# Enhanced Re-Engineering Mechnanism to Improve the Efficiency of Software Re-Engineering

A. Cathreen Graciamary
Research Scholar, BhararthyiarUniversity, Coimbatore

Dr. Chidambaram
Asst prof, Rajah serofiji College, Thanjavur

***Abstract*—Generally, software re-engineering is economical and perfect way to provide much needed boost to a present software system. Software Re-engineering is like to obtain a fully completed software from existing software with additional features if needed. The overall process of Software re-engineering is to analyze the needed requiements & its contents. It also changes the needed contents or transforms the existing software system for reconstructing a novel software system. The difficult part in re-engineering is to understand the traditional system. Most of the software re-engineering mechanisms are aimed to achieve the common re-engineering objectives and the objectives are: improved software quality, reduced complexity, reduce maintenance cost and increased reliability. As a result, several traditional re-engineering mechanisms fail to verify the performance of individual functionality in existing software. This performance evaluation increases the complexity in re-engineering process. To minimizing the complexities in software re-engineering, this proposed system implements a novel approach named Enhanced Re-engineering mechanism. This enhanced mechanism introduces a new idea, before executing the re- build process the developer verifies the performance of particular function in existing system. After that, the function performance is compared with proposed algorithm. Based on the comparison process only rebuild process should be carried out. Finally this proposed mechanism reduces the complexities in software re-engineering.**

***Keywords*—Software Engineering; Software Re engineering; Software Quality; Restructuring**

## I. Introduction

The reengineering process of the software is modifying and reorganizing the existing system of software for making them maintainable [1]. This is a part of rewriting or restructuring the entire system of legacy without functionality changes. The main intention of the reengineering process is to modify or modernize the previous system in a newer version. The changing process of the business becomes too difficult and complex, or too much costly for the implementation. The legacy maintenance is too costly than the solution of these issues [2]. The benefit of software re-engineering is reducing the risk level at low cost.

The process of re-engineering might face several risks like the software engineering is facing. The identification of the risk is a technique. The identification of the risk is important for the assessment of the risk, management and the analysis of the risk. In proposed system, the risks over the potential are being classified and analyzed [3]. The monitoring technique used to classify and analyzed the risk. It is helping the system of reengineering over the maintenance of ease and profit over cost with the low risk at low cost.

### A. Re-engineering- An Overview

The process of reengineering generally includes the grouping of different process like forward engineering, re-documentation, reverse engineering and translation. The reengineering objective has been explained in the fig.1 that makes easy to realize the existing software or its original functionality, redesigning and enhancement with the latest technology, and added subsystem for gaining more benefit with current technology [4].

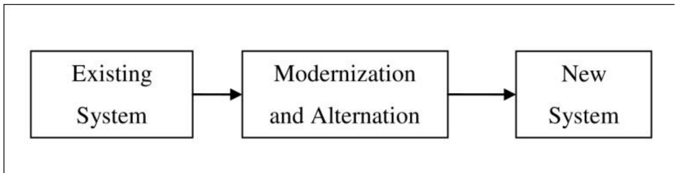


Fig. 1. Re-engineering Process

The latest system which is in use is known as the legacy system. When the system of legacy becomes weak or old for this design, coding, structure, and new module could be added for making the strong legacy system. This process is making the system hard and it is not going to be beneficial over the cost [5].

The software interpretation is simply large and it includes the record of design, other documentation sources and source codes. The reengineering of software is being portioned into two activities sets.

*1)* The first activity set containing the understanding of supporting program, reverse re-engineering, browsing and measurement.

*2)* The second activity set containing the geared evolution of software, like re-modularization, re-structuring, and re-documentation.

### B. Forward Engineering

Forward engineering process is initially begins from requirement specification phase to the software implementation phase. Forward engineering is also known as reclamation or renovation; this process not only extracts design information from traditional software application, but it uses design information to reconstitute or alter the traditional software system in an attempt to get better and overall quality of software application. In most of the cases, re-engineered software application re-executing the existing system functions and also includes a new functions to improve total performance.

### *C. Reverse Engineering*

This process initially begins from the software implementation phase then moving in the direction of developing, designing and requirement specification phase [6]. Thus, procedural design and architectural information of data was discovered from traditional legacy software application.

