# Challenges of Adopting Software Reuse: Initial Results

Sajjad Mahmood
Department of Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, 31261, Saudi Arabia
smahmood@kfupm.edu.sa

Ali Al Zayer
Department of Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, 31261, Saudi Arabia
g200163270@kfupm.edu.sa

***Abstract*—A significant number of software development organizations have started adopting software reuse in order to facilitate achieving quality software, faster and at a lower cost. Software reuse helps organizations to leverage the benefits of systematic reuse with respect to architecture, design, source code and testing artifacts. One of the major issues is that many organizations endorse software reuse prior to understating and testing their readiness for the reuse based development processes. The objective of this paper is to identify challenges associated with software reuse in an organization. We have performed a Systematic Literature Review (SLR) by applying customized search strings derived from our research question. We have identified challenges, such as domain analysis and modeling, lack of reuse skills and knowledge, lack of management support, high reuse cost and lack of component repositories as key challenges in software reuse. Our ultimate aim is to develop a model in order to measure organizations' readiness for software reuse activities.**

***Keywords-systematic software reuse; challenges and barrier; systematic literature review; empirical studies.***

## I. INTRODUCTION

Software industry has been of the view that software artifacts can be reused to develop new applications. Software processes and artifacts have been reused since the early days of computing, as software reuse has the potential benefits to reduce development effort, reduce process risks and increase product quality and standard compliance [1]-[5].

Software development paradigms such as component-based development and service-oriented development encourage software reuse by supporting development based on reusable blocks of source code. In addition to source code, reuse of requirements patterns [6], system architecture, design and testing artifacts also have the potential to help achieve benefits associated with systematic reuse of software artifacts.

In addition to reuse benefits, numerous problems have been reported in the reuse initiatives [7][8]. Software reuse poses certain strategic challenges with respect to difficulty in maintaining a library of reusable artifacts and the cost of locating and adapting reusable artifacts [9]. Despite the importance of this problem, little research has been carried out to improve organizations' for adopting software reuse based development process. Understanding issues related to organizations readiness can help to ensure the successful outcome of projects.

In this paper, we aim at identifying the challenges via a systematic literature review that impact software reuse. Identifying these challenges will help software organizations in addressing them and be ready for systematic reuse. Our long term research objective is to develop a software reuse readiness framework to assist software developers in measuring and improving their software reuse readiness prior to adopting reuse driven development paradigms. To do this, we intend to address the research question as follows:

RQ: What are the challenges associated with adopting software reuse?

The rest of this paper is organized as follows: Section II presents motivation of the paper. Section III presents the background. In Section IV, we present the research methodology and Section V discusses the initial results. We conclude the paper and discuss future work in Section VI.

## II. MOTIVATION

Sherif and Vinze [15] presented a qualitative study based on a series of five cases to explore the individual and organizational barriers associated with the adoption of reuse. The study indicates that barriers to adoption of software reuse occur at both the individual and organization level. Mellarkod et al. [17] identified and assessed factors that influence developers' intention to reuse software assets. The study identified development of an infrastructure, self-efficacy and reuse experience as key factors that motivate individual developers to adopt software reuse. Similarly, Lucredio et al. [16] used survey based approach involving Brazilian organizations to identify some of the key factors in adopting an organization wide software reuse program.

 ISBN: 978-1-61208-367-4 

To the best of our knowledge, no explicit SLR-based empirical study has been conducted to identify the challenges associated with adopting software reuse in an organization. The initial results of this study are important for both practitioners and researchers to better understand the current state-of-the-art literature in the context of adopting systematic reuse. This study uncovers the challenges that need better management during adoption of software reuse in a given organization.

## III. BACKGROUND

Software developers have reused abstractions and process ranging from objects to commercial off the shelf components. Over the last couple of decades, a number of software reuse focused development paradigms have evolved, such as component-based development [18][19], software product lines [20], etc. Figure 1 presents a summary of reuse driven software development paradigms.

However, a significant number of software products developed using these paradigms have faced problems due to insufficient preparation and poor management both by reusable code developers and component integrators.

Understanding issues related to organization's software reuse will help ensure the successful outcome of projects. Hence, in this paper, we conduct a systematic literature review to identify challenges associated with adopting software reuse during development of software applications. The collected data focuses on challenges for effective management of software reuse driven development processes.

