ABSTRACT

This research has proposed a model called “ADAPTIVE REUSABILITY RISK ANALYSIS (ARRA) MODEL” that facilitates analysis of reusability process in Object Oriented Programs. This model has been tested and validated for implementation. The scope of this model is limited to Software Engineering.

Software Engineering is a strategy that ensures quality software product. A component of it known as ‘Software Development Life Cycle’ involves a sequence of different activities during the development process. Any software development consists of five phases namely Requirements Analysis, Design, Coding, Testing and Maintenance. Coding is the process of execution of the software with sample data to ensure that the software works correctly without much risk. Risk analysis is an uninfluenced process to find out any error in software development life cycle as a precaution measure to avoid risks in software projects.

Traditional Procedure Oriented Program and Object Oriented Program (OOP) vary from each other in a number of aspects. Object Oriented Program is different from the former, due its special features such as Inheritance, Data Abstraction, and Encapsulation. Besides, in Object Oriented Language it is easy to identify the relationship between the object, element, attributes etc., Reusability is a precedent to the transformation process of the objects from one project to another that is having similar properties. Risk analysis helps by avoiding the adaptive reusability problems
for transformation of coding. Its main feature is to encourage the concept of reusability that paves the way for use of functions and packages. But certain risks might occur during the coding phases of programming in this methodology. This research work has hence emerged for the introduction of an analytical model termed as ARRA, to deal with these problems. The research work presents the design and development of this model. This proposed model ARRA would be useful for reducing the coding risk for any software development team which is particularly working under the C++, Java, Objective-C and Smalltalk platform environments.

The proposed ARRA is purely an analytical tool. This model may although be applied for analyzing software coding for reusing purpose, it is typically designed for adaptable reusability in a new project. The tool focuses on several parameters for adaptability in a new project.

This research work, through extensive literature survey, aims at program analysis, human ethics in coding preparation and reusable properties. It concentrates on transformation, conscientiousness risks, and recognition of transformation, enervative and destructive risks. The Risk analysis is limited to functions of C++ and to packages of JAVA.

Program analysis is a method that provides information about quantum of coding preparation that is required for any software project. In addition, it can aid in allocating software professionals into appropriate projects based on their skill set and expertise. This research study tried out with a sample coding that was carried out by different programmers with varying levels of programming expertise. Each programmer had different skill set which was exhibited in his coding techniques, program volume, and
manner of execution. Based on these factors, software professionals were
allocated and grouped together with various expertise levels depending on
their specialization and choice in selection of platforms for the smooth
development of coding. This program analysis would bring down the risks
involved and enhances the procedure by reducing the code in the software
programming implementation.

Program analysis is also useful in finding out the human error in
coding and implementation. Reusability technology or simply ‘Reusability’ is
used to minimize the efforts for coding, in addition to reducing time and cost.
Coding is a well tested functionality in existing systems for the reuse of
software component to other systems.

Code development is generally based on patterns of design.
Therefore reusable component is introduced to reduce the risk factors in
projects. Risk is directly proportional to the complexity of a system and risk
is inversely proportional to the number of reusable components used in a
project.

It is too complex for a human being to identify the adaptability of
risk analysis before transformation to any new system. In such a case,
analytical model is needed to be created for ensuring the transformation of
reusability component to the system. That is the rationale behind this
research. Based on this model, the risk reduction process involved in various
stages, such as the assay of transformation, adaptability analysis, analysis of
risk identification, impact level of risk classification and recognition of
transformation with tautological properties of set theory are carried out in this
model. Sample programs are created using C++ and JAVA packages.
The properties of a C++ function such as function links, compilation procedures etc, of one project are transformed to another project in a reusable form through ‘Inside-Out and Outside-In’ adaptive risk analysis by this proposed ARRA model. It is also proved that this model is effective in reusability by reducing the risks and failure levels to a virtual zero state.

Reusability of package in JAVA, based on this ARRA model, involves analysis of the total number of identified packages in a program. In addition, it depends on the number of reusable packages that are required for customer design, measuring coupling and cohesion for reusability. If the risk level is in state one, then it will ensure the reusability package for another project. Else, it would not.

The components elaborated above are analyzed and tested through set theory and tautologies of transformation. It is thus established that this model can be developed as a tool for adaptability transformation of reusable components, as the model has been tested and validated with sample inputs and outputs.

This proposed model, to the knowledge of the researcher, has been developed for the first time in adaptive reusability risk analysis of transformation in Object Oriented Programming. It is not only useful for finding and ensuring the transformation of reusable components, but also to create a specific tool for the reusable transformation of functions in C++ and packages in JAVA. It is thus concluded that this model can be effectively used for finding software quality, through inside - Out, out - Inside, Coupling and Cohesion concepts.