect to reuse.
- Unawareness about how to increment the reuse program towards systematic reuse.

3. Research Methodology

In order to resolve the above stated problems the following approach has been adopted:

- An extensive literature is studied in order to determine the problems related with the study.
- Exploration of existing alternatives for these problems.
- To propose a framework that should have the capability to overcome the deficiencies of previous models and to comply with the international standards & established practices.
- To measure the framework using a comparison based approach with other reference model.
- Compare the outcome of study to mark a clear difference between the study and the earlier stated models.

4. Proposing a New Organizational Framework for Component Reuse

The study proposes a framework called ‘Software Reuse Division’ or ‘SRD’ which acts as one composite solution to resolve the organizational issues. The proposed framework refers to the constitution of a new division in the organization or a department in general. All the issues related to software reuse will come under the jurisdiction of software reuse division. The new division would have the decision making power for all the matters related to software reuse in coordination with the top management. The division should be independent from other software developing divisions but it should be in complete coordination with them.

The sole objective behind the constitution of new division is to segregate all the issues related to software reuse. With this segregation, there will be more clarity in the roles and responsibilities of the personnel and more defined will be the work structure. It will also help to reduce the redundancy in roles, responsibilities and work practices. One should not consider the new division as the one responsible for developing components for reuse and with reuse. Rather it is responsible for getting the components developed through the developers. It directs other departments and communicates with other departments of the organization in concern with reuse related issues.

4.1. Need of SRD

The research studies conducted by reuse researchers [16] points towards the significance of organizational structure [6]. The MIT centre for Information Systems Research conducted a special session in June, 2010 and it determined that companies do not reuse because of the factors relating to behavior, and corporate culture. It concluded that the reason for continuing failure in adoption of organization wide reuse lies in the areas of organization and culture [20]. The reason beyond the creation of reuse division is that with better expertise, the efforts to reuse an asset will be minimized, creation of robust reusable assets. But the creation of such a division becomes possible at higher costs [4].

The creation of reuse division has been advocated on the basis of survey conducted by Daniel Lucrecio *et al.* in Brazilian software industry. The results reflected by the survey are not just for Brazilian industry, but are generalized to other countries and regions such as USA and Europe [16]. The survey results verify the fact that the organizations having independent teams for development of reusable assets obtained 100% success in their reuse program, while the organizations not having independent teams for development of reusable assets obtained only 63% success in their reuse program [16]. The survey data verifies the fact that creating a reuse division for reuse program is a good practice, with a strong influence on reuse success.

4.2. Pre-Requisites For SRD

There are some of the important tasks and activities that are required to be completed before setting up the new division and before the new division start functioning. These tasks and activities are commonly referred to as pre-requisites. All the pre-requisites are discussed in detail as follows:

4.2.1. Economic Survey

Economic survey is a process of estimating the costs and benefits included in the formation of new division. Before setting up the new division in the organization it is essentially required to conduct an economic survey. The economic survey is conducted in order to determine whether the formation of new division will benefit the

organization or not. If there does not exist a monetary benefit then there is no point of moving ahead with the formation of new division [9]. There exist number of economic models and metrics that can be adopted by the organization for conducting cost and benefit analysis. One specific model and metrics cannot suite all the situations and requirements. Therefore the selection of an economic model and metrics for conducting cost and benefit analysis depends upon the business requirements of the organization [6].

The costs and benefits included here in the survey can be of two types, such as, tangible and intangible [9]. The tangible costs include one-time cost of setting up the division and the continuous costs of operating the division. The intangible costs include the cost of services taken from other departments of organization, the cost of considering components for developing for reuse and development with reuse, the cost of searching components from library if nothing is procured [9]. The tangible benefits include the reduction of cost for repetitively developing the similar components, increased control over the activities related to reuse, better management and knowledge of components. The intangible benefits include increased quality of products, faster development of products, and the searching of reusable components becomes easier [9].

4.2.2. Number of Personnel Required

The requirement of personnel in new division refers to the amount of workforce required and the type of workforce required. By the amount of workforce it is meant that how many people are required to accomplish tasks of new division efficiently. By the type of workforce it is meant that people from which discipline are required such as analyst, technical, finance, legal, marketing etc [9]. The number of personnel required can be determined as per the business strategy of an organization and their customer demands [6].

4.2.3. Infrastructure and Technological Requirements

The infrastructure requirements are concerned with a separate workspace for the employees of ‘software reuse division’, hardware and other supportive equipments required for the conduct of operations. The technological requirements are concerned with hardware and software for the functioning of division, for management of component library, for reuse CASE tools, for coordination and communication inside and outside the organization [9]. The infrastructure and technological requirements are directly proportional to the personnel requirements and can be determined as per the structure of an organization and their customer demands [6].

4.3. The Structure of SRD

The structure of ‘software reuse division’ defines the composition of new division, that is, which type of people are working in the division such as technical, legal, marketing, finance etc. It also explains the hierarchy of personnel in the division. That is, the personnel belong to which level in the division. This section provides with the top-down structure of new division as presented in table 1. At the top level is head of division, next to him are vice-head of division, at the third level it comes the heads of sub-division included in the new division. These heads of sub-division work as the team leader having the responsibility of performing a specific function. Then it is the level of subordinate staff of each sub-division which is again parted into two such as senior subordinate staff and junior subordinate staff. The senior subordinate staff is the one having an adequate amount of experience and skill to perform while the junior subordinate staff includes the one without experience or with lesser amount of experience and skill to perform.

It is not necessary that each sub-division should have a team of senior and junior subordinate staff. It varies according to the nature, functionality, and requirement of sub-division. There may exist some other personnel as well which may work directly under the directions of head and vice-head of division but not under any of the sub-divisions. The level of hierarchy of such personnel will be determined by the role they are performing. The structure of the division is so fabricated that the reuse activities should be performed in an integrated way [16]. The functions of the whole division should be controlled centrally by the head of the division and the vice-head should assist him in doing so. The vice-head will share the burden of the head in order to ease the work load on head. The vice-head are required because managing a multiple no. of sub-divisions can be a complex and tedious task. Thus, some of the responsibilities of head of division can be divided to vice-head.

