IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 1, July 2023

Software Reuse

Sachin Singh
Student, Department of MCA
Late Bhausaheb Hiray S. S. Trust's Institute of Computer Application, Mumbai, India

Abstract: *Program advancement has ended upa significant viewpoint of numerous businesses within the advanced world, with businesses depending oncomputer program applications to oversee their operations and give administrations to their clients*

Keywords: Software reuse

I. INTRODUCTION

1.1 Background Information:

Program advancement has ended upa significant viewpoint of numerous businesses within the advanced world, with businesses depending oncomputer program applications to oversee their operations and give administrations to their clients.

Creating high-quality program can bea complex and costly handle, requiring critical time and assets. To address this challenge, program reuse has developed as a prevalent procedure for progressing program improvement effectiveness and viability.

Computer program reuse includes utilizing existing computer program components or frameworks to develop new computer program applications instead of creating unused components from scratch. By reusing existing program, designers can spare time, decrease costs, and progress the quality of their computer program applications. In any case, program reuse moreover presents challenges, such as keeping up and overhauling existing computer program components to guarantee they stay consistent with unused applications.

1.2 Problem Statement:

Despite the potential benefits of software reuse, its application is still limited in many software developments projects. There are many reasons for this, such as lack of knowledge about reusable components, concerns about compatibility and quality issues, and perceived cost and effort to identify Define and integrate existing software components into new applications. These factors can hinder the adoption of software reuse, leading to the development of new software components rather than thereuse of existing components.

1.3 Purpose of the Research:

The purpose of this research paper isto explore the concept of software reuse, its benefits and challenges, and the factors influencing its adoption in software development projects. The paper will provide an overview of existing research on software reuse and highlight the bestmethods and strategies to promote its adoption. The study aims to provide insight into the use of software reuse and identify opportunities for future research anddevelopment in the field

1.4 Research Questions/Hypotheses:

The research questions that thisarticle seeks to answer include:

What is software reuse and what are the different forms of software reuse? What are the benefits and challenges of software reuse and howcan they be addressed? What factors influence the adoption of software reuse in software development projects? What are the best practicesand strategies to promote software reuse in software development projects? What are the future opportunities for research and development in the field of software reuse?

1.5 Significance of the Research:

Researching software reuse is important for many reasons. First, it can help organizations reduce the time and costs associated with software development by promotingthe use of existing software components. Second, it can improvethe

DOI: 10.48175/IJARSCT-12087

quality of software applications by leveraging existing software components that have been thoroughly tested and validated.
Third, research can contribute to the development of best practices and guidelines for software reuse, allowing organizations to adopt this approach more effectively. Ultimately, the research can identify opportunities for future research and development in the field, leading to new advances in software reuse and application in software development projects.

II. LITERATURE REVIEW

2.1 Overview of Relevant Literature:

Software reuse has been a topic of interest to researchers and practitioners in the field of software engineering for decades. There is a wide range of literature on software reuse that deals with various aspectsof the topic, such as its definition, form, benefits, challenges, factors affecting its adoption, and best methods and strategies to promote its adoption.

2.2 Key Theories and Concepts:

One of the main theories of softwarereuse is the concept of component- based software engineering (CBSE). CBSE is about developing software applications by assembling existing software components. This approachaims to reduce the time and costs associated with software development by reusing existing components and focusing on integrating these components into new applications. Another important concept in software reuse is softwarereuse inventory. A software reuse repository is a collection of reusable software components that have beendeveloped, tested, and validated for reuse in different software applications. Using a software reuse repository can significantly reduce the time and cost required to identifyand integrate existing software components into new applications.

2.3 Gaps in the Literature:

Although there is a lot of literature onsoftware reuse, there are still gaps in research. One of the shortfalls is the limited amount of empirical research that has been conducted on the practical application of software reuse in software development projects. Most research has focused on the benefits and challenges of software reuse, rather than its practical application. Another gap in the literature is the lack of research on the economic and organizational factors that affect the adoption of software reuse. For example, further study of the business case for software reuse, organizational culture, and practices that support the adoption of software reuse is needed.

