

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

EFFORT ESTIMATION FOR SOFTWARE PACKAGE PROMOTION

M.Keerthiga¹, B.Thirunavukarasu², R.Prasanneswar³

B.E Computer Science & Engineering^{1,2,3}

SNS College Of Technology, India^{1,2,3}

keerthiga.mn@gmail.com¹, droptothiru@yahoo.com², rvspeswar@gmail.com³

ABSTRACT

In any engineering field the data associated with knowledge is important one for taking decisions for solving problems in the current system development. The specification mining can give support for analyzing collected data to help the project management team to fulfill their responsibilities. In this paper 'Package Specification Mining' is designed by using packages re-usability quality factor. It supports to give effort required for modifying the package to be reusable package for using those packages in new software development. This methodology may reduce the risks in various domains of software engineering.

Keywords– Specification Mining, Re-usability, Effort Estimation, Coupling, Project Management

1. INTRODUCTION

Software Engineering, it focuses on developing software systems to fulfill customers' business problem in a simplest way. Hence project development activities are organized in a proper manner to track the progress of any software project. Project Management Team involved producing project plan by assessing the available resources in industry. The project plan should be effective one for satisfying both development and customer's requirements. To make such characteristic in development domain the effort estimation should be done very prior. For doing effort estimation the available resources and their level of availability should be accounted. In software domain some software packages may not be categorized as reusable package. But through experience the developer can prefer such software packages for further use in the new project construction. As well as to make that package as usable one how much stuff or effort needed to be calculated. To find that package modification effort, in this paper we proposed a technique based on existing available software packages.

1.1 Project Management

Developing a project is a critical task due to satisfy the needs of both clients and software firm. The clients may not be technical person. Hence it must be very difficult to understand clients' requirements to startup the development activities. The project management team comes to handle various activities in development side. Those are

- Effort Estimation
- Requirements Management
- Resource Management
- Risk Management
- Quality Management

The above management activities should be done properly by the project management team to deliver the software with specified quality attributes. Here the ROI (Return on Investment) will be evaluated after the business deal. To measure ROI and increase the ROI the 'Effort Estimation' will be done as per the available resources in industry side. This activity is the initial activity of developing any system in engineering fields. It is discussed in the following sections.

1.2 Types of Effort Estimation

The two types of effort estimation techniques such as 'Top-Down' and 'Bottom-up' are used in software industry to propose a project plan. The first technique calculates the effort for total system and then subsystems can get the effort for their construction. But in 'Bottom-up' approach first the effort for subsystems are identified and then through that the effort for total systems can be found. The COCOMO model will be used in software industry for getting accurate result of effort estimation for planning the project development. During effort estimation the cost, resources, manpower, machines, time and other similarities between the projects are the parameters of the estimation.

Modification Effort

During development the developer may have an idea to acquire some software packages to develop his module. It may not be proposed in the project plan. If it is revealed by the developer that frequently he used a particular software package with little modification then calculated effort is wasted for that particular module. By modifying the existing software packages as reusable one that may help in upcoming project plan. So modification effort has to be calculated for making a non-reusable package as reusable one. Through that activity they can take a decision on whether to go for modification or purchasing suitable components from the market.

1.3 Introduction to Specification Mining

In software engineering ‘Specification Mining used for taking decisions for different activities involved in software development life cycle. It can be used software testing phase also for checking the characteristics made in the software. It is not a data mining approach. A small collection of data sets from the software development are used to construct the specification mining. It could simplify the task of machine learning in engineering platform.

1.4 Package Specification Mining

In this paper we proposed the technique to have package details in Specification Mining for further usage in development. In knowledge management system all the details of developed software can be stored. But analyzing unwanted details takes much time and it wastes effort for doing that activity. So by taking particular details for choosing decisions will be a best method. Only essential features of software packages are taken to record in the specification mining.

1.5 Role of Software Testing

It is mandatory one in any engineering field for verifying and validating the requirements. In software development many types of testing are proposed to check different requirements before deploying the system in customer business environment. The specification mining can be used to record proper test cases for testing any new software system. And that can be used for generating test cases automatically.

2. EXISTING TECHNIQUE

The re-usability factor of software java packages is considered to estimate required effort for modifying any existing software packages. During integration testing side itself the re-usability factor will be measured and it should be documented for further usage in any software platform. The following figure shows the activity flows in the proposed system.

The first module ‘Collecting Packages Data’ is working on the given software for collecting the details of used packages and components in the system. The next step is focused the job on measuring the required package metrics for defining re-usability of those packages. And then the values such as coupling and cohesion are analyzed to declare the re-usability level of software packages. After that the required effort modification could be found as per the instability value of those software packages.