Table 1: Top-down structure of software reuse division

Top-Down Structure of Software Reuse Division		
Sr. No.	Levels	Role
1	Level-1	Head of Division
2	Level-2	Vice-Head of Division
3	Level-3	Head of sub-divisions
4	Level-4	Subordinate staff <ul style="list-style-type: none"> • Senior subordinate staff • Junior subordinate staff

4.4. Manifestation of Details of Each Level/Role

This section provides the details of each level or role included in the new division. These manifestations may include the responsibilities, rights and privileges, work flow.

4.4.1. Level-1/Head of Division

The head of division has the complete power and responsibility of the division. It takes instructions directly from top management and has right to give instructions to the personnel of division. The head is accountable for each and every action of the division [12] [19] [26]. The full control of the division lies with it. It has the final decision making power at the department level. In some cases that are concerned with inter-department level decision making the head of 'software reuse division' has some privileges related to issues of reuse. But when it comes to inter-department level the decision making power lies with other senior authorities [12] [19] [26]. Responsibilities of head of division are illustrated below [12] [19] [26]:

- Identifies reuse related tasks and activities.
- Assigns and allocates these tasks to the concerned sub-divisions
- Sets-up the deadline for completion of tasks.
- Assesses the tasks and performance of personnel.
- Determines the quality of workforce and also the quantity of workforce required.
- Recruits the required workforce.
- Functions as a bridge for coordination and communication within and outside the division.

4.4.2. Level-2/Vice-Head of Division

The vice-head of the division are subordinate to the head of division. It takes instructions directly from head of division and can give instructions to the personnel of division in compliance with the rights and privileges of head of division. It acts as a head of division in the absence of head of division and will have all the rights and privileges. Often it is supposed to assist the head in performing day-to-day operations and taking crucial decisions at the department level [12] [19] [26]. Responsibilities of vice-head of division are illustrated below:

- It can also identify the tasks and activities related to reuse and can brought it to the notice of head.
- Checks the deadline of tasks as directed by head.
- Checks the performance of personnel and assesses the work done by them under the instructions of head.
- Functions as a facilitator for head and other personnel of division.

4.4.3. Level-3/Head of Sub-Divisions

The sub-divisions should be created in order to classify the roles and operations within the division. There may exist sub-divisions belonging to a specific skill such as technical, legal, analyst, marketing etc. Each sub-division has a team of experts under the leadership of head of sub-division. The head of sub-division takes instructions directly from head or vice-head of division as accordingly. It gives instructions to the members of sub-division. It has the decision making powers at the sub-division level [12] [19] [26]. Responsibilities of head of sub-division are illustrated below:

- It is accountable for each and every action of sub-division.
- Final decision making power within the sub-division lies with it.
- Gets the task as assigned from head of division and delivers the work done.
- Assigns and allocates tasks and activities to the subordinate staff of sub division.
- Assesses the tasks and performance of subordinate staff and gives report to the head of division for further action.
- Provide guidelines to complete tasks and activities.
- It also coordinates and communicates with other sub-divisions of division.

4.4.4. Level-4/Subordinate Staff

The subordinate staff is a building block of the division. It is the one responsible for getting the tasks done. It takes instructions directly from head of sub-division. The subordinate staff is parted into two that is, senior and junior. The partition of senior and junior lies due to the adequate experience they possess. But they work as a team and in complete coordination with each other. Both belong to the same level of hierarchy [12] [19] [26]. Responsibilities of subordinate staff are illustrated below:

- They have no decision making powers.
- It takes instructions from the head of sub-division.
- Perform the tasks and activities as assigned by head of sub-division.
- Deliver the work done on the prescribed deadline.
- The work is done under the guidelines of head of sub-division.
- Coordinate with other members of sub-division.

4.5. Operations Performed by SRD

Operations define the process of accomplishing various tasks and activities that are required to perform for the functioning of the division. In this section it will be discussed that who will perform these operations and how these operations will be performed. The operations performed by 'software reuse division' can be broadly classified as per the nature of different sub-divisions. All these operations are discussed below:

4.5.1. Operations Performed by Analysis Sub-Division

The specific task of analysis related to reuse is performed by the analysis sub-division. It conducts analysis of in-house developed software components and non-in-house developed software components. While conducting an analysis of in-house developed software components it determines that whether they can be developed for reuse or not. Also it analyzes the in-house software components for development with reuse. The analysis sub-division will analyze the technical, legal, and financial feasibility of a software component during the course of analysis. It will prepare a detailed analysis report after conducting an analysis for a component. The report shall describe about the feasibility of a component to be reused or not [9].

While conducting analysis of non-in-house developed software components it often considers them for development with reuse but it can also consider them for developing for reuse. It will analyze the technical, legal, and financial aspects of a component during the course of analysis. The legal and financial aspects are analyzed significantly in this scenario. After conducting an analysis of non-in-house developed software components a detailed report will be prepared. The detailed report will consist of technical, legal, and financial aspects related to a software component [9].

4.5.2. Operations Performed by Technical Sub-Division

The operations related to technical sub-division can be observed from two view points, i.e., developing for reuse and development with reuse. It will consider the software components for these two aspects irrespective of the fact that they are developed in-house or not. In case of, developing for reuse, the technical sub-division will assess the present state of software component. While assessing the present state of software component it will determine

- The functionality of the component.
- How it fulfils that functionality.
- The design approach.
- The technological platform on which the component was built.
- Its compatibility with different hardware and software.
- Is necessary documentation available to understand the component?
- Can the component be transformed?
- Does the necessary technology and development facility exist to transform the component?

After assessing the present state of software component the technical sub-division will adopt the general approach to transform the component. That is, it will try to make the software component more generalized in nature. So, that it can be made specific to requirements easily as and when required. It will provide with complete guidelines that how to transform the component. The software component will be transformed by the developing department on the basis of guidelines provided [9].

