



A PERSPECTIVE SOURCE CODE ON SOFTWARE REENGINEERING APPLICATIONS

Dr.S.Manimekalai

Assistant Professor in Computer Science,
Theivanai Ammal College for Women (Autonomous) Villupuram,
Tamil Nadu, India- 605602

Abstract: Software Reengineering can be defined as the examination, analysis, and alteration of any existing software system to reconstitute it in a new form, and the subsequent implementation of the new form [12]. The current trends in software reengineering are “innovation for the third platform”. Legacy is old. Developing on the third platform will accelerate in 2014. For developing a third platform require high skills of security, Social, Cloud, Mobile, Big Data/Analytics, IOT, HCI, Gamification, and Cognitive Computing. Daily as human beings, to use variable naming theories that they neither aware of nor understand, but by the end of the day our communication either forwards or backwards continue without understanding progress. This proposal is to apply a comprehensive Framework for extracting variable names from the source code, clean them through. This would be satisfied to show merely that the conditions within the framework can be used to help us achieve a mechanism for understanding at least some essential properties of the variable names within the system and how they are evolving over a given period of time. The question is to address is whether it is possible to have a framework for arriving at a critical decision as to how to use variable names in software reengineering. These modes of determination need play an important part in the explanation of mutual understanding, of communication.

Keywords: Source reengineering, ASP.net, Line of Codes, Three tier architecture, Variable Naming.

1. INTRODUCTION

This research paper is a real time study in HealthCare Hospital and it's about the reengineering of existing system which has many named variable and to make the named variable meaningful in a particular environment the variable just have to be named domain specifically. Only the source code holds the key to successful application software re-engineering with regards to cost, time to market, flexibility, and portability. This research paper proposes a Framework for extracting these variable names, from available source code, and cleaning them and then storing them in a database. This would satisfy to show merely that the conditions within the framework can be used to help us achieve a mechanism for understanding at least some essential properties of the variable names within the system and how they are evolving over a given period of time.

To develop the process, designing phase is necessary and that design is created by the designer involved in that phase. The design is developed based on their idea and known technology. But it is necessary to convert the design according to the programming language used in the implementation phase. This conversion may leads to time consuming, when the programmer involves in this conversion since it has lot of work from identifying the tool to choosing the appropriate tool from the programming language implemented [7].

2. RELATED WORKS

The real time study is derived from the following references

[1] Has discussed in Many modern software design methods have been developed to improve the reusability and maintainability of software and to reduce the time required

for the maintenance and development operations, but many companies have old or legacy software systems, these companies spend a lot of money to maintain their old systems. These systems cannot be replaced by new systems because they contain implicit information and decisions cannot be lost. For these purposes, re-engineering becomes a useful tool to convert old, obsolete systems to more efficient, streamlined systems.

[2] Models are new in the field of Reengineering of object oriented software systems. The future work is to test these models for suitability to fit on the basis of analysis of current and past data. These models can be accepted as it is or improved or rejected. Once fit and fine these models will help in reengineering the legacy software with optimal cost.

[3] In this paper a systematic review of the work in the area of software reengineering is conducted and review results are presented. The review results shows that majority of the methodologies are process based, where a process is discussed to reengineer a legacy system, no validation or automatic tool is provided and the target legacy system is object oriented and orientation of the system are towards component based system. These results points out that most of the object oriented systems are becomes legacy now and software developers are more interested in migrating object oriented systems into component based systems.

[4] This paper explains about the support of software metrics in Software reengineering. It clearly explains about the effect of metrics in the current system and how the metrics can be used in reengineering on the perspective of technical quality and business value. The collection of metrics for the new system can continue throughout development. To demonstrate, a metric framework has been used for reengineering process. This framework is used in different phases of the Rainfall model to make reengineering process easy, economical and efficient. Case study of

software has been undertaken to validate this metric framework.

[5] This paper deals with the three different modules as follows

- (I) Extraction of Models from Natural Language Requirement.
- (II) Re-Engineering and Extraction of Software Quality Metrics.
- (III) Identification of validated Secure Design for quality assurance.

