iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43333

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

A Review of Incremental Reuse Method in Software Reuse Process

Sheleshma Shukla[1], Dr. Dhirendra Pandey[2], Arpan Kumar[3]
[1, 2]*BabaSaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh*
[3]*Dr. A.P.J. Kalam Technical University, Lucknow, Uttar Pradesh*

Abstract: ***In the late 1960s, the concept of software reuse was proposed. The legacy of software engineering process shows that the scope and field of software reuse was not particularly new. However, it is considered that software reuse is in its primary stages and has yet to reach its full capacity. The utilization of software components from entire stages of the software engineering process in new applications is called as software reuse. Given the economically costly and difficulty of building qualitative software, repurposing prior software investments is enticing. We propose that software reuse should be considered as part of a fully systematic approach that considers a wide range of technological, economical, managerial, organisational, and legal considerations, and we wrap up with a synopsis of the important research issues in all these verticals. This paper summarises our progress in software reuse research.***
Keywords: Software Reuse, Incremental Reuse Method (IRM), Reuse Process, Artifact, Software Component.

I. INTRODUCTION

The phrase "software reuse" means to the process of creating software by recreation of existing software components [1]. These are some examples of reusable components of a software product: Interface and source code design, Software documentation, user manuals, software requirement specifications, and so on. Software Reuse helps in requirement of because many of the system's components are ready-made, less work is required. Reusing software requires less work, and saving time reduces total costs. When you have ready-made components, you may concentrate on the new components that are not readily available. The structured way of software reuse was first introduced in 1968 in NATO Software Engineering Conference [1]. After this event the software reuse has been become a notable area of research and development in software engineering. Software reuse is a kind of trending discussion topic in large and medium software companies who want to obtain the best output in terms of software product within a short span of time using the previously existing software system and documentation.

Many companies worldwide has been claimed that due to software reuse, their productivity has been increased by 20% [2]. In spite of these thing, it will not be so good to say that software crisis can be eliminated by software reuse mechanism. Software reuse is not so easy to implement. It requires a great skill and knowledge to implement the software reuse mechanisms at a suitable level of software project development. However, there are strategies that can assist a corporation in maximising its resources and increasing output. When done correctly in the proper setting, reuse has been proved to provide significant benefits. In this paper, we look at software reusability as a way to enhance the software development process as well as the quality of the final product. The use of previously produced software components in new applications is referred to as software reuse. This has traditionally entailed code reuse by other programmers inside the same company. Reusing software in the creation of systems is an approach that boosts efficiency and quality [3].

II. BASICS OF SOFTWARE REUSE

There are some points which need to be considered while studying and implementing software reuse in software industries. The basics of software reuse are as given below:

A. Process [4]

1) *Domain Analysis Phase:* By the means of the design and analysis of the application domain, this stage will determine if the infrastructure should be reused for domain development.
2) *Domain Engineering Phase:* This phase consists of acquiring a well-known system architecture based on the domain commonness obtained during the previous step, as well as regulating how properties correspond to the system architecture and binding of variable points.
3) *Property Acquisition Phase:* This process involves reuse development and maybe some reuse property adoption from the outside.

©IJRASET: All Rights are Reserved

4) *Property Categorization Phase:* This process is basically a database administration stage, as it involves classifying and preserving reusable properties.
5) *Property Maintenance Phase:* This goal is essentially a maintenance task that combines configuration management with edition control.



Fig. 1 Process of Software Reuse

B. Scope

1) *Process Reuse:* The term "process reuse" refers to the reuse of previous software development processes in order to produce a system automatically or partially automatically utilising a reuse generator.
2) *Product Reuse:* The term "product reuse" refers to the process of reusing software components in order to create a new system through component integration and development.

III.SOFTWARE REUSE TECHNIQUE

This research paper is suggesting the incremental approach of software reuse in software engineering process for small and medium level software development companies. In companies before implementing the software reuse process we should ensure that the staff and current software development approach is compatible with the software reuse technique [1]. This is the reason for that we should always keep in mind that we should not directly implement the new development approach.
The Incremental Reuse Method (IRM) strategy reduces the disadvantages previously mentioned in order to provide a very appealing set of circumstances for a company to "jump into" reuse.

