International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 03 - Issue 02 || February 2018 || PP. 05-09

Enhanced Code Reuse Engineering

R. Anitha
Research Scholar,Department of Computer Science,
Sri Ramakrishna College of Arts and Science for Women, India

Dr. B. Shanmugapriya
Assistant Professor,Department of Computer Science,
Sri Ramakrishna College of Arts and Science for Women, India

Abstract: Code reuse is a critical strategy for all software development groups. By reusing code while moving to the next generation platform, corporations can leverage their existing software investment and lessen time to market. Many companies are struggling to fully implement code reuse throughout their organization. In order to achieve efficient and methodical code reuse, organizations must integrate this goal into their culture. Reusing code provides the greatest benefits to an organization if it is done systematically, rather than sporadically and opportunistically. There are many issues that can prevent systematic code reuse, both technical and non-technical. The project software library modelling is used to enhance the code reusable strength. The software library modelling is the revolutionary method for software developers. It will reduce the time of the software developers and increase the efficiency of the software. Initially a software transaction library has been created for the previous software list. The list contains information like name of the software, language use, number of employee used, time taken to complete the project, accuracy and efficiency of the project, number of modules and etc. All the information will be considered for the forth coming calculations. Also the information will be update for current working software. Now the software library modelling gives the best information from the earlier content. This makes the project manager to assign software and calculate the time period of the software to the team. Most of the code suggestions will be generated by the software library modelling. For example while getting the new orders from the client, the project manager need to work out the time frame and cost to the client. At that time more confusion may occur in assigning the employee, calculating the cost, managing the time frame and etc. In addition with the employee flow modelling can be done using this method. All the employ's performance will be monitored and rated by this modelling method. The data will be used for the further calculations like employee work assignment, critical work assignment and emergency work assignment and etc. The software library is a good example of code reuse. Programmers may decide to create internal abstractions so that certain parts of their program can be reused, or may create custom libraries for their own use. Some characteristics that make software more easily reusable are modularity, loose coupling, high cohesion, information hiding and separation of concerns.

Keywords: Code Reusability, Software engineering, Software Library Model, Project Management, Requirement Analysis

1. Introduction

There is an issue of figuring out how to rank gathering of articles by joining the inclination capacities or from accumulation of positioning. This issue of consolidating inclinations emerges in a few applications, for example, with that of joining the after effects of various web indexes, or the community oriented separating issue of positioning motion pictures for a client in view of the motion picture rankings given by different clients [1]. A group of learning calculations of regularization is proposed and it enables us to neglect the geometry of the negligible appropriation. Concentrate on a semi-administered structure that fuses marked and unlabeled information in a universally useful student. Some diagram learning calculations and standard strategies including bolster vector machines and regularized slightest squares can be acquired as extraordinary cases [2]. This paper intends to direct an investigation on the leastwise way to deal with figuring out how to rank. This approach takes in a positioning capacity by taking individual records as examples and also it limits a misfortune work anticipated rundown and the ground-truth list [3].From the approach of different machine learning strategies, e.g., ranking SVM the figures out how to rank calculations demonstrated their promising exhibitions in data recovery, particularly web seek [4][5][6].

1.1 Component Based Reusability

The goal of component based reuse engineering is to construct a standard collection of reusable components. Instead of reinventing the wheel each time, in future all software will be constructed by choosing a standard architecture and standard reusable framework and inserting standard reusable code artefacts into the

hot spots of the framework. The software products will then be built by composing reusable components. Ideally, this will be done using an automated tool.