The three Modules are integrated to form a framework to fully automate thereby eliminating the manual effort required thus making the system effective and efficient for the reengineering of the software. This framework applies metrics at both the levels design as well as code level by considering coupling of classes and objects. It generates spider graph to give the clear idea about the class coupling. It can identify high coupled class which may be vulnerable to attack in future.

3. CURRENT SYSTEM ANALYSIS

System analysis is the study of a system under consideration (which may be real or imagined). The main purpose is to understand and documentation of the essential characteristics of the system being studied. Its eventual goal is to come up with a specification of the system under study.

A. Functions in system

This research paper deals with the forties Health Care Hospital. The health care domain required the following function under its shadow

- Patient record
- Lab records
- Test records done to the patient
- Materials management
- Human Resources

In the above functions the first three is directly linked with the Operation of the healthcare domain and the rest two will be related to the supporting functions of the operation.

B. Existing Software



Figure 1 Existing System Framework

The Current software system has a lab master. The lab master consists of different lab such as hematology, blood glucose, liver, pathology, biometrics etc. The different forms have to be filled with different data's. The data updating will be in such a way that it can be filled manually or a pivot point or a drop box. After the data is updated in the appropriate fields the data access layer will be introduced to save data.

C. Program Creation in ASP.Net

From the software aspect for developing software they will find the requirement from the customer and convert the requirement into the software language and develop a work breakdown structure and provide the same to all programmers for developing the program as per the requirement of the customer. The ASP.Net programming is having the following flow of work process.

The form creation is the ultimate requirement satisfaction of the research paper. The form is the presentation layer or application layer of the software system. It is how the user is interacting with the database to fetch or save data. The form is created with the help of ASPX coding. Databases are the ultimate source to store and maintain the data. It is also easy to retrieve the data from these databases than other source. This Re-Engineering technique also simplifies the process of the database operations using some implementations. Only the requirements are produced, the query will automatically generate and the result is produced [10].

The preparation of the coding sheet needs an in depth knowledge of coding and programming. The coding sheet provides the flow of function in the application layer. It defines the fields and variable required in the application layer.

Creating a basic table involves naming the table and defining its columns and each column's data type. The SQL CREATE TABLE statement is used to create a new table.

A Stored Procedure (also termed proc, storp, sproc, StoPro, StoredProc, StorePrc, sp or SP) is subroutine available to applications that access a database management (RDBMS), such procedures are stored in the database data dictionary.



Figure 2 Working of ASP.Net

When they work with this data in ASP.NET, they will benefit by using common software patterns. One of these patterns is to separate the data-access code from the business-logic code that governs access to the data or that provides other business rules. In this pattern, these two layers are separate from the presentation layer, which consists of the pages that the Web site user accesses to view or change data [10].

a. Problem Faced in Existing System

- For each and every test the programmer needs to create a new ASP form and the subsequent Stored Procedure, Table Creation, Coding Sheet and Data Layer etc.

- In this format the entry of fields and variable were done manually by the end user who may lead to mistakes in the entries.
- The Presentation layer form was created from a single design page which has more than 1500 no of codes.
- While updating a page many updating comments are used to save, cancel update edit buttons are used.
- The rectification of the error once occurred will be rectified with a tedious effort in updating the Coding sheet, table creation, Stored Procedure, Data and Business Layer
- The Effort of the programmer was more in constructing the page and in rectifying the error

D. Significant of the study

This research paper will clearly explain about the existence software which has a separate form and application for each and every named variables is reengineered to a just by entering the variables in the presentation layer the software will capture the named variables and creates the forms in the application layer.

Only the source code holds the key to successful application software re-engineering with regards to cost, time to market, flexibility, and portability. This research paper proposes a Framework for extracting these variable names, from available source code, and cleaning them and then storing them in a database. The framework allows domain experts, assisted by the framework, to then conclude whether the names used are leading us towards or away from the actual domain or even that the words used are giving us enough knowledge and control of the subject under reengineering through a variable name scoring process.