These are the prerequisites for us [5]:

1) Software reuse should be based on the facts that investment should be almost zero.
2) Software reuse implementation should not require too much training for staff.
3) There should be no requirement to have or model anything for reuse in advance if Software Reuse can be introduced gradually. The artefacts from completed projects are reused in the creation of new ones.
4) There should be no requirement to have or model anything for reuse in advance if Software Reuse can be introduced gradually. The artefacts from completed projects are reused in the creation of new ones.
5) Software reuse process must be robust and flexible.
6) Organization should also consider below given criteria:

a) Problems with Quality Assurance
b) Knowledge Management and Time To Market
c) The value of knowledge as an organisational asset

Asset retrieval approaches and technologies are considered critical in IRM. The advances made in the 1980s, when the aspects technique was published, do not permit looking for artefacts based on their content. A repository must be able to represent all distinct artefacts' contents regardless of their typology, as well as provide retrieval techniques for identifying artefacts by content, to succeed with this strategy. The RSHP meta-model is used to create the repository representation schema in IRM.
The acronym RSHP stands for "RelationSHiP," and it was designed to represent a wide range of data using a single meta-model that allows all possible information models to be stored in the same source, providing for tracking in between portrayed data artefacts.

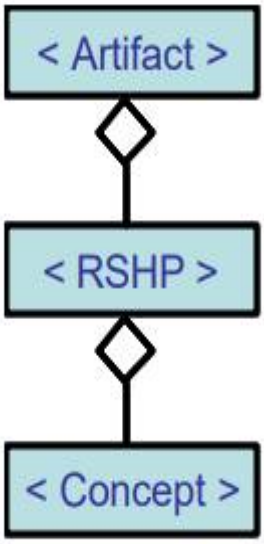


Fig. 2 Meta-Model of RSHP

The Incremental Reuse Method claims to provide a framework for establishing a Reuse Unit within any interested business without making significant expenditures. The dimensional base of these are as shown in figure:

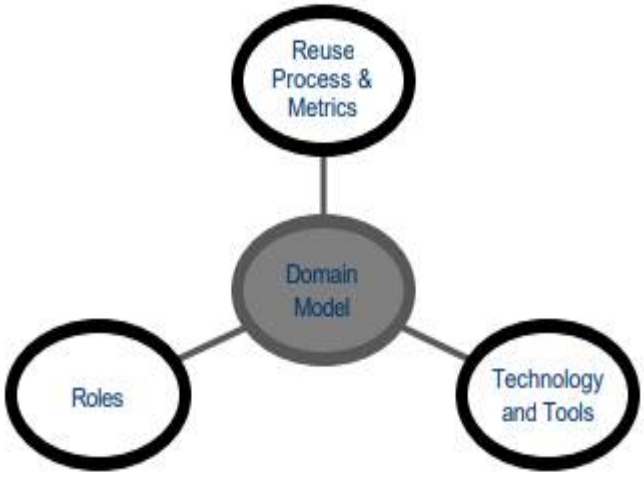


Fig. 3 Dimensions of IRM

To eliminate obstacles and expenses for prospective reusers, the software development process should not be significantly impacted when a reuse programme is implemented.

The reuse process should envelop the organization's SDP, adding just a few new activities to the existing one and a few additional phases to the software lifecycle.