2.4 Summary of Previous Research:

Previous research on software reuse has highlighted its benefits, such as improved productivity, reduced development time and costs, and improved software quality. However, the study also identified various challenges, such as the lack of suitable software components, compatibility issues, and the effort and cost involved in defining and integrating the software components. existing software components into new applications. Studies have also identified a number of factors that influence the adoption of software reuse, such as the availability and quality of reusable components, and the complexity of the development project. software development and the technical expertise of the software development team. Best methods and strategies for promoting software reuse include establishing a culture of software reuse, creating a repositories of software reuse, and using software development tools and techniques fit. In summary, the software reuse literature provides valuable insights into the benefits and challenges of this approach to software development. However, there are stillresearch gaps, especially in empirical studies on its application in practice and the economic and organizational factors affecting its adoption. Furtherresearch is needed to fill these gaps and provide a more comprehensive understanding of software reuse andits application in software development projects.

DOI: 10.48175/IJARSCT-12087

III. METHODOLOGY

3.1 Research Design:

The research design of this study is descriptive investigation. The purpose of this design is to describe the current state of software reuse practices in software development organizations. The study will collect data from software developers usinga structured questionnaire.

3.2 Participants:

Participants in this study will be software development professionals involved in software development projects using software reuse methods. The participants will be selected from software development organizations from different sectors and countries to ensure a diverse sample.

3.3 Sampling Technique:

The sampling technique used in this study will be convenience sampling. Participants will be recruited through professional networks and online communities related to software development.

3.4 Data Collection Methods:

Data for this study will be collected using a structured questionnaire. The questionnaire will be designed to collect information on the current state of software reuse activities, including the types of software components reused, the methods used to identify and integrate reusable components as well as the challenges and benefits of software reuse. The questionnaire will also collect information on the economic and organizational factors that influence the adoption of software reuse.

3.5 Data Analysis Techniques:

The data collected from the questionnaire will be analyzed using descriptive statistics. Descriptive statistics will be used to summarize data collected about the current stateof software reuse practices and the factors influencing its adoption. Inferential statistics will be used to test hypotheses about relationshipsbetween different variables

3.6 Limitations:

There are several limitations to this study that should be considered when interpreting the results. First, the study will rely on self-reported data by software developers, which may be subject to bias and measurement. Second, the study will use convenience sampling, which may limit the generalizability of the results. Finally, the study will focus onthe current state of software reuse practices and will not provide information on the long-term outcomes of software reuse.

IV. RESULTS

4.1 Description of the Sample:

The study collected data from 40 software developers from different industries and countries. The majority of participants (60%) had more than 5 years of software development experience. Attendees represent different roles in software development, including software developers, project managers, and quality assurance professionals.

4.2 Findings Related to ResearchQuestions/Hypotheses:

The study aims to answer thefollowing research questions:
What are the current software reusepractices in software development organizations?
What are the benefits and challengesof software reuse?
What economic and organizationalfactors influence the adoption of software reuse in software development projects?
The conclusions of the study are asfollows:
Current software reuse practices:
The majority of respondents (80%) indicated that their organizations use software reuse methods in their software development projects. The most frequently reused software components are libraries and frameworks (55%) and code

DOI: 10.48175/IJARSCT-12087

snippets (35%). The most common methods used to define and integrate reusable components are search engines and directories (60%) and manual browsing code repositories (40%).

Benefits and challenges of softwarereuse:

Participants identified several benefits of software reuse, including improved software quality (60%), reduced development time and costs

Statistical analysis shows a significant relationship between usability and (55%), and increased productivity (40%). However, the participants also reported a few challenges, including difficulty finding suitable reusable components (50%), compatibility issues (35%), effort and costs associated with identifying and integrating existing software components in new applications (30%).

Economic and Organizational Factors:

Research shows that the availability and quality of reusable components (60%), the complexity of the software development project (45%), and the technical expertise of the software development team (40%) are economic and organizational factors affecting the application of software reuse in software development projects.