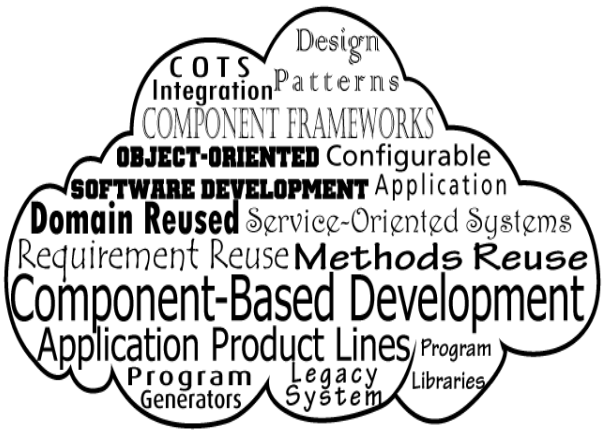


Figure 1. Software reuse paradigms.

## IV. RESEARCH METHODOLOGY

In this study, we aim at identifying the challenges associated with software reuse during development of a software application. In order to address the research question in hand, the Systematic Literature Review (SLR) [10] was used as a tool for data collection. SLR is a well-established empirical study technique for identifying, assessing and analyzing published studies to investigate a specific research question. Furthermore, the SLR approach provides a higher level of validity in its findings. A systematic review protocol was prepared to outline the review process. The main phases [11][12] in our research methodology are as follows:

- Construct search strategy and search relevant articles.
- Carryout the study selection process.
- Apply study quality evaluation.
- Extract and analyze the data.

In this paper, we focus on the challenges associated with adopting software reuse in an organization. In order to address that, we are going to address the following research question:

RQ1: What are the challenges of software reuse in developing projects?

The search strategy used is based on the following steps:

a) Derive the main terms from Population, Intervention and Outcome.
b) Find the synonyms and of the derived terms obtained in the first step.
c) Validate these terms in various academic databases
d) AND operator is used to connect main terms (if allowed depending on the academic databases).
e) OR operators, is used to connect synonyms and similar spellings. (If allowed academic databases).

Based on our search strategy, we have come up with the following search terms:

- POPULATION: software reuse
- INTERVENTION: project development challenges and barriers.
- OUTCOME OF RELEVANCE: challenges and barriers in project development of software reuse.
- EXPERIMENTAL DESIGN: SLRs, case studies, empirical and theoretical studies, researchers and expert opinions.

After testing our main terms in several academic databases, the most relevance terms used to the topic are as follows:

 ISBN: 978-1-61208-367-4 

- Software reuse: "Software reuse" OR "architecture reuse" OR "component reuse" OR "reuse environment" OR "product line based reuse".
- Software Development: "Software Development" OR "Software Implementation" OR "Software Coding".
- Challenges: "Challenges" OR "problems" OR "difficulties" OR "complications" OR "obstacles" OR "barriers" OR "risks".

The final search string has been designed after trail search, which is as follows:
(("Challenges" OR "problems" OR "difficulties" OR "complications" OR "obstacles" OR "barriers" OR "risks") AND
("Software reuse" OR "Architecture Reuse" OR "Component Reuse" OR "Code Reuse" OR "Product Line Based Reuse" OR "Software Reusability")
AND
("Software Development" OR "Software Implementation" OR "Software Application"))

Our focus was based on the following digital library:

- IEEE Explore. (http://ieeexplore.ieee.org)

The following inclusion criteria were used:

- Conference proceedings, magazines and journals published after 1980.
- Papers published in any of the primary or secondary resources mentioned previously.
- Studies which focus on answering our research question.

The following exclusion criteria were used:

- Duplicated or repeated studies.
- Manuscripts written in a language other than English language.
- Technical reports and white papers.
- Graduate projects, Master theses and PhD dissertations.
- Textbooks, whether in print or electronic.

For any paper to pass the initial phase, a quality assessment was done and four quality criteria were defined, as shown in Table I. We have selected 36 articles which meet the inclusion and quality criteria. Next, we extracted data from the final selected papers to address our research question. Table II presents the data extracted from the selected articles.