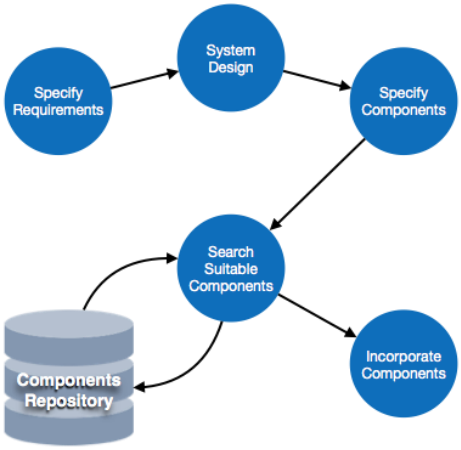


Fig:1. Component Reuse

The figure describes many advantages that accrue through the reuse of code artefacts, design patterns, and software architectures. Hence, achieving component based software engineering would solve numerous problems in software development. In particular, it would lead to order of magnitude increases in software productivity and quality and decreases in time to market and maintenance effort. Unfortunately, the state of the art with regard to reuse is currently far from this ambitious target. In addition, component-based software construction has many challenges, including definition, standardization, and retrieval of components. However, researchers in many centers are actively engaged in trying to achieve the goal of component-based software engineering.

2. Proposed Method

2.1 Efficient Code Reusability

The traditional reason for promoting reuse is that it can shorten the development process. For example, a number of major software organizations are trying to reduce the time to half the time needed to develop a new product, and reuse is a primary strategy in these endeavours. However, as reflected in every $1 spent on developing a product, $2 or more are spent on maintaining that product. Therefore, second important reason for reuse is to reduce the time and cost of maintaining a product. In fact, reuse has a greater impact on post delivery maintenance than on development. Suppose 40 percent of a product consists of components reused from earlier products then this reuse is evenly distributed across the entire product. That is, 40 percent of the specification artefacts consist of reused components, 40 percent of the design artefacts, 40 percent of the code artefacts, 40 percent of the manuals, and so on. Unfortunately, this does not mean that the time to develop the product as a whole will be 40 percent less than it would have been without reuse. First, some of the components have to be tailored to the new product and that one-quarter of the reused components are changed. If a component has to be changed, then the documentation for that component also has to be changed. Furthermore, the changed component has to be tested. Second, if a code artefact is reused unchanged, then unit testing of that code artefact is not required. However, integration testing of that code artefact still is needed. If 30 percent of a product consists of reused unchanged components and further 10 percent are reused changed, then the components time needed to develop the complete product is only about 27 percent. Suppose 33 percent of a software budget is devoted to development, and reuse reduces development costs by 27 percent, then the overall cost of that product throughout 12 to 15 year lifetime is reduced by 9 percent as a consequence of reuse.

3. Methodology

3.1 Initiating the Working Model

The primary module of this application is to create data for a company with the company's basic details. This module is mainly enabled for admin who creates the company data base. It consists of username, password and other basic details. In this module admin should enter the company name, address, phone number, email id and the working domain (software or hardware) of the company. After creating the details of the company the admin should create the employee by giving their basic details such as name, qualification, date of birth, phone number and email id. If the employee is an experienced person his/her year of experience along with his/her designation in the previous organization should be entered. The administrator should also specify

the department such as field work, analysis, design, coding, etc where the employee should works.This module decides whether the employee can be a team leader or a project leader or a developer or a tester or an employee is just a fresher.

3.2 Software Library Management

The main objective of this project is to reduce the work of the software developers in a software company. The work will be allotted for each person automatically. When developing a large project lot of confusion will occur, and some issues will arise between modules and the person who developed it most of the time the employees are facing insufficient data in work allocations. In order to reduce these problems is introduced work flow mining. All the modules should be given as input and it will automatically indicate how many persons need to complete this project, and how much time will be taken to complete the module. The above mentioned problems could be overcome using this method.

3.3 Code Reuse Calculation in Real-Time

Reusability depends on portability, adaptability, understandability, maintainability and reliability. Asthe project deals with code reliability is considered as an issue which affects reusability. Reliability is measured in terms of mean time and fault which is measured during the execution of the program. Considered as with respect to the changes is the business environment adaptability is one of the important factor or weapon. Business market situation is changing frequently, hence the software system should be adaptable to satisfy this requirement.

3.4 Work Assignment and Performance

The administrator should perform project scheduling. Once the project is selected the system should display the project details like start date and end date of the project and the language chosen for this project. The phases of the project should be updated by admin based on the work achievement. If the phases of the code are updated in the system, the system should generate the percentage of work completion status. This process is responsible for checking the work status phase wise. The admin should be able to check the employees those who are assigned work to complete.