Propose to apply a comprehensive Framework for extracting variable names from the source code, clean them through. That would be satisfied to show merely that the conditions within the framework can be used to help us achieve a mechanism for understanding at least some essential properties of the variable names within the system and how they are evolving over a given period of time.

The question to address is whether it is possible to have a framework for arriving at a critical decision as how to use variable names in software reengineering. These modes of determination need play an important part in the explanation of mutual understanding, of communication. Or whether, as in the case of a name, it is sufficient for two people to understand one another that they know both associate with a predicate of the same reference. This is being achieved would mean it would be very possible for any person to read, understand source code that one has not written without problems and in the least possible timeframe.

E. Data Collection:

The Research paper is based on the exploratory research model which clearly explains a survey of related literature; experience survey such as focus Group and depth Interview; and Analysis of ‘insight-stimulating’ instances such as case studies. As a part of developing a program in an organization quantifying the work done will always exist in these cases the LOC is calculated at each layer through a separate framework developed by the focus group. The LOC Data Collection of different layers is as follows.

Table 1 : Line Of Codes in Different Layers

Descript ion	ASP Page	Codi ng Sheet	Busin ess Layer	Data Access Layer	Table Creati on	Stored Proce dure
BLOOD GULCOSE	1587	1482	1458	1458	46	45
HEMATO LOGY	1592	1422	1483	1486	42	42
LIVER	1590	1452	1471	1472	44	44
Old Format Total	4769	4356	4412	4416	132	131

4. RESULTS AND DISCUSSION

There are three different test available in this old format such as hematology, blood glucose and liver test. Initially the ASP form was created for each and every individual test and the patient name is linked with the test and the patient history was entered in the form.

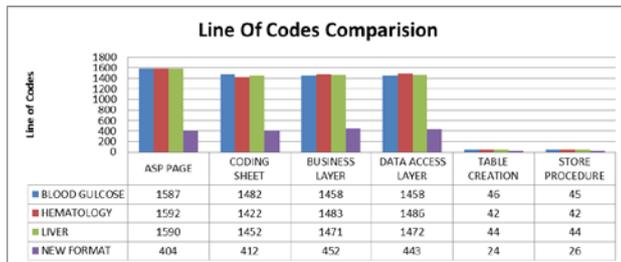
I. Significant of the New Format

- The ASP page is created for the clubbed forms. The Lab Master, Test master, test report entry, test result entries are clubbed to present as a separate module.
- The Entry of variables and Fields are replaced by the drop down box and button system which can avoid the mistakes at the time of entries.
- The presentation layer form is created with n number of master, n number of module, with dynamic updating of the records.
- The coding is created in such a way that only adds and save options is used in updating the form.
- The rectification of error is easy the sheets are linked in such a way that coding sheet, table creation, Stored Procedure, Business and Data layer are interlinked.
- The Efforts of the programmer are less only they needs to add the Fix the error.

II. Comparison of the New System:

The line of code analysis existed in the old format and the new formats are as follows and the same is tabulated in table 1.

- Blood Glucose which has line of codes Asp Page-1587,Coding Sheet-1482,Business Layer-1458,Data Access Layer-1458,Table Creation-46,Stored Procedure-45
- Hematology which has line of codes Asp Page-1592,Coding Sheet-1422,Business Layer-1483,Data Access Layer-1486,Table Creation-42,Stored Procedure-42
- Liver which has line of codes Asp Page-1590,Coding Sheet-1452,Business Layer-1471,Data Access Layer-1472,Table Creation-44,Stored Procedure-44
- New Format Asp Page-404,Coding Sheet-412,Business Layer-452,Data Access Layer-443,Table Creation-24,Stored Procedure-26



The Total Comparison of total No of LOC for the Old and New Format is tabulated below