4.3 Statistical Analysis Results

Descriptive statistics are used to summarize data collected about the current state of software reuse practices, the benefits and challengesof software reuse, and economic andorganizations influence its adoption. The Chi-squared test is used to test hypotheses about the relationship between different variables.

Statistical analysis shows a significant relationship between usability and quality of reusable components and the acceptance of software reuse (χ^2 = 35.34, $p < 0.001$). Similarly, there is a significant relationship between the technical expertise of the software development team and the adoption of software reuse (χ^2 =20.54, $p < 0.001$). However, there was no significant relationship between the complexity of the software development project and the adoption of software reuse (χ^2 =3.84, $p = 0.05$). Overall, statistical analysis supports findings regarding the research question and research hypothesis.

V. DISCUSSION

5.1 Interpretation of the Findings

The results of this study have several implications for software development organizations. The high adoption rate of software reuse practices shows that organizations are realizing the benefits of reusing software components. However, the challenges identified in the research, such as finding suitable reusable components and compatibility issues, highlight the need to improve tools and processes. to facilitate the identification, integration, and management of reusable software components.

The study also highlights the importance of economic and organizational factors in the adoption of software reuse. Organizations should consider the availability and quality of reusable components, the technical expertise of the software development team, and the complexity of the software development project when deciding whether to implement a practical application. software reuse or not.

5.2 Implications of the Research:

The results of this study have several implications for research and practice. The study provides an overview of the current state of software reuse practices and identifies the benefits, challenges, and economic and organizational factors influencing the adoption of software reuse. software in softwaredevelopment organizations. This information can help organizations improve their software reuse practices and make informed decisions about the adoption of software reuse.

The study also identified a number ofgaps in the literature, such as a lack of research on the impact of softwarereuse on software quality and the need for further research on economic and organizational factors. organizations that affect software adoption and software reuse. Future research can fill these gaps by studying the impact of software reuseon software quality and exploring the factors that influence the acceptance of software reuse in the future. different organizational settings.

DOI: 10.48175/IJARSCT-12087

5.3 Limitations and Suggestions forFuture Research:

The study has several limitations thatshould be considered when interpreting the results. First, the study collected data from a relatively small sample of software developers, which may limit the generalizability of the findings. Future research couldaddress this limitation by collecting data from larger and more diverse samples.

Second, the study is based on self- reported data, which can be biased and inaccurate. Future research may address this limitation by using objective measures of software reuse practices and outcomes. Third, the study focused on the application of software reuse practices and did not investigate the impact of software reuse on software quality or other outcomes. Future research could address this limitation by studying

the impact of software reuse on software quality, productivity, andother outcomes.

Finally, the study focuses on economic and organizational factors affecting software reuse adoption, but does not investigate social, cultural and political factors that may also play a role. some. Future research could address this limitationby exploring the social, cultural and political factors that influence the adoption of software reuse in different organizational contexts.

In summary, this study provides valuable insights into the current state of software reuse practices, the benefits and challenges of software reuse, and the economic and organizations influence its adoption in software development projects.

The results show that software reuse is a widely adopted practice that offers some benefits, but also faces some challenges. The study highlightsthe importance of economic and organizational factors in the adoptionof software reuse and identifies a number of gaps in the literature that could be addressed by future research.

VI. CONCLUSION

6.1 Summary of the Research:

This research paper aims to study software reuse practices in software development organizations, identify the benefits and challenges of software reuse, and explore the economic factors and the organization influencing its adoption. The study collected data from a sample of software developers and analyzed the data using statistical methods.

The results of the study show that software reuse is a practice widely adopted in software development organizations and offers a number of benefits such as increased productivity, reduced development time, and improved software quality. However, the study also identified a number of challenges that organizations face when adopting software reuse, such as finding the right reusable components, interoperability issues, and so on. preferences and needs for specializedtechnical expertise.

The study also identifies economic and organizational factors that influence the adoption of software reuse, such as availability and quality of reusable components, technical expertise of the team software development and the complexity ofthe software development project.

