A DECADE OF SOFTWARE REUSE: A RESEARCH DIRECTION

[1]Fatin Izatul Nadia Mohd Zulkurnain, [2]Suryani Ismail,[3] Masita @ Masila Abdul Jalil, [4]Fatihah Mohd

[1] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu, Malaysia. Email: fatin.izatulnadia@gmail.com
[2] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu, Malaysia. Email: sue@umt.edu.my
[3]Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu, Malaysia. Email: masita@umt.edu.my
[4]Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan. Email: mpfatihah@gmail.com

Abstract

Software reuse has been one of the most interesting research topics for more than forty years ago. Throughout that time, the interest shown in this area is growing rapidly and contributed to many results being published. There is a broad knowledge and experiences enclosed in conferences and journals focusing on this field. Hence, there are a number of issues that need to be further addressed. The major focus of this study is to investigate the current trends in software reuse research through a detailed literature review. A detailed literature review study is done based on various sources such as Web of Science (WoS), SCOPUS*, ScienceDirect, IEEE Xplore and ACM Digital Library (ACM DL) which has been selected from January 2009 until December 2019. Using the available resources, this study highlights two aspects of software reuse which are the intensity of research activity and the research topics. The mentioned trends indicate that software reuse is still one of the interesting fields that call for further research.*

***Keywords**: Component based software development (CBSD), component based software engineering (CBSE), research direction, software reuse.*

I. INTRODUCTION

The term software reuse was introduced by [1] at the first International Conference on Software Engineering (ICSE) in 1968. Throughout all these years, researchers and practitioners have been implementing software reuse approaches to increase the productivity and cost savings in developing new software. A steady incremental growth in publications throughout these years indicates that software reuse remains to be an interesting area to investigate in software engineering research.

Software reuse can be interpreted in various perspectives, according to the users' point of view. In 1992, [2] stated that software reuse can be described as the software development process in creating software systems by using existing software components instead of developing software system from scratch. According to [3], software reuse can be considered as the usage of existing software artifacts that subsequently improves the software quality. Furthermore, [4] describe software reuse as the use of available software artifacts to create new software.

Generally, the term software reuse is related to a situation where some software is implemented in more than one project. Here software is interpreted loosely as one or more items are considered as part of the software engineering process of an organization's guideline that produces software product. Hence, software could either be referred to source code or any other related products of the software lifecycle like the requirements, designs, test plans, test packages or its documentation [5].

Several studies have identified that by building new systems from existing and pre-tested components, developers will save more on the cost of designing, writing and testing new codes. There are companies that have embraced its approaches to their development method have claimed beneficial enhancement in development productivity, quality and maximize the profits [6]-[14].

Nowadays, software reuse practice is widely growing among software development community because it contributes in increasing the software quality and production, improving resources used as well as the maintenance. Other than that, software reuse approach may reduce risk, time and costs for the development of new software systems [15]. Despite that, after many years of development, the question emerges whether the previous research in this area has achieved its goals, or if there are any overlooked issues for further investigation in order to know the current trends in software reuse research [16], [17].

This paper emphasizes on the arising issues in software reuse, particularly for open source software development in the period of January 2009 until December 2019. The goal is to discover the recent research trends and scopes for further improvement in software reuse.

The rest of the paper is organized as follows: Section 2 describes the related work in the area of interest and Section 3 presents the research methodology involved, whereas, Section 4 shows the results and discussion of the main findings of this study. Finally, Section 5 presents the conclusions.

II. RELATED WORK

There is a considerable amount of literatures that have been published on software reuse since it was first introduced in 1968. A study by [16], provided a thorough literature review on component based software engineering (CBSE). The study aimed to investigate the five dimensions of CBSE which focused on the goals, research themes, application domains, the intensity of the investigation and research methods used. The authors analyzed 1231 studies published from 1984 until 2012. The authors found that enhancing productivity, reducing expenses and enhancing quality are often cited CBSE goals. In addition to the analysis, the study discovered open topics and acknowledge the domains that needed further research.