In case of, development with reuse, the technical sub-division will assess the software component on the basis of requirements of undergoing project. While assessing the software component it will determine the present state of component as it was done in case of developing for reuse. But in addition to that it will determine

- The compatibility of technological platform of component on which it was built with the technological platform of the undergoing project.
- If it is not compatible, then to which technological platform it can be transformed.
- It will provide with complete guidelines to transform the component.
- After transformation, will the component adhere to accuracy, reliability, data security, and ease of access?
- After transformation, will the component comply with the design approach of undergoing project?

Here, the technical sub-division will adopt the more specific approach. That is, it will try to make the software component more specific to the nature of the project. In addition, to the above stated tasks the technical sub-division will perform the task of managing the component library. It will manage the in-house and non-in-house developed components in component library based on different classifications. Also, it will give report about the software components to the analysis sub-division.

4.5.3. Operations Performed by Legal Sub-Division

The legal sub-division will perform the legal tasks related to reuse of software components. These legal tasks comprise of copyrights, patents, trade secret, and licensing processes of software components. It is to be noted that the copyrights, patents, trade secret, and licensing paradigms of software components differ from that of

traditional software. It is because of the fact that traditional software serves a different purpose than that of a software component, the size and quality of software may distinguish from that of a software component, and also software acts as a stand-alone application whereas a software component can be a part of a stand-alone application [3]. The legal sub-division will finalize the terms and conditions related to attainment and grant of licenses of software components. While finalizing the licensing terms and conditions it will take care of facts related to ownership, payments, liability, support and warranties, and confidentiality. The assumptions regarding the purpose, interoperability, size, quality, function and flexibility, adaptability, indemnification and limitation of liability provisions, and support and warranty obligations provides the basis for driving the licensing terms and conditions [3]. The legal sub-division will cover the following subjects:

- It will attain copyrights, trade secret, and patent rights for completely in-house developed software components.
- It will attain and grant licenses for the reuse of software components.
- It will formulate and assess the base point for licensing terms and conditions.
- It will negotiate and finalize the licensing terms and conditions for the reuse of software components.
- It will inform developers of the organization and the analysis sub-division about the usage of concerned software components and also explains the risks associated in case of violations of license agreement and infringements in intellectual property rights.

In addition, to the above stated tasks, the legal sub-division will handle issues of license violation and issues related to infringement of intellectual property rights. If it finds such violations and infringements in license agreement and in intellectual property rights, it can sue the party in the court of law [3]. Also, it replies for notices and defends the organization for being sued in similar cases.

4.5.4. *Operations Performed by Finance Sub-Division*

The finance sub-division conducts an economic survey of given software components covering the cost and benefit aspects as prescribed by analysis sub-division. It will submit a report comprising the financial aspects of the software components to the analysis sub-division. It will coordinate with the legal sub-division regarding the payment of fees for obtaining copyrights, trade secret, and patents [3]. Also, it coordinates with legal sub-division while making the transactions of software components. These transactions can occur while acquiring and granting of software components. The transactions of software components include the transference of software components and the monetary transference as well. The operations related to monetary transference will come under the jurisdiction of finance sub-division. It will handle the single and multiple monetary transactions covering aspects related to costs, indemnification and limitation of liability provisions, support and warranty obligations etc [3]. Illustration of subjects under finance sub-division is as follows:

- It will accomplish the single monetary transactions of software components covering their costs, indemnification and limitation of liability provisions, support and warranty obligations etc. either while acquiring or granting a component.
- It will track the multiple monetary transactions of software components covering their costs, indemnification and limitation of liability provisions, support and warranty obligations etc. either while acquiring or granting a component.
- It will prepare a financial report for software components either on daily, weekly or on monthly basis. It will submit financial report to the head of software reuse division.

4.5.5. *Operations Performed by Marketing Sub-Division*

The operations under the marketing sub-division include the promulgation of completely in-house developed software components, promulgation of non-in-house developed software components if permissible, devising the marketing and ad-campaign strategy, execute the marketing strategy and ad-campaign strategy, analyze the market conditions and strategies with respect to reuse of software components at regular time intervals. It will also circulate the features of software components while advertizing for them. The objective behind the establishment of marketing sub-division is to promote the sale of reusable software components in the outside world. It will coordinate with the finance sub-division for financial assistance to its ad-campaigns. Coordination with other sub-divisions will be done as and when required. Illustration of operations by marketing sub-division is given below:

- To propagate the completely in-house developed software components in the world outside the developing organization.
- To propagate the non-in-house developed software components, if permissible.
- To devise the market and ad-campaign strategy.
- To execute the market strategy and ad-campaign strategy.
- Analyze the market conditions and strategies at regular time intervals with respect to reuse of software components.

- To coordinate with the finance sub-division for financial assistance to its ad-campaigns.
- To coordinate with other sub-divisions as well as and when required.

4.5.6. *Operations Performed by Renewals Sub-Division*

The operations under the 'ReNews' sub-division is mainly concerned with the propagation of reusable artifacts inside the organization on the basis of different projects being developed in the organization [5]. This sub-division in fact is an internal medium of communication related to reusable software components for an organization. To circulate the information about software components it can adopt different methods as it suits. But one method that has been suggested in addition to other methods is the concept of 'ReNews', i.e. Reuse Newsletter [5]. The concept of reuse newsletter can be implemented inside the organization by circulating it at relevant spots. The reuse newsletter should be developed and circulated through the 'ReNews' sub-division. 'ReNews' should include information about related to reusable artifacts, their specifications, reuse references, sources of reusable parts, reuse CASE Tools etc [5]. Illustrations of operations under ReNews sub-division are given below:

- Promote the reusable software components inside the organization in order to make the reuse of software components easier.
- Circulate the information and specifications about the reusable artifacts.
- Circulate the reuse newsletter within the organization that contains all relevant information about the reuse of software components.
- Provide information of relative reusable software components with respect to the different projects under development in the organization or for the projects under pipeline.