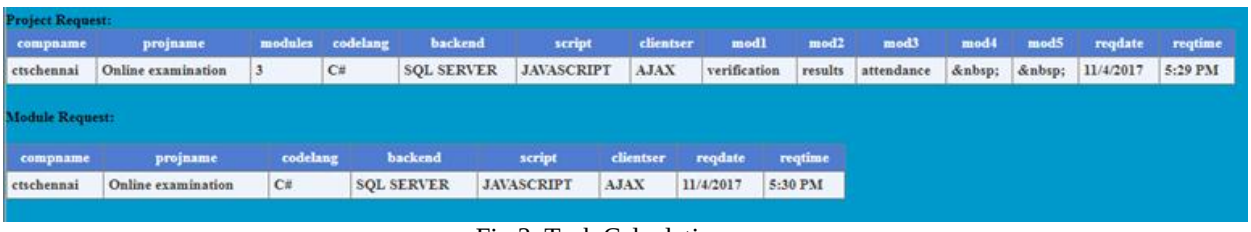
Performance of employees are then calculated by the administrates. This module is mainly designed for team leaders to maintain their own project. In this process the leader should maintain the information like, project's status, project's phase status, developer's report and the work completion chart. The developers under a team leader should report their work status to the team leader frequently. The team leader needs to update the project report.

4. Result and Discussion

The impact of product line engineering participants was inquired as to identify whether they feel current programming product offering designing practice is affecting reuse. Reactions to this inquiry (which are negative) should change the way of code. 87% of the respondents feel that reuse instruction will enable them to take in more about late reuse innovation in making reuse conceivable in their association.

Weightage	Modules	Task	Reusability	Members
0-20 %	1	5	100 %	1
20-40 %	2	10	70 %	2
40-60 %	3	15	50 %	3
60-80 %	4	20	30 %	4
80-100 %	5	25	0 %	5

Fig 2. Reuse Calculation



Project Request:

compname	projname	modules	codelang	backend	script	clientser	mod1	mod2	mod3	mod4	mod5	reqdate	reqtime
ctschennai	Online examination	3	C#	SQL SERVER	JAVASCRIPT	AJAX	verification	results	attendance			11/4/2017	5:29 PM

Module Request:

compname	projname	codelang	backend	script	clientser	reqdate	reqtime
ctschennai	Online examination	C#	SQL SERVER	JAVASCRIPT	AJAX	11/4/2017	5:30 PM

Fig 3. Task Calculation

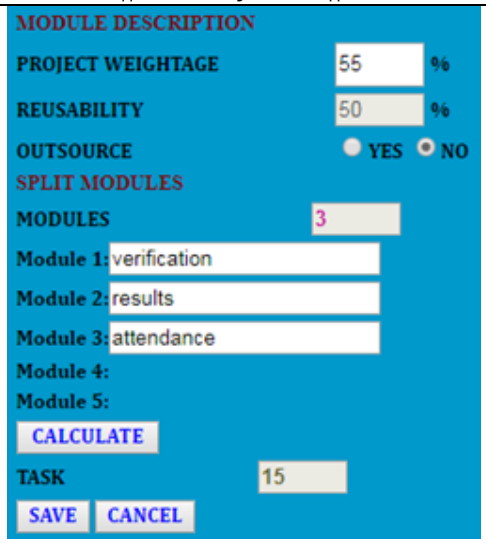


Fig 4. Result for Code Reuse

This module is mainly designed for team leaders to maintain their own project. In this process the leader should maintain the most important information like, project's status, project's phase status, developer's report and the work completion chart. The developers under a team leader should report their work status to the team leader frequently. The team leader needs to update the project report.

5. Conclusion

Each employee of an organization could be benefitted using the enhanced code reusability model, that is registered and allocated based on their performance and the job can be completed in faster time. The code reusability model checks project wise as well as module wise for the code reusability system. The code exchange will be done between one developer to another developer for efficient reuse modelling.

Thus this project can be used for finding the best reuse code to do a particular project in a short span of time and allocating work automatically for employee by finding the right person for the right work by the team leader based on the information that is being updated continuously. This reduces the work of admin and the time of the software developers in a software company.