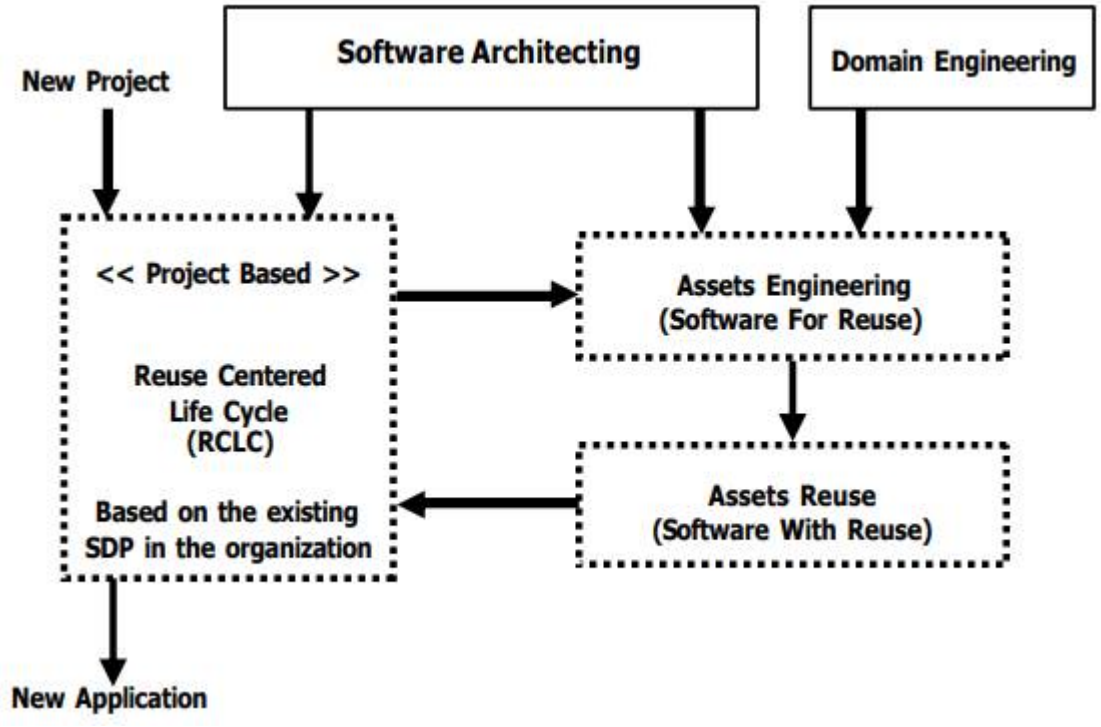


Fig. 4 IRM Software Reuse Process

IV. CONCLUSION

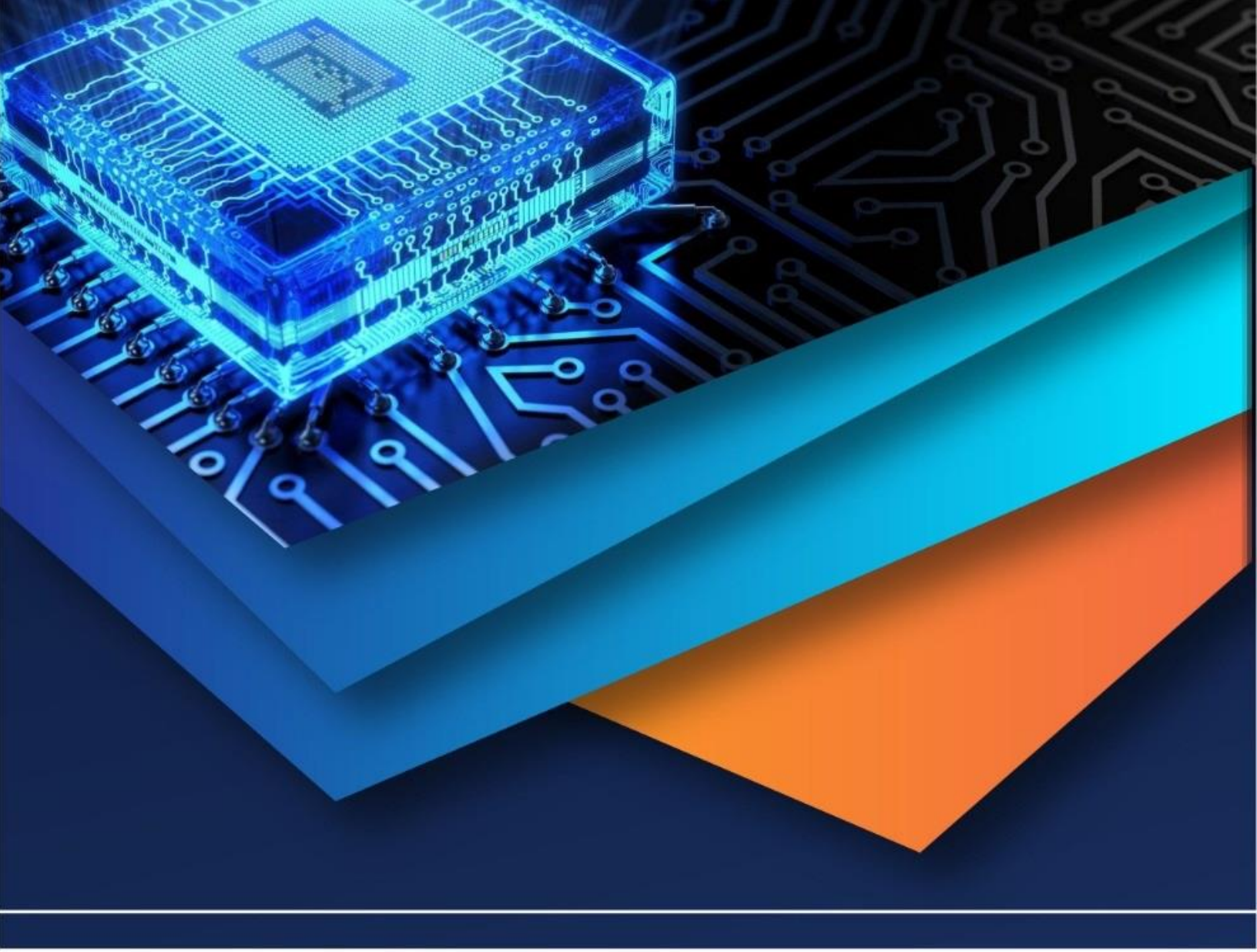
Despite a variety of obstacles, the preceding article illustrates how reuse is beneficial in today's software development context. So far, the outcomes have been mixed. While there have been some achievements, the obstacles and problems experienced during the endeavour have also been fascinating. Small businesses are distinct in their need to stay current with market developments and complete each project successfully.

A small firm, unlike bigger corporations or even single project teams inside larger organisations, cannot afford to fail on any project since the company's and every employee's survival depends on maintaining market share. As a result, most software development firms have inadequate reuse activities.

The Incremental Reuse Method (IRM) is built on allowing all activities in the software development process to retrieve and reuse artefacts. These artefacts are kept in a repository that is kept up to date by incrementally indexing all of an organization's project work products. Its cheap installation cost and minimal impact on current SDP activities have resulted in high implementation success rates in real-world scenarios.