### *D. Strategic Reengineering*

It refers to the re-development of software system for meeting the criteria of long term plan of the company standard [7]. The lifecycle of the strategic reengineering are involved in the four phases that is planning of Information System (IS), planning reengineering, software reengineering and the building of the reusability framework.

### *E. Definition of software reuse*

The definition of the software reuse is mentioned below [8]:

*1)* "The systematic procedure for creation or development of the software from the stock of blocks, so, the requirement similarity or the architecture in amid of the application could be exploited for achieving the benefit in business performance, productivity and quality".

*2)* "The capability for using the routines of software in novel application".

### *F. Why Reuse Software?*

The reuse of a better or efficient software provide the facility to increase the reliability, quality and productivity and reducing the time of implementation and cost of the software [9]. The repository and reuse process of the software is producing the knowledge base, which is improving the quality of software after each process of reuse, reducing the size of developing work that require for the projects of future and minimizing the novel project risk that based on the knowledge repository.

For that purpose, this proposed system implements a novel approach named enhanced re-engineering mechanism for minimizing the complexities in software re-engineering. In this mechanism, this system compares the performance of functionalities of existing software with the functionalities of new software. Based on the performance evaluation process, re-build process can be carried out.

The rest of this research work is organized as follows. Section 2 states the related work of this proposed work. Section 3 presents the detailed description of this proposed work. Section 4 presents the results and discussion part. Section 5 concludes the future work of this proposed work.

## II. RELATED WORK

Normally Re-engineering process discuss about the "alteration of business process". The changes over the process of business need some novel requirement over the systems [10]. The researcher in [12] has included the changes in process of business, even the present situation for occurring problem also with over time in the development of a system in any organization and that have to be utilized in another organization. There are no more expert in the field of reengineering like design field and in those reengineering, engineers also don't have experience over the research field. Legacy system problem had posed all over the place in world. The Research work [13] is defining about the Legacy systems that significantly oppose evolution and modification of the new changes over the requirement of the business in spite of using technology for designing. The system of legacy is being replaced with the some novel system with the improved and same functionality [11]. The author of the [14] has mentioned in their research work that presenting the reengineering of the iterative over the function of legacy that describe the gradual reengineering process of the components procedural legacy system. The proposed technique of this research work is enabling the system of legacy for making gradual empty over the system of reengineering without freezing the legacy system or duplicate system of legacy. This process is containing the components of legacy system that initially restore the system and then move towards the system of reengineering. At the same time, legacy system could exist in both parts reengineering and restoring of the system. At the last stage of the process, a single system could be existed that is reengineered system. The technique has applied over the original system of reengineered and displayed the ability of it that is support for the gradual reengineering, maintenance of the system during the process of work, the request for the minimum requirement for the freeze maintenance, renewal of the reengineered system for operative environment within the system of legacy and, eliminate every system over the symptoms of aging. The reengineering model of Dual-Spiral for proposed legacy system research work [15] performs a cyclic approach. The reengineering model of the Dual-Spiral workflow require two system together for the work (Each target and legacy requires one system respectively), and functionality move from legacy to target stepwise. During the whole process, the legacy system active functionality is in the pattern of decree-mental, and the novel system of the target over the active functionality is in the pattern of incremental.

For improving the Legacy system quality, the process of software re-engineering must have to enable its new technologies and functions to make sure the efficiency of information management contain the legacy system. The reengineering of Software process involves in the re-documenting and restructuring of legacy system through adding the evolution attempt for easy maintenance and increase the competition and rises over the technology that have forced the organization to adopt new and enhanced approaches for the renovation of the system of legacy within the product, service and process. The method of the evolution over the reengineering of software helps to gain the control over the effective cost, improvement in the quality and the reduction over the risk and time. In few last years some of the framework for reengineering and risk management was developed, but only some of framework was able to recognize the factor of risk in the process of reengineering for successfully creating the risk solution of reengineering by system of software. The research work [18] has developed metrics framework for evaluation of the software system legacy complexity for outsource support. Legacy system framework considers the two dimensions: source code and documentation.

The author in [16] has described the gradual reengineering process over the legacy system for the procedural components. The proposed technique enables the system of legacy to become gradually empty in the system of reengineering; there is no need to do freeze and duplication of legacy system. The research work [19] has focused on the reengineering aspects of technical and the political risk; these are the reason of reengineering effort failure. But, there is more other risks are available in it like risk of reliability, risk of performance, risk of technology, risk of complexity, risk of availability, risk of security, risk of modularity, and risk of usability [17]. Re-engineering risk measurement and identification is necessary competence for successful effort of software re- engineering that provides better strategies for the improvements of quality, control over the cost, and reducing the risk and time for the evolution of legacy system.