Besides that, the most recent published work is from [18]. The authors conducted a tertiary study based on systematic secondary studies published until July 2018. Then, the authors identified 4423 studies relevant to software reuse, from which 3102 were narrowed down by selecting criteria and quality assessment to come out with 56 related studies. After that, the authors pointed out 30 current research topics and 127 upcoming research proposals, classified them into three categories which are software product lines (SPL), other reuse approaches and general reuse topics. The authors emphasized that recurring reported themes are requirements and testing in life cycle stages for SPL and systematic reuse for decision making in general reuse. The authors also highlighted upcoming research proposals such as requirements, evolution and variability management for SPL and systematic reuse for decision making in general reuse.

The review methods used in both published research are inspired by the guidelines for performing systematic mapping studies which are introduced by [19]. This mapping study process is one of the most highly cited in review methods research [20]-[25].

III. RESEARCH METHODOLOGY

The research methodology implemented in this study is based on [19]. The adopted mapping process as illustrated in Fig. 1 includes ten phases which are defining research questions, scope review, conduct study, all papers, paper screening, relevant papers, search more specific keywords, classification scheme, data extraction and systematic map. Initially, the definition of research questions indicates the goal and the scope of this study. This study is driven by formulating research questions which further narrowed the scope of this study. The research is initiated by identifying all software reuse related papers by using software reuse keyword.

Then, the publication period is set for January 2009 until December 2019 to select more specific papers. Next, a screening process is conducted by specifying more specific keywords to select related papers and classified them according to their research topics. After that, the classification scheme is utilized to extract all the data needed to address the research questions. Further explanation of each phase is explained in the following subsections.

A. Research Questions

The major issue that motivates this study and indicates the goal of this study is: "How to describe the contributions of software reuse in the last decade?". In order to address this inquiry, two research questions were formulated, focusing on different prospects of the software reuse field. The research questions that have been formulated are described as follows:

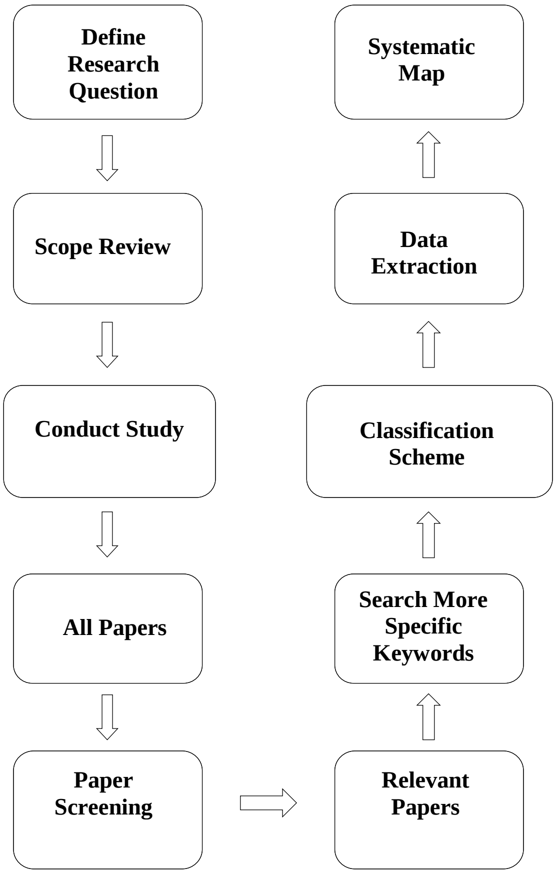

Fig. 1. Mapping study process

- RQ1. What is the regularity of research interest in software reuse?
  This question is focused towards the quantity of studies and selected journals that involved in software reuse publication.
- RQ2. Which are the most frequently investigated software reuse research topics and how they have changed over time?
  This question investigates research topics raised by the software reuse studies. The research topics mentioned are further explored and used to identify research gaps.

B. Conduct Research

Following the principles reported in [24] and supported by [25], this study carried out automatically and manual search on selected journals associated with a software reuse field. To deflect from bias in the automated searching, four complementary electronic data sources (EDS) are incorporated which are widely-known among researchers and academics as proposed by [26]. They are Web of Science (WoS), SCOPUS, IEEE Xplore and ACM Digital Library (ACM DL). The first two identified as the biggest general indexing services which consisting of papers published by ACM, IEEE, Elsevier, Springer and

Wiley. Meanwhile, the last two comprised of the most essential journals and conferences in software engineering [18], [27], [28].

