



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 9, November 2013

A CONCISE SURVEY OF RISK MANAGEMENT IN SOFTWARE REUSABLE COMPONENTS

M.Iyapparaja¹ Dr. S.Sureshkumar² V.B Santhoshraj³

Senior Asst.Professor, Dept. of CSE, K.S.R College Of Engineering, Tiruchengode, Tamilnadu, India¹

Principal, Vivekanandha College Of Technology For Women , Tiruchengode, Tamilnadu, India²

Final M.E-CSE, K.S.R College Of Engineering, Tiruchengode, Tamilnadu, India³

ABSTRACT: Risk analysis is a process that involves mitigation of errors during various processes of software development. Early identification of risks should shorten the software development completion time. It will also provide efficient solutions. A complete risk analysis along with identification, monitoring and management plan is sure to ensure high quality software products. In this paper reviewed different perception of reusability and the techniques followed to mitigate the risk arose while reusing components.

Keywords: Software Engineering, Software Reusability, risk Management, Object oriented programming

I. INTRODUCTION

Software Engineering is a systematic approach to development, operation, maintenance and up gradation of software. In general, Software Engineering is a discipline which provides tools and techniques to develop quality software in an orderly fashion. It provides an engineering approach that is used to build software for end users using well defined, well managed, consistent and cost effective processes. The Software Development Life Cycle (SDLC) is the sequence of different activities that take place during the development. Any software development consists of five phases namely Requirement Analysis, Design, Implementation, Testing and Maintenance (Pressman 2000). The SDLC begins with the identification of the requirements of a software and ends with the formal justification for the development of software against that requirement. SDLC is the period of time that starts, when a software product is conceived and ends when the product becomes dormant.

II. NEED FOR RISK ANALYSIS IN SOFTWARE PROJECT

Occurrences of errors during the execution of a program need to be predicted before implementation. This is generally done through Risk analysis. For instance, classifying data types with variables would reduce the risk of errors. Literatures have pointed out the requirements of risk analysis for these types of problems. A risk is a potential problem which may or may not occur. Though software development is a very difficult process that determines risks and taking practical measures to avoid them, risk analysis becomes mandatory. In addition, to manage the risks, in order to reduce the complexity and the efforts taken for it, also becomes mandatory (Boehm 1991).

A.) *Software Risks:* The risk has two characteristics,

- 1) Uncertainty: The risk may or may not occur.
- 2) Loss: Unwanted consequences or losses will occur if risk is present.

To quantify the levels of uncertainty and the degree of loss associated with the risk, the type of risks should be categorized as follows: Risks which are external to program coding have been delimited to the study for this research. Four of these risks namely

- 1) 'People Risk' would lead to reduction of efficiency,
- 2) 'Project Risk' is sensitive to project cost
- 3) 'Technical Risk' is related to the time of implementation and
- 4) 'Business Risks' influence the software viability. Hence these issues are considered for the research study.
- 5)

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 9, November 2013

III. CONCEPT OF REUSABILITY

A system is said to be reusable, if it has the capability of using its components in an orderly way. In short, a part or whole of one system can be incorporated to assemble a new system. As far as software is concerned, it makes the reusability simpler and competent to frame a new module which is free from bugs and risks (William Frakes and Carol Terry 1996). Table 1 depicts the merits and demerits involved in the concept of reusability.

Merits	Demerits
Increases adaptability Reduces process risk Reduces time and cost	Lacks in tool support Creating and maintaining a component library which creates overhead Adaptability in reusability is cumbersome

Table 1. Reusability Merits & Demerits

IV. LITERATURE REVIEW

The technique of programming is widely brought up into two categories which are as follows:-

- 1) 1. Procedure oriented programming
- 2) 2. Object oriented programming

These two vital techniques i.e., Procedure Oriented Program and Object Oriented Program are different from each other (Firesmith 1993 and Ryder et al 2005).

The former language which is procedure oriented usually follows algorithm-centric approach in which a program is driven by an algorithm that traces its path of execution from beginning to end. Data is an external entity the operation of which is defined upon by the algorithm. Fundamentally, these classes of programming languages are characterized by data that are considered and separated from the operations or program part. Algorithm which is the primary most stepping stone, which being the driver, with data being subsidiary to the algorithm, the procedure oriented program just breaks up a larger job into a number of subtasks and subprograms independently and conveniently as functions and subroutines. The functions and the subroutines which makes a complex job with best output, put together as a whole forms an efficient, easily debugging capability rich and a best programming part.

The main notion of the procedure oriented language is to make the process of debugging a program and to increase the number of reusable procedures in programs. The importance of program reuse is thus realized by software professionals. By adopting reusability techniques in programming, the cost of building software would be reduced considerably which have paved way for the emergence of Object Oriented Language. Khan Al-A'ali and Girgis (1995) have described the Object oriented programming language to be data or Object centric.