2.1 Analyzing Metrics of Re-usability

To measure the re-usability of software packages the ‘Coupling’ metrics is required.

- Import Coupling (IC)
- Export Coupling (EC)

In this paper we take above types of Coupling to measure the ‘Instability’ factor. The formula required for deriving ‘Instability’ is given below.

$$\text{Instability} = (\text{IC} + \text{EC}) / \text{EC}$$

The value of instability of any software package will be in the range of 0 to 1.

If Instability = 1 then that package is fully complex due to very high in and out communication such as data passing or control passing between modules. So that package is very difficult to redesign to make it as reusable package for deploying in any project construction.

If Instability value lies in between 0 to 0.5 may be considered for further modification.

2.2 Structure of Package Specification Mining

In this proposed methodology we take specification mining as ‘Package Specification Mining’ for identifying modification effort. And here we preferred only few package details to be maintained in specification mining. The following table shows the data about package to be maintained in the specification mining.

Table I Package Specification Mining

S.No.	Package Attributes
1	Package Name
2	Amount of Services
3	Complexity
4	Instability value
5	Reusable or Not
6	Who Developed it?
7	Memory Size

Package Name: Each package has technical name to understand its objective in the developed system. So the technical team put perfect technical word associated to the system functionality.

Amount of Services: This property describes the number of services grouped in a single package or component. It is also essential to know the usefulness of it in the used systems.

Complexity: The complexity of source code has to be measured for future modification. If it is in high complexity then huge amount of resources may be required for modification. So it may cause some problems in resource utilization.

Instability: This field’s value can help the proposed methodology for checking its stability. It has a reason that if it is still instable after some modification then it may not be useful. It may affect system performance when it is not stable.

Reusable or Not: It indicates the level of the package reusable. It describes that communication among software package or components in the system.

Who Developed? : This field has the name of the developer who developed that particular module or package in the software system. Keeping the technical developer detail will help to appoint him for future modification, because already he had technical experience in that proposed module. So appointing the same expert can reduce some effort for doing modification.

Memory Size: It is another constraint in the configuration of the system. After modification if the takes huge memory space from its original memory space it may disturb the system performance while it is under usage.

The above parameters are used to take a decision for satisfying the objectives of the project management team. The activities of the proposed methodology described in the following section.

3. PROPOSED METHODOLOGY

The re-usability factor of software java packages is considered to estimate required effort for modifying any existing software packages. During integration testing side itself the re-usability factor will be measured and it should be documented for further usage in any software platform. The following figure shows the activity flows in the proposed system.

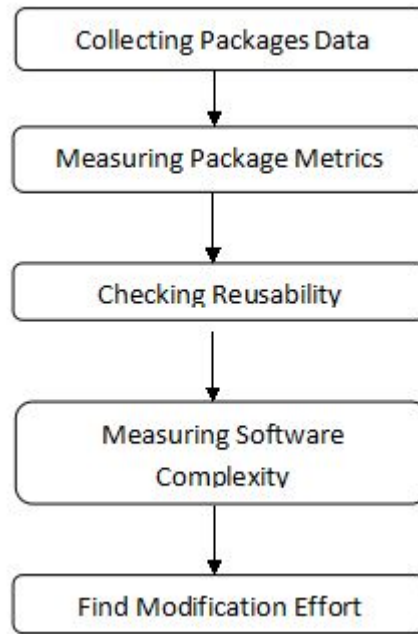


Fig. 1. Flow of Proposed Methodology

The first module ‘Collecting Packages Data’ is working on the given software for collecting the details of used packages and components in the system. The next step is focused the job on measuring the required package metrics for defining re-usability of those packages. And then the values such as coupling and cohesion are analyzed to declare the re-usability level of software packages. After checking re-usability the complexity of the software product can be measured so that the possibility of developing the reusable software can be identified. After that the required effort modification could be found as per the instability value of those software packages.

3.1. Steps in Proposed System

1. Scan Source code of the system
2. Finding Packages and Packages elements
3. Measure Package Metrics (Package Specification)
Coupling and Types of Coupling
Stability through Instability
4. If Coupling>20 then ‘ High Coupling ’
Else
‘ Low Coupling ’
5. Identify types of Coupling
Import Coupling (IC) & Export Coupling (EC)
6. Compute Instability (I)
7. Measure the complexity
8. Record All Metrics values

9. Apply Specification Mining

$$I = IC / (IC+EC)$$

If I==0

Then 'Package is Stable '

If I==1

Then 'Package is not Stable & High Coupling '

If I<1 && I>0

Then 'Some Effort Needed to become Stable'

3.2 Cyclomatic Complexity

The general formula for calculating the Cyclomatic Complexity is as follows:

$$V(G) = E - N + 2 * P$$

Where V(G) is the Cyclomatic Complexity,

E is the number of edges,

N is the number of nodes and

P is the number of modules in the software product.