Table I shows the journals along with their h-indices that have been utilized in the study. These journals represent the selected publications that are used in this study.

Table- I: List of journals and their h-indices on 19th December 2019.

Journal	h-index
IEEE Transactions on Software Engineering	151
ACM Computing Surveys	132
Journal of Systems and Software	94
Information and Software Technology	88
Empirical Software Engineering	59
Computer Standards & Interfaces	58
Journal of Theoretical and Applied Information Technology	23
Journal of Software Engineering and its Applications	17
International Journal of Open Source Software and Processes	7
CRC Reports	3

Table II depicts the most cited papers with their h-indices that have been utilized in this study. These papers represent the selected publications used in this study [1], [3], [16], [17], [19], [22], [23], [26], [27], [29]–[31], [4], [32]–[41], [6], [42]–[45], [7]–[9], [12]–[14].

Table- II: List of the most cited papers and their h-indices on 23rd December 2019.

Title	h-index
Systematic Mapping Studies in Software Engineering	1948
Software Reuse	1799
Guidelines for Snowballing in Systematic Literature Studies and A Replication in Software Engineering	909
Software Reuse Research: Status and Future	853
Mass-Produced Software Components	790
Software Reuse: Metrics and Models	491
Empirical Evidence in Global Software Engineering: A Systematic Review	363
A Systematic Review of Systematic Review Process Research in Software Engineering	358
A Large Scale Empirical Study on Software Reuse in Mobile Apps	115
Code Reuse in Open Source Software Development: Quantitative Evidence, Drivers, and Impediments	103
Systematizing Pragmatic Software Reuse	91
Twenty-Eight Years of Component-Based Software Engineering	84

Towards an Evidence-Based Understanding of Electronic Data Sources	73
On the Extent and Nature of Software Reuse in Open Source Java Projects	72
A Systematic Review of Productivity Factors in Software Development	69
A Survey on Software Reuse Processes	50
Reverse Engineering Reusable Software Components from Object-Oriented APIs	34
Components–the Past, the Present, and the Future	27
Designing Code Level Reusable Software Components	26
Software Reuse: Developers' Experiences and Perceptions	26
A Pragmatic Approach to Software Reuse	25
Evaluating Strategies for Study Selection in Systematic Literature Studies	24
Software Reuse and Plagiarism: A Code of Practice	23
Software Reuse versus Stability: Evaluating Advanced Programming Techniques	20
Software Reuse in Practice	15
Breaking the Boundaries for Software Component Reuse Technology	14
Software Reuse in Open Source: A Case Study	13
Review of Software Reuse Processes	11
A Study of Factors Affecting the Design and Use of Reusable Components	10
Reliability of Search in Systematic Reviews: Towards a Quality Assessment Framework for the Automated-Search Strategy	8
Software Reuse Methods to Improve Technological Infrastructure for e-Science	7
A Comparison of Software Reuse in Software Development Communities	7
A Framework for Software Reuse and Research Challenges	7
REACT – A Process for Improving Open-Source Software Reuse	1
Trends in Software Reuse Research : A Tertiary Study	1

On the other hand, Table III exhibits the list of associated authors, their total citations and their h-indices throughout their career as researchers in software engineering field of research.

Table- III: List of associated authors, their total citations and their h-indices on 23rd December 2019.