These programming languages which are described to be object oriented plays role in the promotion of objects which in turn helps in identification of the relationship between data and object that enhances creativity in the technique of reusability. When certain common data and objects are used in various different programs, it will greatly contribute in the reduction of time of development for the programmers and thereby increasing the efficiency of the program. Reusable components are chosen in such a way that they will be best suited for the activities in the implementation phase, to reduce risk. The implementation of code reusable components is mainly focused in this thesis.

Nunamaker et al (1989) put forth brief ideas of approach in order to understand the highly complex process of development of software and to improve the productivity of software including the using and designing of automated software development tools. The study of human factors in software development and application of software productivity measurement and evaluation technique are some other processes.

An environment to facilitate the reuse of software components is described. Such an environment supports the identification, creation, categorization, storage, selection, usage, and maintenance of reusable components. A meta system environment that allow users to define functionalities, structures, and constraints of various software components is discussed. Informations regarding all these components is being used by a knowledge-based system in order to support the selection, configuration, and distribution of the components that are reusable.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 9, November 2013

Ramamoorthy et al (1993) have stated that the implementation of risk management over the reusability technique of software development is based on two knowledge tools,

A.) *Risk assessment*

This tool will greatly help the knowledge based systems that are involved in analysing all the metric data to high risk and low risk components. Influence Diagram Based Expect System (IDES) helps in the estimation of expected loss of each risk item. The influence of problem and decision on it, rules and inference engine are solved by the system. The conflict resolution strategy is also provided by it.

B.) *Consistency management*

Consistency management system ensures proper operation of the system after correction and measurement of the relationship between the components in each phase. This approach is called Assumption-Based Truth Maintenance System (ATMS) which is used by any team for dependency relationship between the software object and data throughout the life cycle of the development of system. This tool can help in the prevention of disaster and also in the reduction of impacts of the risks.

Domain Analysis method is to determine similar software patterns from the experiences of development to satisfy the current needs of customers. Past experience guides to reuse the existing software products and improves the quality of the delivered product. The system analyst has to undertake a full fledged process of analysis of the current problems of the customer and finding the similarity of requirements from any completed projected requirement. Domains of products have to be identified for comparing the current problem domain to minimize the efforts for developing the system. This technique of analysis of domain was declared by Basili et al (1994).

Object Oriented Metrics rely on programs and are estimated through an empirical study which focuses on improvement process which increases the necessity of using software measures and metrics. The need for such metrics was particularly notable when an organization was adopted for any further enhancement in programs. An automated data collection tool was designed and implemented to collect an empirical sample of these metrics at two field sites to demonstrate their feasibility and sorted out ways in which project managers may use these metrics for process of improvement by the teams of Chidamber and Kemerer (1994).

Paul Raymond et al (1996) have carried out an elaborated discussion in their paper, the use of an object oriented data modeling approach for quality and risk requirements pertaining to software projects. The proposed approach provides an environment for users to develop individual views and permits the system to process semantically heterogeneous queries. Here a temporal modeling framework is employed which allows the users to capture temporal semantics of information associated with software metrics. Formalism for temporal modeling of metrics data using a set of temporal relations is also created. Further, an object oriented query model to receive semantics associated with software quality and risk is obtained for modeling temporal semantics of software data. To express queries, the notion of scenario was also introduced where one scenario is identified through raw values or processed data. Temporal modeling is used to express complex scenarios spanning various metrics and to help in identifying risk and also for evaluating the quality of the software project at various levels of abstraction.

The higher conceptual studies on examining the process how the organizations actually exploit the reuse technologies and evaluate the reuse factors are reported. These factors finally affect the rate of reuse in an organization thereby to attempt to enhance the measurement of the rate of reuse.

The effectiveness of reuse by establishing conceptual foundations in the literature for reuse and conducting an empirical investigation of organizations using Ada technology is reported. This study established a wide difference among the software reuse criteria such as domain, humans, tool, organization, software metrics, and environment.

The rate of reuse (as shown by Nam Yong Lee and Litecky (1997)) significantly depends upon

- 1) reuse capability,
- 2) software development effort,
- 3) object-oriented design capability,
- 4) repository development effort,
- 5) Ada technology capability, and
- 6) domain capability.

The management of changes in reusable software was discussed and carried out by David Kane et al (1997) and his fellow members. Organizations which undertake software reuse need to manage changes in components which grow due to expanded usage. Often, this is an expanded life cycle of software assets. The six patterns to support reusable software that emerged out of this discussion report are



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 9, November 2013

- 1)develop a shared platform
- 2)maintain the reuse platform identity
- 3)integrate reuse and tie to the bottom line
- 4)reuse code
- 5)treat reusable components like product and
- 6)merge clones.

Each of this pattern must be described with their problem statement and with their solution of reusable software. United State Navy Software Maintenance Organization management has created a process improvement activity for upgrading the overall quality of internal software maintenance. This was exclusively explained by Charette et al (1997). It provides a simple consensus method of risk assessment and mitigation.