## III. Proposed System

### A. Proposed Work Overview

Enhanced re-engineering mechanism is a software re-engineering process that exploits not just a single, but a mixture of abstraction methods and abstraction levels to transform a traditional software system to a new software system. The figure.1 illustrates the overall process of proposed Enhanced Software re-engineering mechanism. Here, proposed system utilizes both forward engineering and reverse engineering techniques. Initially, this proposed mechanism performs feasibility study to check the compatibility of system, and then analyze the components required for that re-engineering process. After collecting the requirements, it move to the next phase. In 2nd phase, mapping of Restructured Software Requirements Specification (SRS) to complete the design document i.e. integration of new Software Requirements Specification to obtain the redesigned document. This re-designed document is an output of the 2nd phase. After completing phase 2, the process is moved to the next phase. In phase 3, the programming part is being customized based on the changes done in the re-design document. It can able to backtrack from this phase to second phase and vice versa if it is required. Then this proposed mechanism perform re-testing and re-integration of various software modules to execute a particular functionality. In this phase, proposed system compares the performance of functionalities of existing software with the functionalities of new software. As per the performance result, better algorithm is replaced with the existing system. This process reduces the complexity and improves the quality of new re-engineering software. After the completion of various modules integration, there is a need to implement the modified system and get the target system which is required by the user.

### B. Working Methodology

This Enhanced re-engineering mechanism consists of five important phases. These are

1) *Feasibility study and requirements*
2) *Restructured System Requirements Specification*
3) *design to code*
4) *Comparison of Existing and proposed Functionalities*
5) *Implementation*

#### 1) Feasibility study and requirements

In this proposed work the initial stage of Enhanced Re-engineering mechanism is the feasibility study and requirements. In this stage, the feasibility study of re-engineering is done i.e. verify the configuration and compatibility of the computer system. After completing the feasibility study, the needs are re-specified based on the user's demand. The SRS contains the entire requirements in a written structure and is an authorized document. To re-specify the system requirements, this system need to map this with the Software Requirements Specification.

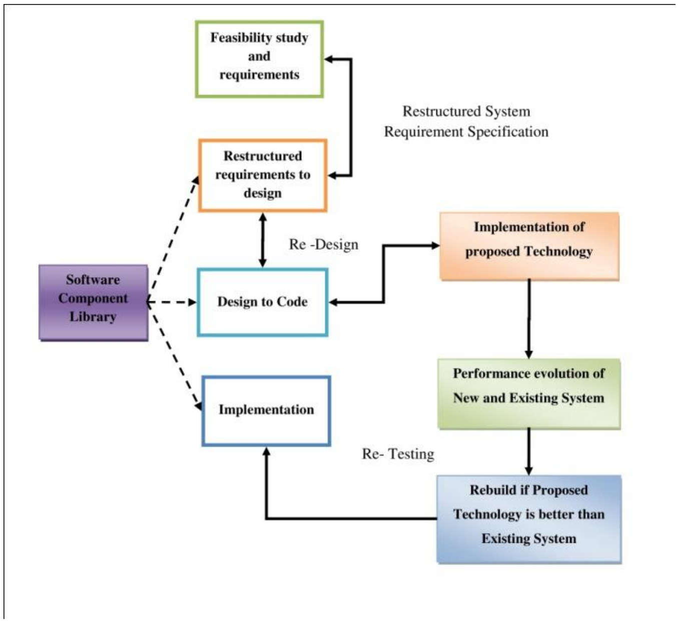

Feasibility study and requirements
Restructured System Requirement Specification
Restructured requirements to design
Re -Design
Implementation of proposed Technology
Software Component Library
Design to Code
Performance evolution of New and Existing System
Implementation
Re- Testing
Rebuild if Proposed Technology is better than Existing System

Fig. 2. System Architecture

#### 2) Restructured System Requirements Specification

This stage describes in detail about the restructured Software Requirements Specification Process. Documentation is a significant attribute in the software development process as it reproduces the entire components of the total re-engineering process and performs as a blue print for the end product. Here, the experts compare the requirements of existing system with new proposed mechanism. SRS is used to integrate the new Software Requirements Specification with existing Software Requirements specification.