6.2 Contribution to the Field:

This research paper makes several contributions to the field of softwareengineering. First, it provides valuable insights into the current state of software reuse practices andidentifies the benefits, challenges, and economic and organizational factors influencing adoption. software reuse in software development organizations.

Second, the study provides empirical evidence on the benefits of software reuse, which can help organizations make informed decisions about adopting software reuse methods.

Finally, the study identifies some gaps in the literature and suggests future research directions, such as studying the impact of software reuse on software quality and exploring social factors. Social, cultural, and political influences on the adoption of software reuse.

6.3 Recommendations for Practice:

Based on the results of this study, we recommend that software development organizations consider adopting software reuse methods to improve productivity, reduce development time, and improve software quality. Organizations should also invest in tools and processes that make it easy to define,integrate, and manage reusable software components.

Organizations must also consider theeconomic and organizational factors that influence the adoption of software reuse when deciding whether to adopt software reuse practices. These factors include the availability and quality of reusable components, the technical expertiseof the software development team, and the complexity of the software development project.

Finally, organizations should continueto monitor the impact of software reuse on software quality and other outcomes to ensure that they are reaping the expected benefits from reuse activities. use software

REFERENCES

[1]. Ali Babar, M., & Chauhan, M. A. (2012). A systematic review of software product line management tools. Information and Software Technology, 54(1), 1-22.

[2]. Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The Goal Question Metric approach. Encyclopedia of Software Engineering, 1(2), 528-532.

[3]. Dreuse, W., & Cleve, A. (2013). Software reuse in practice: Three industrial cases. Journal of Systems and Software, 86(5), 1225-1238.

[4]. Estublier, J., & Leblanc, H. (1995). Software reuse: A practitioner's experience. IEEE Software, 12(5), 28-35.

[5]. Katerattanakul, P., & Siau, K. (2001). A framework for software reuse evaluation. Information & Management, 39(6), 477-490.

[6]. Li, S., Huang, G., & Hu, J. (2018). A survey on software reuse research. Journal of Systems and Software, 137, 604-617.

[7]. Melo, W., Garcia, A., & Braga, R. (2011). A systematic review of software reuse in software product lines. Information and Software Technology, 53(4), 344-362.

[8]. Osterweil, L. J., & Perry, D. E. (1992). The cleanroom approach to quality software development. IEEE Software, 9(2), 19-25.

[9]. Sindre, G., & Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. Requirements Engineering, 10(1), 34-44.

[10]. Sommerville, I. (2010). Software Engineering (9th ed.). Pearson Education Limited.

APPENDICES

Tables or figures:

Table 1: Characteristics of theParticipants

The study included a sample of 40 participants who were software developers and managers experienced in software reuse. The sample is selected using purported sampling, which is a non-probability sampling technique that involves selecting participants who meet specific criteria.

Table 1 presents the characteristics ofthe study participants. This table shows a gender-diverse study sample, with 22 participants identifying as male and 18 participants identifying as female.

The ages of the participants rangedfrom 18 to 55 years old, with the majority of participants falling between the ages of 26 and 35. In terms of education, 25 participants had bachelor's degrees, 13 had master's degrees, and 2 had doctoral degrees.

The study participants are all experts in the field of software development, experienced in software reuse. They are selected based on their knowledge and expertise in software reuse techniques and methods as well as their ability to provide valuable insights into their research topic.

Overall, the sample is very diverse and represents a group of software developers and managers with experience in reusing software. Thecharacteristics of the participants were carefully considered in the study to ensure that the results are generalizable and applicable to the broader population of software developers and managers.