The primary focus of this approach is on addressing the real risk rather than perceived ones. They state that there is a significant difference between maintenance oriented risk management and development oriented risk management. They have listed a suggestion of specific risk management information as input to specific risk management process.

V. CONCLUSION

In any software development set up, professionals differ from each other in their attitude, technical skill and ability. Talented software professionals form a great asset to an organization. They, therefore reduce the levels of risks involved in successful product development.

The extensive literature reviewed in this paper, has indicated that there is no model available for adaptive reusability with reusable components in risk analysis for object oriented programs. An innovative model has been proposed in the name of '**Proficient Risk Extenuation Model (PRE-M) for Adaptive Software Reusable Components**' and the design and developmental details along with its validity should be measured to focus on mitigating the risk occur in software component reusability.

REFERENCES

- [1] Basili V.R., Bri L.C. and Thomas W.M., 'Domain Analysis for the Reuse of Software Development Experiences', Experimental Software Engineering Group (ESEG), Vol. 11, pp. 86-95,1994.
- [2] Boehm B. W., 'Software Risk Management: Principles and Practices', Journal of IEEE Software, Vol.8, No 1, pp. 32-41,1991.
- [3] Charette R.N., Adam K.M, White M.B., 'Managing risk in Software Maintenance', IEEE Software ,Vol.14, No.3,pp.43-50,1997.
- [4] Chidamber S.R. and Kemerer C.F., 'A metrics suite for object oriented Design', IEEE Transaction on Software Engineering, Vol. 20, No.6, pp. 476-493,1994.
- [5] David Kane, 'Managing Change to Reusable Software', PLoP 97 Conference, pp.1-12,1997.
- [6] Firesmith D.G., 'Testing Object Oriented Software', Proceedings of Eleventh International Conference on Technology of Object Oriented Languages and Systems, pp.406-427,1993.
- [7] Khan Al-Aali and Girgis, 'Object Oriented Programming for Structured Procedural Programming', Journal of Systems and Software, Vol. 17, pp.111-117,1995.
- [8] Nam Yong Lee and Litecky C.R., 'An Empirical Study of Software Reuse with Special Attention to Ada', IEEE Transaction on Software Engineering ,Vol.23, No.9, pp.537-549,1997.
- [9] Nunamaker J.F. and Chen M., 'Software productivity: a framework of study and An approach to reusable components' System Sciences, Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference, Vol. 2, pp.959-968,1989.
- [10] Paul Adams, Cornelia Bolyreff, David Nutter and Stephen Rank, 'Adaptive Reuse of Libre Software Systems for Supporting Online Collaboration', Proceedings of the fifth workshop on Open source software engineering, pp.1-4,2005.
- [11] Pressman R.S., 'Software Engineering: A Practitioner's Approach', Fifth Edition, McGraw-Hill International Edition, 2000.
- [12] Ramamoorthy C.V., Chandra C., Ishihara S. and Ng Y., 'Knowledge based tools for risk assessment in software development and reuse', Proceedings of the 1st International Conference on tools with Artificial Intelligence, pp.364-371,1993.
- [13] Ryder B.G., Mary Lou Soffa and Margaret Burnett, 'The impact of software engineering research on modern programming languages', ACM Transactions on Software Engineering and Methodology, Vol. 14, No. 4, pp. 432-477,2005.
- [14] William Frakes and Carol Terry, 'Software reuse: metrics and models', ACM Computing Surveys, Vol.28, No.2, pp 415-435,1996.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 9, November 2013

BIOGRAPHY



M. Iyapparajais presently a Senior Assistant Professor, Department of CSE, K.S.R. College of Engineering, Tiruchengode, Namakkal, Tamilnadu, India. He received the B.E degree from Anna University, Chennai and M.E degree from Anna University of Technology, Coimbatore in 2006 and 2010 respectively. His research interests include Software Testing, Software Engineering and Agile Testing. He is a member of ISTE.



Dr. S. Sureshkumaris presently the Principal, Vivekanandha College of Technology for Women, Tiruchengode. He received the B.E degree from National Engineering College, M.S, degree in Software system from Birla Institute of Technology and Science, M.Tech., degree from Indian Institute of Technology, Kharagpur and Ph.D., degree from Anna University, Chennai in 1988, 1993, 2000 and 2009 respectively. He has published 30 numbers of papers in refereed international journals and conferences. His research interests include Image processing, parallel processing and Energy. He is a member of ISTE and IET.



V.B. SanthoshRaj is presently doing his Final M.E degree under the department of CSE, K.S.R. College of Engineering, Tiruchengode, Namakkal, Tamilnadu, India. He received the B.Sc degree in Computer Technology from Anna University, Chennai and MCA degree from Autonomous, Anna University of Technology, Coimbatore in 2009 and 2012 respectively. His project area of interests includes Software Testing and Software Engineering. He is the member of IEEE.