#### 3) Design to code

This stage provides the details about design to code process. In this stage, as per the re-design document code has been done by the programmer. Usually, legacy algorithms are implemented in traditional development languages with existing functionalities which are required to be re-written in new functionalities. For example, this system has created one diagnosis system with Naïve Bayes algorithm. But as per the technology requirement the necessary of time and accuracy improvement is very important. So it has planned to re-engineer this application with SVM techniques.

#### 4) Comparison of Existing and proposed Functionalities

This stage provides the details about Re-testing process. For Re-testing, initially takes both existing software application and a new software application. Then, it compares the performance of functionalities of traditional software application with the functionalities of new software application. For performance evaluation, this system utilize the metrics like running time, memory usage and system configuration. Then, the existing function performance is compared with proposed algorithm. Based on the comparison process, rebuild process should be performed. In comparison result, if an algorithm obtains more performance than other algorithms then the better performance algorithm is taken for re-build process.

#### 5) Implementation

This stage is the final stage of Enhanced Re-engineering mechanism. As per the result of previous re-engineering stages, the implementation of a software application can be carried out. In implementation, a specific part can replaced with the good one which fully depends on the previous four stages of this Enhanced Software Re-engineering mechanism.

## IV. Results and Discussion

### A. Experimental setup

In order to analyze the performance of this proposed Re-engineering mechanism a series of experiments on a below mentioned dataset were conducted. In these experiments, this system implemented and evaluated the proposed methods in following configuration: Intel i3(R), CPU G2020, 2GB RAM, processer speed 2.90 GHz, operating system -Windows 7, Front End -JAVA and Back End-MySQL.

### B. Dataset Details

TABLE I. Details of Datasets

| *S.No* | *Dataset Name* | *Attributes* | *Class* |
|---|---|---|---|
| 1. | breast-cancer | 09 | 02 |
| 2. | Diabetes | 08 | 02 |
| 3. | Heart Disease | 13 | 02 |

*For Brest Cancer,*

Attribute 1: Clump_Thickness integer [1,10]
Attribute 2: Cell_Size_Uniformity integer [1,10]
Attribute 3: Cell_Shape_Uniformity integer [1,10]
Attribute 4: Marginal_Adhesion integer [1,10]
Attribute 5: Single_Epi_Cell_Size integer [1,10]
Attribute 6: Bare_Nuclei integer [1,10]
Attribute 7: Bland_Chromatin integer [1,10]
Attribute 8: Normal_Nucleoli integer [1,10]
Attribute 9: Mitoses integer [1,10]
Attribute Class {benign, malignant}

*For Diabetes,*

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

*For Heart disease,*

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. old peak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by fluoroscopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
14. Absence (1) or presence (2) of heart disease

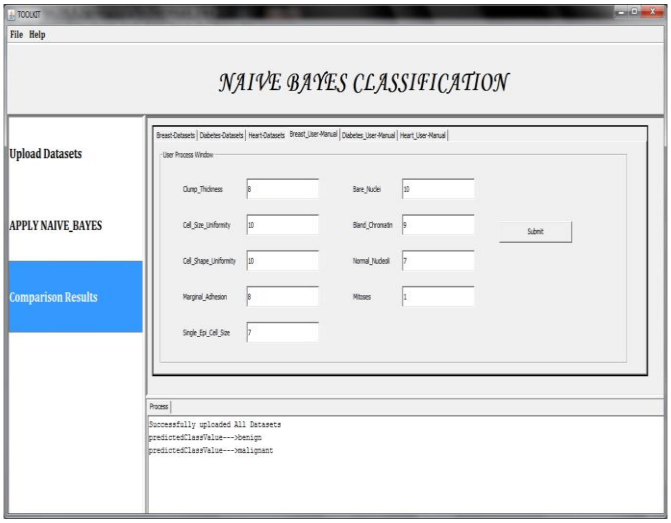
### C. Performance Evaluation

In this proposed system, it has created one medical diagnosis and prediction system for three different diseases like heart disease, breast cancer and diabeties patients. Here it consider two medical diagnosis and predication applications such as

- Naive Bayes Classification System
- SVM Classification System

At first, this system evaluate the performacne of Naive Bayes Classification System. In this, three types of datasets are taken as an input for classification proeess. This classification system works based on the probability distribution of each data set and it efficiently predict the diseases. This predication fully depends on the probability distribution. But computation time of Naive Bayes classification system is too high. Later, it evaluate the performacne of Support Vecor Mechine Classification System and the same three types of disease datasets are taken as an input for classification process. This SVM classification system classifies the data sets based on the training process. This system also effectively predict the datasets,but the Computation time of this algorithm is very low.