References

[1]. Yoav Freund, Raj Iyer, Rpbert E. Schapire, Yoram Singer and G. Dietterich, "An Efficient Boosting Algorithm Combines Preferences," Journal of research, 4(2003) 933– 969.

[2]. M. Belkin, p. Niyogi, "Manifold Regularization: A geometric Framework for learning from Labeled and unlabeled Examples,"Journal of research, 7(nov):2399– 2434, 2016.

[3]. Z. Cao and T. Yan Liu, "Learning to Rank: Pairwise to Listwise Approach," In icml '07: proceedings of the 24th International Conference on machine learning, pages 129– 136, 2007.

[4]. BO. Geng, L. Yang, C. Xu, and X-S. Hua, "Ranking Model Adaptation for Domain-Specific Search," In cikm '09: proceeding of the eighteenth acm gathering on information and learning administration, pages 197– 206, 2009.

[5]. "Breaking Down the Barriers to Software Component Technology," by Chris Lamela, intellectmarket, inc

[6]. D'Alessandro, M. Lachini, P.L. Martelli, A" The Generic Reusable Component: An Approach to Reuse Hierarchical Designs," appears in: software reusability,1993.

[7]. Charles W. Krueger Software Reuse "acm computing surveys (CSUR) volume 24, issue 2 (june 1992) pages: 131 - 183.

[8]. Article "Assess Reuse Risks and Costs "www.goldpractice.thedacs.com/practices/arr".

[9]. M. Pat Schuler, "Increasing Productivity Through Total Reuse Management (TRM)," proceedings of technology2001: volume 2, washington dc, december 1991, pp. 294-300.

[10]. Constance Palmer, "A CAMP update," aiaa-89-3144, proceedings of computers in aerospace 7, monterey ca, oct. 3-5, 1989.

[11]. Michael, L. Nelson, Gretchen, L. Gottlich, David, J. Bianco, Sharon, S. Paulson, Robert, L. Binkley, Yvonne, D.Kellogg, Chris, J. Beaumont, Robert, B. Schmunk, Michael, J. Kurtz, Alberto Accomazzi,

and Omar Syed, "The nasa technical report server," internet research: electronic network applications and policy, vol. 5, no. 2, september 1995 , pp. 25-36.

[12]. Pamela Samuelson, "Is Copyright Law Steering the Right Course?," ieee software, september 1988, pp. 78-86.

[13]. Cai, M.R. Lyu, K. Wong, "Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes," in proceedings of the 7th apsec, 2000

[14]. Jihyun Lee, Jinsam Kim, and Gyu-Sang Shin "Facilitating Reuse of Software Components using Repository Technology," Proceedings of the tenth asia-pacific software engineering conference (apsec'03). 1984, pp. 574-588.

[15]. T.J. Biggerstaff and A.J. Perlis, eds.,"Software Reusability: Concepts and Models" acm press, new york, vol. 1, 1989.

[16]. B.Jalender, Dr.A.Govardhan, Dr.P.Premchand, Dr.C.Kiranmai, G.Suresh Reddy, "Drag and Drop: Influences on the Design of Reusable Software Components," International journal on computer science and engineering vol. 02, no. 07, pp. 2386-2393 July 2010.

[17]. B.Jalender, N.Gowtham, K.Praveenkumar, K.Murahari, K.Sampath, "Technical Impediments to Software Reuse," International journal of engineering science and technology (ijest) , vol. 2(11),p. 6136- 6139.Nov 2010.

[18]. W.A. Hegazy, "The Requirements of Testing a Class of Reusable Software Modules," Ph.d. Dissertation, department of computer and information science, the ohio state university, columbus, oh, june 1989.

[19]. B.Jalender, Reddy, P.N. "Design of Reusable Components using Drag and Drop Mechanism," Ieee international conference iri sept. 2006 pages: 345 – 350.

[20]. R.G. Lanergan and C.A. Grasso, "Software Engineering with Reusable Designs and Code," ieee transactions on software engineering, vol. Se-10, no. 5, september 1984, pp. 498-501.