5. **Measuring Proposed Framework Using a Comparison Based Approach**

The proposed framework has been measured using a comparison based approach. It has been compared with the RiSE-Reference Model (RRM) [6]. A description of RiSE-Reference Model and the reason for choosing it for comparison has been given in below sections:

5.1. *Description of RiSE-Reference Model*

The RiSE-RM model was developed by RiSE group in thorough discussion with industry practitioners and software reuse researchers [6] [7]. The developers of the model studied the relative literature and conducted a survey in industry to tailor the model as per the requirements of industry and in compliance with the international standards. The RiSE-RM was developed to make improvements in development process, reuse process areas and practices [6]. The RiSE-RM uses the concept of maturity levels, process areas, maturity goals, and process results for its implementation.

The RiSE-RM model is used to determine goals, key practices, process areas that should be adopted by companies for implementing a systematic reuse approach. The RiSE-RM achieves two main goals, i.e., it helps in assessment of maturity level of a company in terms of reuse practices, and helps the organizations to improve their productivity, quality, and competitiveness through software reuse practices [6]. The model is structured with five elements such as maturity level, process areas, process purpose, process results, and work products. The element maturity level is used to assess the maturity of an organization for implementing the reuse processes. The process areas highlight the key process areas where the organization should focus on for improving its reuse practices. The process purpose defines the objective that a process should achieve. Process results refer to the achievement of process purpose. A work product is a product produced from a process [6].

5.2. *Why RiSE-Reference Model*

The reason for choosing RiSE-Reference Model is due to its compliance with various international standards. The model complies with the international standards such as ISO/IEC 12207:2008, CMMI-DEV, and MPS.BR as per the requirements of software reuse community [6]. The model fulfills the needs of reference process models as defined by the international standard 15504-2 [6].

In standard CMMI-DEV some of the sub-practices are scattered through the model, whereas the RRM model covers this deficiency [6]. There exist some other reference models as suggested by Frakes and Isoda (1994), Rine (1997), Rine and Sonnemann (1998), Lucredio et al. (2008), but these models lack in suggesting any incremental path for implementing software reuse practices in an organization [6].

5.3. *Comparison of Proposed Framework with RiSE-Reference Model*

While comparing the proposed framework i.e. 'Software Reuse Division' with RiSE-Reference Model, the maturity level of the proposed framework will be identified and the capability of the processes will be determined. For identifying the maturity level of the proposed framework, the influencing factors of SRD (Software Reuse Division) will be mapped against the process areas of the RRM (RiSE-Reference Model) [6]. By this mapping technique, the maturity level of the SRD (Software Reuse Division) can be identified. For determining the capability of the processes of the proposed framework, the processes of the SRD (Software Reuse Division) are mapped against the common process attributes affiliated with process capability of the

RRM (RiSE-Reference Model) [6]. By this mapping technique, the capability of the processes of the proposed framework can be determined and the process areas where improvements are required can be focused.

A list of different influencing factors of the proposed framework with their description has been presented in table 2 [24].

Table 2: Identified influencing factors of proposed framework

Sr. No.	KPA	Influencing factors	Description of influencing factors
1	Business	Domain Focus	The term domain refers to the environment in which the development organization produces solutions for. The domain focus is an instrument for indicating the similarities between the products of a same domain.
2	Organization	Top management support and organizational mechanisms	Top management support factor influences the success ratio of reuse program in the organization. The level of management support can be indicated with the help of indicators. It is the responsibility of top management to define the organizational mechanisms to enforce the desired work practices. The organizational mechanisms refer to the practices followed in context of reuse program.
3	Organization	Organizational structure and roles	The organizational structure defines the composition of personnel with different skill sets. Personnel with different skill sets cover different responsibilities. The responsibilities elaborate the roles played by personnel.
4	Organization	Communication channels and organizational support	This factor indicates the level of various communication channels involved for propagating information within and outside the organization.
5	Process	Reuse analysis	Reuse analysis is an indicator of systematic software reuse process. It uncovers each and every aspect of reusable artifacts such as technical, legal, and financial.
6	Process	Financial management	Measuring reuse related organizational activities is another indicator of systematic software reuse. When these activities are measured, the performance level of organization can be determined which can then be linked to cost models and as a result the cost benefit analysis can take place.
7	Process	Quality management	Quality management refers to the processes and reuse activities that correspond to quality models and in compliance with the quality standards.
8	Process	Legal asset management	It is also considered as an indicator of a systematic software reuse. Managing the legal aspect of reusable software components can prove to be a useful asset such as licensing, trademarks, patents.
9	Process	Information management	Information management is also an indicator of a systematic software reuse. Propagation of reuse related information can be a turning point while development and attainment of reusable software components.
10	Process	Marketing management	Marketing management factor refers to the promotion of reusable software components outside the organization in order to generate revenue by selling them. Strategic marketing plays a vital role in it.
11	Process	Configuration management	In this factor, the configuration of reusable software components is determined and used in projects under development. Besides this, the components are made general so it can fit into a more number of projects.
12	People	Producer expertise and experience	It refers that the developing organization should have a certain amount of expertise and experience in implementing software reuse processes and conducting reuse activities. Expertise and

			experience should be at a level that the organization should be able to produce more generic components.
13	People	Consumer expertise and experience	It refers that the organizations who are involved in development through reuse should have adequate skills and experience to integrate the components as per the requirements of the project.
14	Technology	Repository support	Repository support is concerned with the usage of a repository or a component library for software reuse. The storage, searching, and retrieval factors are associated with it.
15	Technology	CASE tool support	It refers to the programming tools that can play a vital role in performing reuse activities efficiently.
16	Technology	Communication tool support	It refers to the need of communication tools for the purpose of communicating useful information about the reusable assets in an effective and an efficient way.