The above script describes the activities to be done in the proposed system.

4. RESULTS & DISCUSSIONS

The approach in this proposed methodology is collecting data associated with software packages implemented in the developed software project. Here we have taken some sample data set for describing the function and output of the system.

A developed software system has 4 software packages. The package details are collected from the proposed system as per the specification mining.

Property	Package1	Packag2	Package3	Package4
Classes	20	10	15	6
Amount of Services	5	2	4	1
Instability	0.6	0.8	0.4	0

Designing specification mining is a tough task in software engineering field, because different details may be required in different software development phase. So here we plan to have a decision support system for taking decisions on purchasing reusable components from the market. If that purchased reusable component is not supporting for further modification then that purchase may be a useless one. Even it is supporting for enhancement appointing or identifying such experts will be a difficult one due to code study or complexity of its code.

5. MERITS & DEMERITS

Here we proposed package based specification mining for taking decisions for designing reusable packages which need low effort. So, resource utilization can be improved by admitting idle resources to modify the existing eligible package to be modified as reusable one. Through that project plan can be prepared very easily. Here specification mining contains the details of developers who designed the existing software packages. Hence that helps the project management team to allot such human resource for improving the features of those software package elements. From the above discussion the advantages are listed below.

- 1) Effective Resource Management
- 2) Risks can be identified
- 3) Quality of Software can be improved by applying low effort
- 4) People Management will be efficient.
- 5) Customer could be satisfied with good quality of work.

This proposed methodology has few drawbacks. Those are listed below.

- * Collecting details takes huge time
- * Additional system memory space used for keeping essential package attributes values
- * Applying suitable algorithm for getting expected result from package specification mining.

6. CONCLUSION

Measuring the required effort for proposing a modification in the existing package will lot of support to project management and to the development team. But choosing suitable parameters for taking decisions is important one. So the specification mining can help the industry to get an idea very quickly for a decision. This proposed system works based on the re-usability level of the software package. This is very useful parameter for improving the quality in all aspects in software development.

7. REFERENCES

- [1] Claire Le Goues and Westley Weimer "Measuring Code Quality to Improve Specification Mining", *IEEE Transactions on Software Engineering*, vol. 38, no. 1, January/February 2012.
- [2] Jyoti Mahajan, Devanand and Kashyap Dhruve, 'REBEE – Reusability Based Effort Estimation Technique using Dynamic Neural Networks', *Global Journal of Computer Science and Technology Volume 11 Issue 7 Version 1.0 May 2011*.
- [3] Vinit Lohan, 'Component based effort estimation During software development: problematic view', *IJCSMS International Journal of Computer Science and Management Studies*, Vol. 11, Issue 03, Oct 2011.
- [4] Rajesh Bhatia, Mayank Dave and R. C. Joshi, "Retrieval of Most Relevant Reusable Component Using Genetic Algorithms", *Software Engineering Research and Practice -SERP*, pp.151- 155, 2006.
- [5] Caldiera, G. and Basili, V.R, "Identifying and qualifying reusable software components", Feb, *IEEE Computer Society*, 1991.
- [6] Parvinder Singh Sandhu, Janpreet Singh and Hardeep Singh, "Approaches for Categorization of Reusable Software Components", *Journal of Computer Science* pp. 266-273, 2007.
- [7] Haeng - Kon Kim, "Automatic Identification of Potential Reusable Mobile Components International Journal of Computer Science", Nov 2008.
- [8] Valentin Dallmeier, Nikolai Knopp, Christoph Mallon, Gordon Fraser, "Automatically Generating Test Cases for Specification Mining", *IEEE Transactions on Software Engineering*, Vol. 38, No. 2, March/April 2012

- [9] Nancy Bazilchuk and Parastoo Mohagheghi (2005), "The advantages of Reused Software Components", *IEEE Transactions on Software Engineering*, Vol. 23, No. 9, pp. 556-565.
- [10] R.Kamalraj, B.G. Geetha and V. Shyamaladevi, "Checking Reusability of Software Packages using Integration Testing", *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, May 2009.
- [11] R.Kamalraj, Dr.A.Rajivkannan and P.Ranjani, "Stability based Component Clustering for Designing Software Reuse Repository", *IJCA*, Vol. 27, No. 3, Aug 2011.