Here comparison performance of both Naive Bayes Classification System and SVM Classification System showed. Based on the performance, SVM classification System is better than Naïve Bayes Classification System because the time consumption and prediction accuracy of Support Vector Machine is very High. Inorder to improve the Naive Bayes Classification System, software Re-engineering is required. In this proposed Re-engineering mechanism, there is no need to re-engineering the entire classification system. Instead of Naive Bayes Classification Algortithm, this system replacing the Support Vector Machine. So, the performance of existing classification system is also increased. This proposed re-engineering process reduce the complexities in software re-engineerring process.


TOOLKIT
File Help
NAIVE BAYES CLASSIFICATION
Upload Datasets
APPLY NAIVE_BAYES
Comparison Results
Breast-Datasets | Diabetes-Datasets | Heart-Datasets | Breast_User-Manual | Diabetes_User-Manual | Heart_User-Manual
User Process Window
Submit
Process

Clump_Thickness 8
Cell_Size_Uniformity 10
Cell_Shape_Uniformity 10
Marginal_Adhesion 8
Single_Epi_Cell_Size 7
Bare_Nuclei 10
Bland_Chromatin 9
Normal_Nucleoli 7
Mitoses 1

```
Successfully uploaded All Datasets
predictedClassValue--->benign
predictedClassValue--->malignant
```

Fig. 3. Naive Bayes Classification System

The above figure represents a medical diagnosis and prediction system called Naive Bayes Classification system for predicting the various types of diseases like breast cancer, heart disease and diabetes.

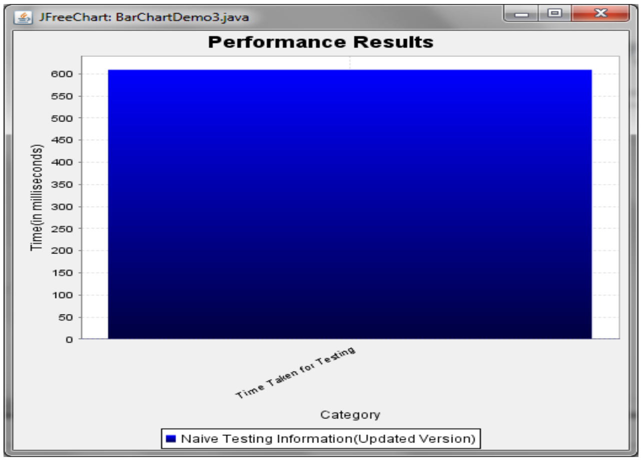
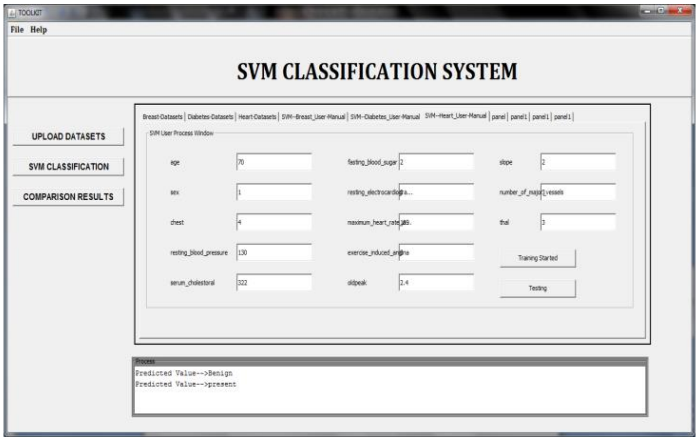


Fig. 4. Performance Evaluation of Naive Bayes

*Classification system*

The above graph represents the performance evaluation for Naive Bayes Classification system. For performance evaluation it consider execution time parameter and the results shows that it takes maximum time for prediction.