REFERENCES

[1] Ajay Kumar, "International Journal of Computer Science and Communication Vol. 2, No. 2, July-December 2011, pp. 659-662".
[2] Matsumoto, Y.; 'Some Experiences in Promoting Reusable Software: Presentation in Higher Abstract Levels'; In: Software Reusability. Concepts and Models, vol. II; Biggerstaff, T.J., Perlis, A.J. (ed.); ACM Press, Addison-Wesley, Reading, Mass.; 1989
[3] Boehm, B., "Managing Software Productivity and Reuse", Computer, vol. 32, issue 9, Sept 1999, pp. 111- 113.
[4] Deepshikha Jamwal, "Proceedings of the 4th National Conference; INDIACom-2010 Computing For Nation Development, February 25 – 26, 2010".
[5] Llorens, Juan & Fuentes, José M. & Prieto-Diaz, Ruben & Astudillo, Hernán. (2006). Incremental Software Reuse. 4039. 386-389. 10.1007/11763864_29.
[6] Boehm, B., "Managing Software Productivity and Reuse", Computer, vol. 32, issue 9, Sept 1999, pp. 111- 113.
[7] Tracz, W.; 'Software Reuse Maxims'; ACM Software Engineering Notes; Oct 1988; Vol.13 No.4 P28-31



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)