The RiSE-Reference Model consists of a large number of key factors which corresponds to their respective process areas. Now, table 3 depicts the relation between key factors and process areas of the RRM have been given below [6]:

Table 3: Relation between key factors & process areas of RRM [6]

Key Factors	Process Area
Software organization and team size	-
Project team experience	MRU
Software reuse education	BRT; MRU; REI
Rewards and incentives	RGD
Independent reusable assets development team	BRT; MRU
Product family approach	DAN; DDE; DRZ; DTT; RAP
Kind of software developed	DAN; DDE; DRZ; DTT
Application Domain	RAI; REI
Software development approach	BRT; RAI; DWR; REI
Programming language	BRT
Repository systems usage	RAM
CASE tools usage	BRT; RAM
Quality models usage	-
Systematic reuse process	DAN; DDE; DRZ; DTT; RAP
Kind of reused assets	RAI; ROS; RIF
Origin of reused assets	RAI; ROS; DAN
Previous development of reuse assets	BRT; RAI; DWR; TRA; REI; DAN
Specific function in the software reuse process	DWR; REI
Software reuse measurement	DRS
Software certification process	DRS
Configuration management of the reusable assets	TRA
Business planning and economic issues	RCM
Management support	MRU
Reuse goals definition	RGD
Software reuse history (knowledge reuse)	RAI; DWR
Legal, Contractual, Accounting considerations	RCM
Reuse adoption process	RGD; MRU; REI
V&V processes for reuse assets	DWR; DTT; SAD
Improving organizational reuse process	RGD; REI; RAP
Low up-front risk of reuse adoption	-
Reusable artifacts standards	BRT; RAI; ROS; RIF; DRS
Traceability between reusable artifacts	TRA; DAN; DDE; DRE; SAD

In the table given above it can be seen that some of the key factors do not correspond to the process areas. The factor '*Software organization and team size*' does not correspond to a process area because this is an issue concerned with the organizations as per their business strategy and their customer demands [6].

It has been experienced while studying the literature that the factor '*Quality models usage*' is not directly related to software reuse. Though, some of the key practices described in the quality models can help in improving the reuse activities [6].

The factor '*Low up-front risk of reuse adoption*' does not relate to any process area, rather it is mainly concerned with the organization. The RRM model tries to implement the reuse adoption process in an incremental way which helps in reducing the risk of reuse adoption [6].

5.3.1. Mapping of Influencing Factors of SRD Framework with Process Areas of RRM Model

A tabular representation of mapping of influencing factors of SRD framework with the process areas of RRM model has been given in table 4:

Table 4: Mapping of influencing factors of SRD framework with process areas of RRM model

Mapping influencing factors with process areas		
S. No.	Factor Name	Process Area
1	Domain Focus	RAI; REI
2	Top management support and organizational mechanisms	MRU
3	Organizational structure and roles	DWR; REI
4	Communication channels and organizational support	BRT; MRU; REI
5	Reuse analysis	RCM; RGD
6	Financial management	RCM
7	Quality management	AQC
8	Legal asset management	RCM
9	Information management	RAI; DWR
10	Marketing management	MCA
11	Configuration management	TRA
12	Producer expertise and experience	MRU
13	Consumer expertise and experience	RAI; REI
14	Repository support	RAM
15	CASE tool support	BRT; RAM
16	Communication tool support	TRA; DAN; DDE; DRZ; SAD

After mapping the influencing factors of SRD framework with process areas of RRM model, the maturity level of each influencing factor can be identified. A pictorial representation of the process areas and their corresponding maturity levels as stated by RRM model has been given in figure 1[6]:

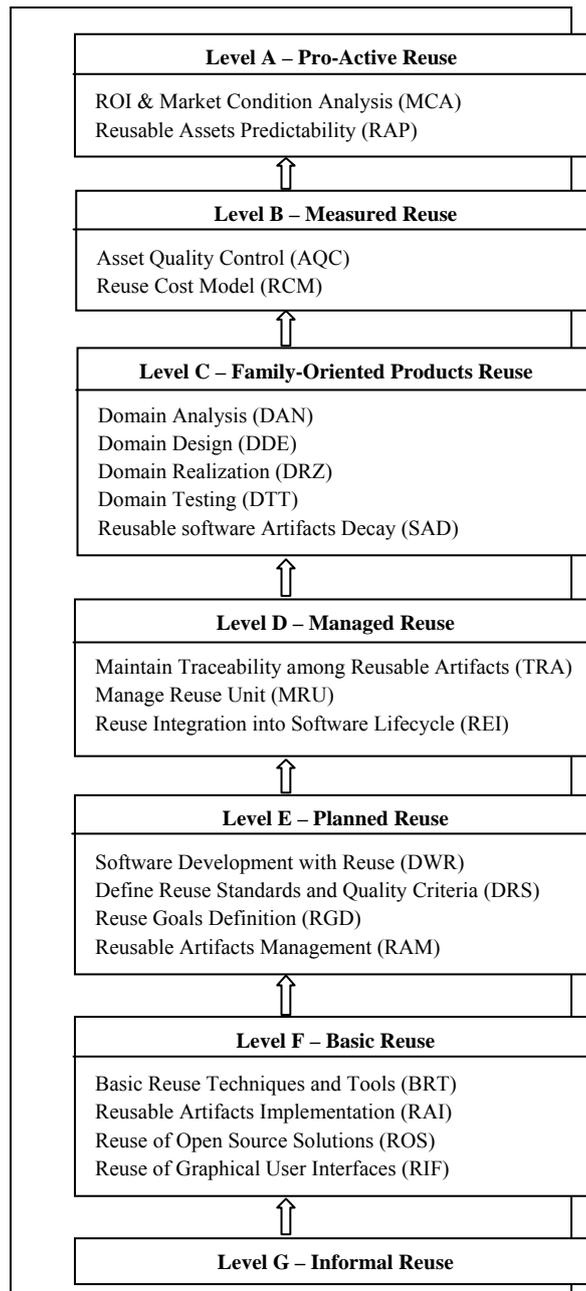


Figure 1: Process areas and their corresponding maturity levels [6]