TOOLKIT
File Help
SVM CLASSIFICATION SYSTEM
UPLOAD DATASETS
SVM CLASSIFICATION
COMPARISON RESULTS
Breast-Datasets | Diabetes-Datasets | Heart-Datasets | SVM--Breast_User-Manual | SVM-Diabetes_User-Manual | SVM--Heart_User-Manual | panel | panel1 | panel1 | panel1
SVM User Process Window
Training Started
Testing
Process

age 70
sex 1
chest 4
resting_blood_pressure 130
serum_cholestoral 322
fasting_blood_sugar 2
resting_electrocardiogra...
maximum_heart_rate 109.
exercise_induced_an[illegible]na
oldpeak 2.4
slope 2
number_of_majo[illegible] vessels
thal 3

```
Predicted Value-->Benign
Predicted Value-->present
```

Fig. 5. SVM Classification System

The above figure represents a proposed medical diagnosis and prediction system called Support Vector Machine Classification system for predicting the different types of diseases like heart disease, diabetes and breast cancer.

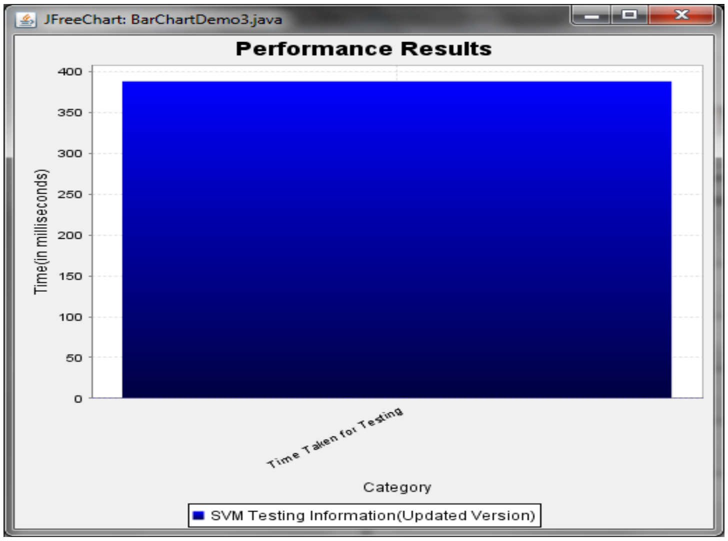
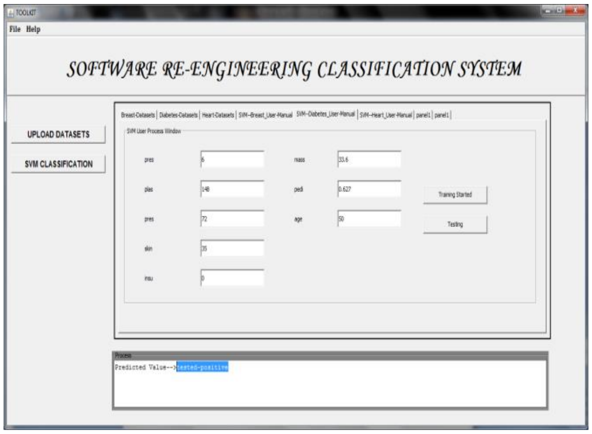


Fig. 6. Performance Evaluation of SVM Classification System

The performance graph represents the time consumption process of SVM classification system. Here, the graph shows the proposed mechanism takes less time for prediction process.


TOOLKIT
File Help
SOFTWARE RE-ENGINEERING CLASSIFICATION SYSTEM
UPLOAD DATASETS
SVM CLASSIFICATION
Breast-Datasets | Diabetes-Datasets | Heart-Datasets | SVM--Breast_User-Manual | SVM--Diabetes_User-Manual | SVM--Heart_User-Manual | panel1 | panel1
SVM User Process Window
Training Started
Testing
Process

pres 6
plas 148
pres 72
skin 35
insu 0
mass 33.6
pedi 0.627
age 50

```
Predicted Value-->tested-positive
```

Fig. 7. Software Re-engineering Classification System

The above figure illustrates the proposed software reengineering classification system window. In software reengineering process, based on the performance evaluation result, the best algorithm is replaced with existing algorithm. Finally, the re-engineering process is successfully completed.

## V. Conclusion

Nowadays, a lot of changes are rapidly involved in hardware and software because of the rapid growth of computer industry. Reengineering of software provides minimized risk level. New software development is really a high risk process because the software development process has some problems like development issues, specification problem, employee problems and cost. Software reengineering process overcomes the abovementioned software development problems because in re-engineering process some of the parts should be changed. To improve the performance of software re-engineering, our proposed system implements an enhanced reengineering mechanism. This mechanism is proposed to reduce the cost and time. Finally, our proposed system increases reliability of software and improve the quality of service with minimum development efforts.