Authors	**Citation**	**h-index**

	s	
1. Kitchenham, Barbara	32,718	76
2. Randell, Brian	19,939	50
3. Wohlin, Claes	19,563	57
4. Runeson, Per - P. Runeson	17,246	42
5. Host, Martin - Martin Höst	14,195	30
6. Regnell, Bjorn	13,972	42
7. Kang, Kyo	12,212	34
8. Brereton, Pearl	10,288	38
9. Budgen, David	9,794	36
10. Ali Babar, Muhammad	8,137	47
11. Frakes, William	8,046	25
12. Turner, Mark	7,131	24
13. Feldt, Robert	6,313	34
14. Crnkovic, Ivica	6,198	33
15. Petersen, Kai	5,929	32
16. Gorschek, Tony	5,108	41
17. Han, Jun - J. Han	4,749	34
18. Wagner, Stefan	4,027	32
19. De Almeida, Eduardo Santana	3,438	29
20. Mattsson, Michael	3,114	17
21. Rainer, Austen	3,076	19
22. De Lemos Meira, Silvio Romero	2,958	27
23. Meira, Silvio Romero De Lemos	2,958	27
24. Boldyreff, Cornelia	2,406	22
25. Šmite, Darja	2,128	26
26. McIlroy, M. Douglas	1,987	114
27. Stol, Klaas-Jan	1,851	22
28. Krueger, Charles W.	1,724	21
29. Capiluppi, Andrea	1,721	23
30. Schneider, Jean-Guy	1,535	23
31. Garcia, Vinicius Cardoso	1,434	19
32. Agresti, William W.	1,390	19
33. Chen, Lianping	1,359	14
34. Naur, Peter	1,293	18
35. Lucrédio, Daniel	1,184	19
36. Benitti, Fabiane B. V.	1,123	12
37. Usman, Muhammad	1,065	22
38. Alvaro, Alexandre	1,045	18
39. Mujtaba, Shahid	1,016	4
40. Silveira Neto, Paulo Anselmo Da Mota	926	13
41. Ali, Nauman Bin	608	10
42. Zhang, He	604	17

43. Ruhe, Melanie	491	7
44. Govardhan, Aliseri	471	15
45. Cavalcanti, Yguaratã Cerqueira	409	10
46. Antovski, Ljupcho	311	9
47. Premchand, Parvataneni	289	12
48. Vale, Tassio	226	7
49. Terry, Carol	208	2
50. Matalonga, Santiago	183	8
51. Jatain, Aman	162	6
52. Jalender, B.	94	6
53. Barros-Justo, José L.	68	6
54. Joshi, Salil	36	4
55. Imeri, Florinda	31	4
56. Buxton, J.	15	1
57. Keswani, Raman	4	2

C. Selection Criteria

Each paper was evaluated and selected based on the selection scope allocated for this study, in order to validate whether it was convenient to deal with the research questions. The selection scope is narrowed down into two selection criteria which are inclusion and exclusion criteria. For the inclusion criteria, the previous studies which are published in the chosen journals that concentrate on any software reuse issues while, exclusion criteria involve the grey area in literature such as books, technical reports and short papers (less than five pages).

D. Screening Papers

The screening process involved filtering the papers to more specific keywords of selected research topics in this study such as functionality, quality, interaction, methodology, management, evolutions and tools. The publication period is then specified in January 2009 until December 2019 to retrieve more specific papers. The key wording technique is applied to the titles and abstracts of the selected paper to produce classification scheme. Moreover, this classification scheme is used to extract data from the selected papers with the purpose of incorporating all the collected data from the mapping process.

IV. RESULTS AND DISCUSSION

The following section represents a summary of outcomes of the research questions by analyzing and evaluating the results of this study and also a discussion on the most significant aspects related to each research question in this study. The outcomes are grouped according to each research question.

A. RQ1. What is the regularity of research interest in software reuse?

Three aspects were explored to answer this question. They are the advancement of publications throughout the years, the selected journals and the citations. A total of 1881 publications related to software reuse extends across the years from January 2009 until December 2019. Fig. 2 represents the allocation of these publications over a decade time span.

Overall, it can be said that throughout the study period, there was a reduction amount of published papers in software reuse research. With reference to the selected journals which publish software reuse papers (see Fig. 3), Web of Science and IEEE Xplore were the lead contributors in this study. They had almost double the number of publications compared with the others, contributing 721 and 559 publications respectively.

The result clearly shows a significant declining interest in software reuse as the years go by but the number of publications is still stable and above 100 publications by the end of year 2019. As the reviewing and publishing procedure become more thorough these days, it has contributed a lot to the declining of publication rate per year as there is already a broad body of knowledge. In addition, software reuse has been incorporated into other means like object oriented development, service oriented development, software product lines, model based engineering and cloud computing [16].