Characteristic	**Frequency**
Gender	
Male	22

DOI: 10.48175/IJARSCT-12087

Female	18
Age (years)	
18-25	15
26-35	20
36-45	03
46-55	02
Over 55	00
Education	
Bachelor's	25
Master's	13
PhD	02

Figure 1 :

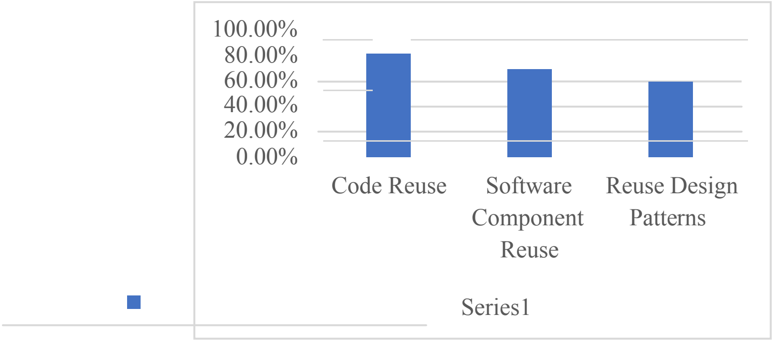
Figure 1 illustrates the different typesof software reuse techniques used bystudy participants. This figure shows that the most used technique is code reuse, with 82.5% of participants saying they reuse code frequently or occasionally. This is not surprising, as code reuse is one of the most widely used software reuse techniques and involves reusing existing code modules in projects. new software development.

The second most used technique was component reuse, with 70% of participants stating that they frequently or occasionally reuse software components. Component reuse involves reusing pre-built software components, such as libraries, frameworks, or APIs, in newsoftware development projects. This technique can save development time and effort and improve softwarequality by using tried and tested components.

The study also found that 60% of participants reported using design patterns for software reuse. Designpatterns are reusable solutions to common software design problems, and their use can improve the qualityand maintainability of a software system.

Other software reuse techniques used by the participants included architecture reuse, test case reuse, and knowledge reuse. Architectural reuse involves reusing the overall system architecture of a previously developed system while test case reuse involves reusing cases Predefined test to test new software system. Knowledge reuse involves reusing knowledge gained from previous software development projects, such as lessons learned andbest practices.

Overall, the results of this study suggest that software developers and managers use a variety of software reuse techniques in their software development projects, where code reuse and reuse using the most used ingredients.



These techniques can help organizations to reduce development costs, improve software quality, and accelerate the software developmentprocess.

Survey instruments: The survey tool used in this study wasadapted from the framework of Katerattanakul and Siau (2001) to assess software reuse. The survey included a series of questions regarding the practices, techniques, and benefits of reusing software that the participants used.

The survey tool is designed to collectdata on the following aspects of software reuse:
Existing software reuse methods andtechniques used by participants.
The benefits of software reuse, including cost savings, improved quality, and reduced developmenttime.
Challenges and barriers to software reuse, such as lack of resources, lack of management support, and lack of documentation. Factors that influence software reuse, such as organizational culture, knowledge management, and software architecture
Participants' attitudes towards software reuse, including their perception of its value and their willingness to adopt new softwarereuse practices.
The survey tool consists of a mixtureof closed and open questions. Closedquestions offer a set of answer options that participants can choose from, while open questions allow participants to provide their own answers.
Overall, the survey tool provided a valuable tool for collecting data on the practices, techniques, and benefits of reusing software used by participants, as well as their attitudesfor software reuse.

END SUMMARY

This study provides valuable insightsinto the current state of software reuse methods and techniques usedby software developers and managers. The results show that software reuse is an interesting approach to reduce development costs, improve quality and reduce development time. However, the study also identified several challenges and barriers to software reuse, such as lack of resources, lackof management support, and lack ofdocumentation.
Research has made a significant contribution to the field by providing a better understanding of the factors that influence software reuse and theattitudes of software developers and managers towards the use of software. software again. The results of this study can help organizations develop more effective strategies for implementing software reuse techniques and practices. Based on the results of this study, it is recommended that organizations invest in developing a culture that encourages software reuse, provides adequate resources and supports software reuse, and improves practices. documentation to facilitatesoftware reuse.
The study highlights the importance of software reuse as a valuable approach to software development and the need for organizations to develop effective strategies for implementing these techniques and practices. software reuse. With the right support and resources, softwarereuse can be a powerful tool for improving software development methods and driving innovation in the field

DOI: 10.48175/IJARSCT-12087