The process areas of RRM model will be instrumental in identifying the maturity level of each influencing factor of SRD framework. Table 5 depicting maturity level of each influencing factor of SRD framework has been given below:

Table 5: Maturity level of each influencing factor of SRD framework

KPA	Influencing factor	Maturity Level	Level No.
Business	Domain focus	Level D	4
Organization	Top management support and organizational mechanisms	Level D	4
	Organizational structure and roles	Level D	4
	Communication channels and organizational support	Level D	4
Process	Reuse analysis	Level B	6
	Financial management	Level B	6
	Quality management	Level B	6
	Legal asset management	Level B	6
	Information management	Level E	3
	Marketing management	Level A	7
	Configuration management	Level D	4
People	Producer expertise and experience	Level D	4
	Consumer expertise and experience	Level D	4
Technology	Repository support	Level E	3
	CASE tool support	Level E	3
	Communication tool support	Level C	5
Average		Level D,C	4,5

In the table 5 the maturity level of each influencing factor of SRD framework has been identified, also the level no. corresponding to that particular level has been identified. The average maturity level and the level no. of the complete SRD framework have been identified. It shows that the maturity level of the SRD framework lies between level D & C and between level no. 4 & 5 respectively. It shows that the model is at managed reuse or family-oriented products reuse level. For some of the influencing factors it shows higher maturity level. This means that the model is approaching for higher maturity levels with focus on improvements.

From table 5, it is clear that the overall maturity level of proposed model is 4.5. This shows that the proposed framework has a moderate maturity level.

5.3.2. Mapping the Processes of SRD Framework with Common Process Attributes Affiliated with Process Capability of RRM Model

Firstly, a table depicting the process capability and the corresponding capability level of RRM model has been given in table 6 [6]:

Table 6: Process capability and corresponding capability level

Capability Level	Process capability
0	Incomplete Process
1	Performed Process
2	Managed Process
3	Established Process
4	Predictable Process
5	Optimizing Process

The table 6 contains six levels of process capability. A description of each level is presented as follows:

- Level 0 – Incomplete Process
 - The process is not implemented, or fails to achieve its purpose. At this level there is a little or no evidence of any systematic achievement of the process purpose [6].
- Level 1 - Performed Process
 - It determines the extent to which the process purpose is achieved. The achievement of this attribute involves the defined outcomes [6].
- Level 2 – Managed Process
 - Performance Management – It determines the extent to which the performance of the process is managed. The achievement of this attribute involves the planning, monitoring and adjusting the process performance [6].

- Work Product Management – It determines the extent to which the work products produced by the process are appropriately managed. The achievement of this attribute ensures that work products are appropriately established, controlled and maintained [6].
- Level 3 – Established Process
 - Process Definition – It determines the extent to which the process is established as a standard process within the organization. The achievement of this attribute involves the definition of the process in terms of required competencies and roles for performing a process, required infrastructure and work environment, methods for monitoring its effectiveness and suitability and tailoring guidelines [6].
 - Process Deployment – It determines the extent to which the process is effectively deployed as a tailored instance of the standard process. The achievement of this attribute is reflected in the fidelity to the standard process, the effective deployment of resources to the implementation of the process, and the collection and analysis of data for understanding and refining the behavior of the process [6].
- Level 4 – Predictable Process
 - Process Measurement – It determines the extent to which process measurements are used to ensure that performance of the process supports the achievement of defined business goals. The achievement of this attribute is concerned with the existence of an effective system for the collection of measures relevant to the performance of the process and the quality of the work products. The measures are applied to determine the extent of achievement of the organization’s business goals [6].
 - Process Control – It determines the extent to which process is quantitatively managed to produce a process that is stable, capable, and predictable within defined limits. The achievement of this attribute implies the application of analysis and control techniques to ensure that the process performs within defined limits and that corrective action are taken to address deviations [6].
- Level 5 – Optimizing Process
 - Process Innovation – It determines the extent to which changes to the process are identified from analysis of variation in performance, and from investigations of innovative approaches to process definition and implementation. The achievement of this attribute is concerned with the existence of a proactive focus on continuous improvement in the fulfillment of both current and projected business goals [6].
 - Process Optimization – It determines the extent to which changes to the definition, management and performance of the process result in effective impact that achieves the relevant process improvement objectives. The achievement of this attribute is concerned with an orderly and proactive approach to identifying and introducing appropriate changes to the process minimizing undesired disruption, evaluating the effectiveness of changes and making adjustments as necessary [6].

Now, the processes of SRD framework as identified from the proposed framework have been given in table 7:

Table 7: Processes of SRD framework

Processes of SRD framework
<ul style="list-style-type: none"> • Process of analysis • Process of component maintenance • Process of configuration management • Process of repository management • Process of legal asset management related to a component • Process of managing financial assets of a component • Process of creating business and marketing strategy • Process of managing information of reusable assets

When these processes of proposed framework are mapped against the process attributes of process capability of RRM model, it has been determined that the processes of proposed framework relate to the Level 3 – Established Processes. This is because the proposed framework satisfies all the requirements of level 3. The processes of proposed framework have roles defined in them to accomplish these processes. The people with adequate experience and expertise are assigned to perform these processes which are useful in making processes competent. A dedicated team of experts for each process creates the required work environment. A head of sub-division is provided for each process in order to monitor and provide guidelines.

Thus, the processes of the proposed framework have the capability level 3, i.e. established processes. This capability level of the processes is more than the moderate level. So, it can be said that the processes are capable enough to fulfill their purpose, achieve the desired results, and produce the required work artifacts [6].

6. Validity

Here, the three kind of validity is discussed: construct validity, internal validity, and external validity [16].