TABLE I. QUALITY ASSESSMENT

| Criteria | Notes |
|---|---|
| Are the findings and results clearly stated in the paper? | Yes =1<br>No =0 |
| Is there any empirical evidence on the findings? | Yes =1<br>No =0 |
| Are the arguments well- presented and justified? | Yes =1<br>No =0 |
| Is the paper well referenced? | Yes =1 |

TABLE II. DATA EXTRACTION FORM

| Extracted Data | |
|---|---|
| ▪ Publication Name | ▪ Author(s) |
| ▪ Publication Date | ▪ Geographical Location |
| ▪ Reference Type | ▪ Publication Type |
| ▪ Publisher | ▪ Challenges |

## V. INITIAL RESULTS AND DISCUSSION

In this paper, we report our initial results based on IEEE electronic database. Table III shows the total number of results retrieved from IEEE electronic database. After initial round of screening by reading the title and abstract, 73 articles were selected. Next, full text of the 73 articles was read and 36 primary studies were finally selected.

TABLE III. SEARCH EXECUTION

| Resource | Total Results | Initial Selection | Final Selection |
|---|---|---|---|
| IEEExplore | 1395 | 73 | 36 |

In our SLR, we have classified the papers found into seven study strategies, which are commonly used in the empirical software engineering, as shown in Table IV. The majority of the selected articles used case study research method.

TABLE IV. STUDY STRATEGIES USED

| Study Type | Count |
|---|---|
| Case Studies | 24 |
| Interviews | 1 |
| Experience Report | 3 |
| Systematic Literature Reviews | 0 |
| Survey/Questionnaire | 6 |
| Delphi Study | 2 |
| **Total** | **36** |

Table V shows the country-based analysis for the papers included in the SLR study. Twenty studies were carried out in USA, three each in China and Spain, and two in Canada and United Kingdom, respectively.

 ISBN: 978-1-61208-367-4 

TABLE V. STUDY COUNTRIES

| Country | Count | Country | Count |
|---|---|---|---|
| Canada | 2 | Saudi Arabia | 1 |
| China | 3 | Spain | 3 |
| Germany | 1 | Italy | 1 |
| Japan | 1 | United Kingdom | 2 |
| Malaysia | 2 | USA | 20 |

To answer our research question, the data were extracted and synthesized from the 36 finally selected studies. We have identified eight challenges for systematic reuse during development of software applications, as shown in Table VI.

TABLE VI. LIST OF CHALLENGES

| Challenges | Freq. (n=36) | % |
|---|---|---|
| Domain analysis and modeling | 29 | 83 |
| Lack of reuse skills and knowledge | 27 | 75 |
| Lack of management support | 12 | 33 |
| High reuse cost | 12 | 33 |
| Lack of component storage | 9 | 25 |
| Lack of documentation | 7 | 20 |
| Lack of proper IT infrastructure | 4 | 11 |
| Lack of team awareness | 2 | 6 |

In our study, the most common software reuse challenge is ‘domain analysis and modeling’ (83%). The fact that in software reuse-driven development, practitioners need to carry out detailed domain analysis and modeling to search and select suitable reusable components. The second highest ranked challenge is ‘lack of reuse skill and knowledge’. For example, Gonzalez [13] identified that the users of object oriented software components face cognitive gap in knowledge and often face difficulty in understanding the vocabulary used in component documentations.

About 33% of the articles in our study described ‘lack of management support’ and ‘high reuse cost’ as another major challenges. ‘Lack of component storage’ has been mentioned in about 25% of the articles. The main reason for this challenge is lack of standard reuse environments [9] and repositories. Furthermore, ‘lack of documentation’ has also been an important challenge in reuse based development. For example, Mahmood and Khan [14] empirical study indicates that the lack of good component documentation presents a risk for use of reusable components.

## VI. CONCLUSION AND FUTURE WORK

Systematic software reuse facilitates achieving quality software, faster and at a lower cost. Despite the potential benefits associated with software reuse, software organizations struggle with adopting reusable components during development of a software application. Due to availability of a number of reuse-driven development paradigms and the increasing trend of adopting reusable components, we aim to discover challenges associated with systematic reuse.