Table 2: Line of Codes Old and New Format

Descripti on	ASP Page	Coding Sheet	Busines s Layer	Data Access Layer	Table Creati on	Stored Proce dure
BLOOD GULCOSE	1587	1482	1458	1458	46	45
HEMATOL OGY	1592	1422	1483	1486	42	42
LIVER	1590	1452	1471	1472	44	44
OLD FORMAT TOTAL	4769	4356	4412	4416	132	131
NEW FORMAT	404	412	452	443	24	26
% Reduction	92%	91%	90%	90%	82%	80%

5. CONCLUSION AND FUTURE WORK

Throughout this research paper, it has explored and closely followed the main reason why and how the application software ages overtime. The Approach is a gently skeptical approach, accepting and building on existing concepts and principles have to build a new more robust framework for extracting variable names from source code. The following contributions and conclusions may therefore be put forward

- To developed a complete framework to explore source code files for variable names and their attributes;
- To proposed several steps to be followed from extraction, cleaning, matching and storing the variable name;
- As part of our Framework, to explore variable names from source code files, and also gone through several Software engineering principles.

Future Work

In our approach, it’s clearly ignoring the effect of Architecture of the System as an Enterprise Solution. The assumption that Architecture are not altered; only additional classes and variable names were added, subtracted or

updated due to new business rules or change thereof. But no major rework has been conducted on the in as many as Architecture.

However, where the Architecture have been altered, then a framework for understanding the Grammar and Architecture and a comprehensive follow-up of the same over a period of time has to be taken into consideration and mitigate all of them to come up with a comprehensive and wholesome Framework.

REFERENCES

- [1]. Dr. Ashok Kumar, Bakhshish Singh Gill, “Maintenance vs. Reengineering Software Systems”, Global Journal of Computer Science and Technology, Volume 11 Issue 23 Version 1.0 December 2011.
- [2]. Ahmed Saleem Abbas, W. Jeberson, V.V. Klinsega, “The Need of Re-Engineering in Software Engineering”, International Journal of Engineering and Technology, Volume 2 No. 2, February, 2012.
- [3]. AmanJatain and DeeptiGaur, “Reengineering Techniques for Object Oriented Legacy Systems”, International Journal of Software Engineering and Its Applications, Vol. 9, No. 1, 2015
- [4]. Dr. SumeshSood, “A Framework for Software Reengineering Using Set of Software Metrics”, International Journal of Computer Science Engineering, Vol. 3 No.02 Mar 2014.
- [5]. Mr. Thakore Devendra Singh,, “Design And Develop An Environment Analyse Object-Oriented Software And Quality Assurance”,Shri Jagdish Prasad Jhabarmal Tibrewala University, Vidyanagari, Jhunjhunu, Rajasthan, 2013.
- [6]. Alshammari Bandar, Fidge C. J., and Corney D., “Security metrics for object oriented class designs”, in Proceedings of the Ninth International Conference on Quality Software (QSIC 2009), (Jeju, Korea).
- [7]. Manimekalai S, Kirubakaran E, “Re-Engineering for GUI Conversion from Markup Language to Programming Language”, International Journal of Advanced Research in Computer Science,2010.
- [8]. Amjan Shaik, C. R. K. Reddy, Bala Manda, Prakashini. C, Deepthi. K, “An Empirical Validation of Object Oriented Design Metrics in Object Oriented Systems” Journal of Emerging Trends in Engineering and Applied Sciences 2008.
- [9]. Anandha Mala G.S., Jayaradika J., and Uma G. V, “Restructuring Natural Language Text to Elicit Software Requirements”, in proceeding of the International Conference on Cognition and Recognition 2006.
- [10].Manimekalai S, Kirubakaran E, “Wrapper Based Design Conversion to Database in Reengineering”, International Journal of Advanced Research in Computer Science, 2011.
- [11].Arun Kumar S., Arun Kumar T.and Swarnalatha P. (2010), “Significance of Software Metrics to Quantify Design and Code Quality”.
- [12].[https://msdn.microsoft.com/en-us/library/bb470374\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/bb470374(v=vs.100).aspx)
- [13].M Afshar Aalam, Tendai Padenga, “Application Software Reengineering”,Pearson Education India, 2010 Edition.