6.1. Construct Validity

Construct validity considers whether the models and metrics used in the study are a valid representation of real world under study [16]. In this case of study, it refers to the quantitative and qualitative indicators that are used to establish a relation between the factors and reuse success [16]. As it can be seen in section 4.1, the organizations having independent teams for software reuse have obtained 100% success whereas the organizations not having independent teams for software reuse have obtained only 63% success [16]. This clearly shows the relation between having an independent reuse team and reuse success. Also the session conducted by MIT centre for Information Systems Research in June, 2010 pointed out that the reason for continues failure of software reuse lies in the area of organization and corporate culture [20]. Finally, these quantitative and qualitative factors form the basis for constructing a new reuse framework.

6.2. Internal Validity

Internal validity considers whether the experimental design is able to support conclusions [16]. In this case of study, a comparison based approach is followed to measure the proposed framework. The comparison is made with RiSE – Reference Model. While comparing the proposed framework with RiSE – Reference Model a method of descriptive analysis is adopted [16]. The descriptive analysis is performed by mapping the influencing factors and processes of proposed framework against the maturity level and capability level respectively of RiSE – Reference Model.

The threat to internal validity can be the biased identification of influencing factors and processes of proposed framework. Also, the threat can be due to irrelevant mapping. But the RiSE – Reference Model complies with international standards and work practices. Therefore, after comparison when the proposed framework matches with related maturity level and capability level of processes the framework satisfies the requirements meant for international standards and practices.

6.3. External Validity

External validity in this case of study, refers to the selection of a specific model for comparison [16]. There could be other existing models that can be chosen for comparison. The reason for this argument is that the RiSE – Reference Model complies with international standards and it has been developed in thorough discussion with industry practitioners and reuse researchers [6]. There may exist very less chance to question the validity of RiSE – Reference Model. If the RiSE – Reference Model is valid then the results generated after comparison with it should itself be valid. If the RiSE – Reference Model fulfills the criteria for international standards and work practices then the proposed framework should also comply with that. Thus, it can be stated that the impact of external validity on the proposed framework should be minimum [16].

7. Related Work

This section reflects the contributions of this study in the field of software reuse research. For this purpose, excerpts from the existing studies are presented.

Chavez in 1998 studied the state of licensing paradigms in software industry, where it found that the licensing paradigms are changing from software licensing to component licensing. In the study Chavez elaborated certain prospects such as license granting, mode of payment, issues of ownership and liability, assumptions regarding support and warranties, and the provisions of confidentiality [3].

In 2001 Fichman and Kemerer established the fact that systematic reuse is one of the ways to implement reuse program in an organization. It states that many of the organizations have started using systematic reuse program and in that course many reuse frameworks have been developed. It also explores the compatibility of incentives with reuse programs as its main objective [4].

Garcia et.al in 2007 realized that the software reuse practices are adopted on ad-hoc basis and they are not explored to its full potential. It then suggested systematic reuse as an approach to improve productivity and quality. It proposes a ‘Rise Maturity Model’ that consists of features of continuous improvement and reuse adoption in a systematic way [7].

Spoelstra in 2010 investigated the problems that existed in the conventional development practices related to software reuse. During investigation it determined that the practices are not systematic for reuse, they are not explored to its full potential, also they are agile in present course of context. To override these deficiencies it gives a solution tool that instigates top management for continuous improvement in development methodologies

related to reuse, increases maturity level of organization, and measure the performance level using metrics set [24].

In the year 2010 Garcia provided a model as a solution for adoption of software reuse in an incremental and systematic way. The model takes an account of different problems faced during the implementation of software reuse in an organization. In order to overcome those problems it proposes a model that complies with several international standards and work practices. The model consists of a maturity level, a process capability level that helps the organizations in assessing their maturity to implement reuse and determine the capability level of processes of the organization [6].

Holmes and Walker in 2012 developed a tool suite called Gilligan. It enables the developers to systematically investigate the boundaries of a source code in order to determine that whether it can be reused or not. It has been developed after evaluating it over the years both quantitatively and qualitatively [13].

Soora in 2014 floated the idea of domain engineering as one of the techniques that act as a solution framework for the implementation of software reuse. Domain engineering is one of the key ideas for the implementation of systematic software reuse [23].

The concept presented by Chavez guides for the creation of legal sub-division in the proposed framework. The study tries to systematize the process of software reuse by creating a dedicated team of experts as per their expertise for software reuse. In Most of the previous studies it has been pointed out that organization is one of the factors in implementation of software reuse. The solutions developed earlier were individualistic in their approach of development. They focused just on one or two aspects while designing a solution. The proposed framework in this study has taken an integrated approach to design a solution framework for reuse implementation in an organization. It advocates for the creation of dedicated team on the basis of expertise and skills. It defines the roles and responsibilities as per the structure and expertise. It also highlights the significance of communicating information on reuse artifacts within the organization.

8. Conclusion and Future Work

The study proposed a framework termed as ‘software reuse division’. The proposed framework is a result of the systematic and dedicated approach to address the problems concerning reuse of software components. In the framework the roles and responsibilities of personnel have been defined. The structure and hierarchy that defines the flow of work has been stated clearly. The operations that come under the jurisdiction of framework have been figured out. Thus the framework can be considered as an integrated effort to address the reuse problems of software components. The framework has been measured considering various influencing factors that have been identified on the basis of key process areas. A comparison based approach has been followed to measure the framework. During the course of comparison, the maturity of the framework has been identified and the capability of the processes has been determined. The results of measurement are obtained by performing a descriptive analysis of comparison between SRD framework and RRM model.

In future, the work can be done to improve the state of the proposed framework. That is, the improvements and effective changes can be made to take the framework to the higher levels of maturity. The processes can be structured and mechanized in such a way that the capability level of processes should increase. It can be seen that both the maturity level and the capability level of processes of the proposed framework are proportional to each other. Thus, the increase in level of one will affect the level of another. Also, the work in the area of domain engineering can be done. Domain engineering will facilitate the software components in such a way that the components belonging to a particular domain should be treated as per the requirements of that specific domain.