## References

[1] Francisca O. Oladipo, Jude O. Raiyetumbi," Re-Engineering Legacy Data Migration Methodologies In Critical Sensitive Systems", Vol 6, No . 11, Journal of Global Research in Computer Science, November 2015.

[2] Manojkumar. P.K , " Classification And Analysis Of Risks In Reengineering Projects " , Vol . 2 , Nehru E- Journal - A Journal for Arts, Science and Humanitis , June –Dec 2014

[3] Hausi A. M¨ uller , Jens H. Jahnke , Dennis B. Smith , Margaret-Anne Storey , Scott R. Tilley , Kenny Wong , "Reverse Engineering : A Roadmap " , ICSE , 2000.

[4] Frank Weil, Ph.D , UniqueSoft, LLC , " Legacy Software Reengineering " , UniqueSoft LLC , 2015.

[5] Dr. Larisa Melikhova, Albert Elcock, Andrey A. Dovzhikov, Georgii Bulatov, Dr. Dmitry 0. Vavilov, "Reengineering for System Requirements Reuse: Methodology and Use-Case", 2007, IEEE.

[6] Krishan Kumar , Prabhpreet Kaur , " A Generalized Process of Reverse Engineering in Software Protection & Security " Vol. 4 , International Journal of Computer Science and Mobile Computing , May 2015

[7] T.G.J. Schepers , M.E. Iacob , P.A.T. Van Eck, "A lifecycle approach to SOA governance "https://www.researchgate.net/publication/234810165 , January 2008.

[8] M.H.Arifa Banu, N.Mohamed Thoufeeque, K.Archana, "Study of Software Reusability in Software Components", Vol 5 No 3, International Journal of Engineering and Technology, Jun-Jul 2013.

[9] Mohammed-V Agdal Univ, Ecole Mohammadia d'Ingénieurs (EMI), Siweb Research Team , "All About Software Reusability: A Systematic Literature Review " , Vol.76. No.1 , Journal of Theoretical and Applied Information Technology , 10th June 2015

[10] Ahmed Saleem Abbas, W. Jeberson & V. V. Klinsega, "Implementation of Fusion Model to Re-Engineer Legacy Software", International Journal of Computer Science and Engineering (IJCSE), 2013.

[11] Shekhar Singh, Significant role of COTS to design Software Reengineering Patterns, International Conference on Software Engineering and Applications(ICSEA),2009.

[12] Daniel Gjörwell, Staffan Haglund, Daniel Sandell," Reengineering And Reengineering Patterns", The Department for Computer Science and Engineering Mälardalens Högskola, 2002-02-24.

[13] Key Management Group Inc., " Legacy System Data Conversion And Migration " , Legacy System Data Conversion and Migration.

[14] Napas Methakullawat and Yachai Limpiyakorn, " Reengineering Legacy Code with Model Transformation " , Vol.8, No.3 , International Journal of Software Engineering and Its Applications , 2014

[15] Xiaohu Yang et al, "A Dual-Spiral Reengineering Model for Legacy System", TENCON 2005 - 2005 IEEE Region 10 Conference ISBN: 0780393112 Year: 2005 Pages: 1-5 Provider: IEEE Publisher: IEEE.

[16] Alessandro Bianchi, Danilo Caivano,Vittorio Marengo, Giuseppe Visaggio," Iterative Reengineering of Legacy Functions", 17th IEEE International Conference on Software Maintenance (ICSM'01),Florence, Italy, ISBN: 0-7695-1189-9,November 07-November 09.

[17] Er. Anand Rajavat , Dr. (Mrs.) Vrinda Tokekar , "Techrisk –A Decisional Framework To Measure Technical Dimensions Of Legacy Application For Rejuvenation Through Reengineering " , Vol.2, No.3, International Journal of Software Engineering & Applications (IJSEA), July 2011

[18] Oliver Hummel , Stefan Burger , "A Pragmatic Means for Measuring the Complexity of Source Code Ensembles " , IEEE. , 2013

[19] Basem Y. Alkazemi, "A Framework To Assess Legacy Software Systems", Vol. 9, No. 1, Journal Of Software, January 2014.