In our initial results, the frequently mentioned challenges for systematic reuse are domain analysis and modeling, lack of reuse skills and knowledge, lack of management support, high reuse cost and lack of component repositories.

As part of future work, we plan to carry out SLR in other major databases, namely, ACM, Science Direct, Springer Link, and John Wiley. We also plan to identify solutions, in the form of best practices, for each of the frequently mentioned challenge. We intend to find the best practices by carrying out an empirical study with software industry.

## ACKNOWLEDGMENT

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for its continuous support of research.

## REFERENCES

[1] W. B. Frakes, "Software reuse research: status and future", IEEE Trans. Softw. Eng., vol. 31, pp. 529-536, 2005.

[2] W. B. Frakes and S. Isoda, "Success factors of systematic reuse. Software", IEEE Software, vol. 11, pp. 14-19, 1994.

[3] A. Mili, S. Yacoub and H. Mili, "Toward an engineering discipline of software reuse", IEEE Software, vol. 16, pp. 22-31, 1999.

[4] L. Prechelt, B. Unger, W. F. Tichy and P. Brossler, "A controlled experiment in maintenance comparing design patterns to simpler solution", IEEE Trans. Softw. Eng., vol. 27, pp. 1134-1144, 2001.

[5] S. Watanabe, "Professionalism through OO and reuse", IEEE Software, vol. 14, pp. 26, 1997.

[6] X. Franch, "Software requirement patterns", 35th International Software Engineering Conference, pp. 1499-1501, 2013.

[7] D. Card, and E. Comer, "Why do so many reuse programs fail", IEEE Software, pp. 114-115, 1994.

[8] M. Jha, and L. O'Brien, "A comparison of software reuse in software development communities", 5th Malaysian Software Engineering Conference, pp. 313-318, 2011.

[9] S. Mahmood, M. Ahmed, and M. Alshayeb, "Reuse envirnoments for software artifacts: analysis framework", 12th IEEE/ACIS International Conference on Computer and Information Sciences, pp. 35-40, 2013.

[10] B. Kitchenham, and C. Charters, "Guidelines for performing systematic literature reviews in software engineering", Keele University and Durham University Joint Report - EBSE 2007-001, EBSE 2007-001, 2007.

[11] S.U. Khan, M. Niazi, and R. Ahmad, "Factors influencing clients in the selection of offshore software outsourcing vendors: an exploratory study using a systematic literature review", Journal of Systems and Software, vol. 84, pp. 686-699, 2011.

[12] M. Niazi, S. Mahmood, M. Alshayeb, A. M. Qureshi, K. Faisal and N. Cerpa, "Towards identifying the factors for project management success in global software development:

 ISBN: 978-1-61208-367-4 

Initial results", The Eigth International Conference on Software Engineering Advances, pp. 285-290, 2013.

[13] P. A. Gonzalez, "Applying knowledge modelling and case-based reasoning to software reuse", IEEE Software Proc., vol. 147, pp. 169-177, 2000.

[14] S. Mahmood, and A. Khan, "An industrial study on the importance of software component documentation: A system integrator's perspective", Information Processing Letters, vol. 111, pp. 583-590, 2011.

[15] Karma Sherif and Ajay Vinze, "Barriers to adoption of software reuse: A qualitative study", Information and Management, vol. 41, pp. 159-175, 2003.

[16] Lucredio et al., "Software reuse: the Brazilian industry scenario", Journal of Systems and Software, vol. 81, pp. 996 - 1013, 2008.

[17] V. Mellarkod, R. Appan, D. R. Jones and K. Sherif, "A multi-level analysis of factors affecting software developers' intention to reuse software assests: an empirical investigation", Information and Management, vol. 44, pp. 613 - 625, 2007.

[18] M.A. Khan and S. Mahmood, "A graph based requirements clustering approach for component selection", Advances in Engineering Software, vol. 54, pp. 1-16, 2012.

[19] S. Mahmood, R. Lai and Y.S. Kim, "Survey of component based software specification", IET Software, vol.1, pp. 57-66, 2007.

[20] F.V. Linden, K. Schmid and E. Rommes, "Software product lines in action: The best industrial practice in product line engineering', Springer, 2007.

 ISBN: 978-1-61208-367-4 