References

- [1] Bauer V., Eckhardt J., Hauptmann B., and Klimek M., (2014), “An Exploratory Study on Reuse at Google”, in SER &IPs 2014 Proceedings of the 1st International Workshop on Software Engineering Research and Industrial Practices, ACM New York, USA, ISBN: 978-1-4503-2859-3, pp.14-23.
- [2] Bhatia M.K., Kaur R., Chabra S., and Kaur P., (2007), Research Methodology and Statistical Methods, Kalyani Publishers, New Delhi, ISBN: 978-81-272-3961-9, Chap. 21, pp. 424-441.
- [3] Chavez A., Tornabene C., and Wiederhold G., (1998), “Software Component Licensing: A Primer”, IEEE Software, 15(5), ISSN: 0740-7459, pp. 47-53.
- [4] Fichman R.G., and Kemerer C.F., (2001), “Incentive compatibility and systematic software reuse”, The Journal of Systems and Software, 57(1), ISSN: 0164-1212, pp. 45-60.
- [5] Frakes W.B., (1991), “Software Reuse: Payoff and Transfer”, in Proceedings of the 8th AIAA Computing in Aerospace Conference, pp. 829-831.
- [6] Garcia V.C., (2010), “RiSE Reference Model for Software Reuse Adoption in Brazilian Companies”, from web site http://ivanmachado.com.br/research/rise/thesis/files/2010_ViniciusGarcia_phd.pdf accessed on 25/03/2016.
- [7] Garcia V.C., Lucredio D., Alvaro A., Almeida E.S.D., Mattos Fortes R.P.D., and Lemos Meira S.R.D., (2007), “Towards a Maturity Model for a Reuse Incremental Adoption”, from web site <http://www.ic.unicamp.br/sbcars2007/tecnicas/files/sbcars2007-garcia-toward.pdf> accessed on 21/08/2015.
- [8] Gill N.S., (2003), “Reusability Issues in Component-Based Development”, ACM SIGSOFT Software Engineering Notes, 28(4), ISSN: 0163-5948, pp. 1-4.
- [9] Gill N.S., (2009), Software Engineering, Khanna Book Publishing Company, New Delhi, ISBN: 978-81906116-3-3, Chap. 20,21.

- [10] Grinter R.E., (2001), "From Local to Global Coordination: Lessons from Software Reuse", in GROUP '01 Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, ISBN: 1-58113-294-8, pp. 144-153.
- [11] Griss M.L., (1993), "Software Reuse: From Library to Factory", IBM Systems Journal, 32(4), ISSN: 0018-8670, pp. 548-566.
- [12] Griss M.L., (1996), "Systematic Software Reuse: Architecture, Process and Organization are Crucial", from web site <http://martin.griss.com/pubs/fusion1.htm> accessed on 30/03/2015.
- [13] Holmes R. and Walker R.J., (2012), "Systematizing pragmatic software reuse", ACM Transactions on Software Engineering and Methodology, 21(4), pp. 20:1-20:44.
- [14] Joshi A., (2007), "A Model for Effective Asset Reuse in Software Projects", from web site <https://www.infosys.com/IT-services/application-services/white-papers/Documents/asset-reuse-model.pdf> accessed on 26/03/2016.
- [15] Linda P.E., Bashini V.M., and Gomathi S., (2011), "Metrics for Component Based Measurement Tools", International Journal of Scientific & Engineering Research, 2(5), ISSN: 2229-5518, pp. 199-204.
- [16] Lucredio D., Brito K., Alvaro A., Garcia V.C., Almeida E., Fortes R., and Meira S.L., (2008), "Software Reuse: The Brazilian Industry Scenario", The Journal of Systems and Software, 81(6), pp. 996-1013.
- [17] Pennings J.M., (1973), "Measures of Organizational Structure: A Methodological Note", The American Journal of Sociology, 79(3), pp. 686-704.
- [18] Poulin J.S., (1998), "Selling Software Reuse to Management", from web site <http://jeffreypoulin.info/Papers/JCIS98/sellingreuse.html> accessed on 25/03/2016.
- [19] Pressman R.S., (2001), Software Engineering-A Practitioner's Approach, 5th ed., McGraw-Hill Inc., New York, ISBN: 0073655783, Chap. 27.
- [20] Rehesaar H., (2011), "Capability Assessment for Introducing Component Reuse", 12th International Conference on Software Reuse, LNCS 6727, pp. 87-101.
- [21] Schmidt D.C., (2006), "Why Software Reuse has Failed and How to Make it Work for You", from web site <http://www.cs.wustl.edu/~schmidt/reuse-lessons.html> accessed on 26/03/2016.
- [22] Slyngstad O.P.N., Gupta A., Conradi R., Mohagheghi P., Ronneberg H., and Landre E., (2006), "An Empirical Study of Developers Views on Software Reuse in Statoil ASA", in ISESE'06 Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, ISBN: 1-59593-218-6, pp. 242-251.
- [23] Soora S.K., (2014), "A Framework for Software Reuse and Research Challenges", International Journal of Advanced Research in Computer Science and Software Engineering, 4(10), ISSN: 2277-128X, pp. 441-448.
- [24] Spoelstra W., (2010), "Reusing Software Assets in Agile Development Organizations-A Management Tool", from web site http://essay.utwente.nl/59917/1/MA_thesis_W_Spoelstra.pdf accessed on 30/03/2015.
- [25] Therriault R. and Nederveen K., (1994), "Industry Versus DoD: A Comparative Study of Software Reuse", from web site <http://www.dtic.mil/dtic/tr/fulltext/u2/a285708.pdf> accessed on 26/03/2016.
- [26] Tran Q., Tian Y., (2013), "Organizational Structure: Influencing Factors and Impact on a Firm", American Journal of Industrial and Business Management, 3(2), ISSN (O): 2164-5175, pp. 229-236.
- [27] Wu z., Tang J., Kwong C.K., and Chan C.Y., (2011), "A Model and its Algorithm for Software Reuse Optimization Problem with Simultaneous Reliability and Cost Consideration", International Journal of Innovative Computing, Information and Control, 7(5B), ISSN: 1349-4198, pp. 2611-2621.