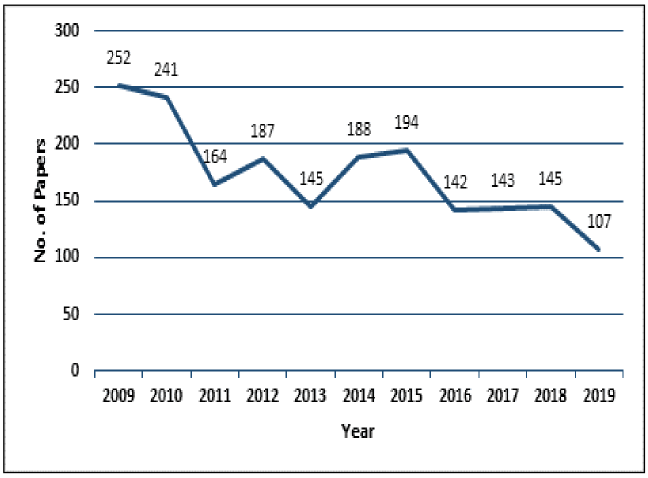

Fig. 2. Number of publications within the time span specified in this study

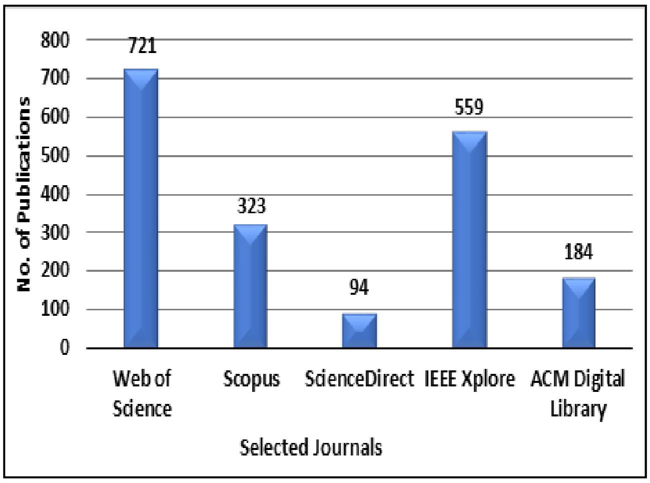

Fig. 3. Selected journals

B. RQ2. Which recurring research topics in software reuse that has been investigated and how they evolved throughout the decade?

This inquiry pointed out frequently focused software reuse theme and its advancement throughout the decade. A classification which comprised of six major research themes in CBSE proposed by [17] is adopted in this study. It has been modified to suit the focus area of this study, which is software reuse. The six main research topics are functionality, quality, interaction, methodology, management, evolution and tools.

Quality was the most recurring topics in 292 studies, followed by evolution and tools (223 studies), management (193), functionality (129) and methodology (107). Interaction was the least topic, with 52 studies. Fig. 4 illustrates a summary of this distribution. With regards to the transformation of research topics for a decade, it can be seen that quality is the most explored topic compared to the other research topics. This is because quality is one of the main objectives of applying software reuse in software development. The results point out that the interests of most researchers focus on the topics such as quality, evolution and tools as well as the management. However, the interest for the remaining research topics continues to grow over the years.

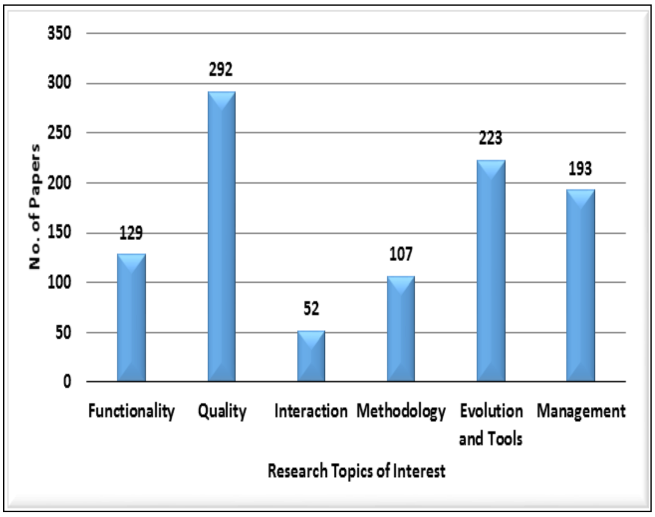

Fig. 4. Addressed research topics

V. CONCLUSIONS

A detailed literature review has been carried out with the focus on analyzing trends in software reuse research. The study is performed based on several sources like WoS, SCOPUS, ScienceDirect, IEEE Xplore and ACM DL. The finding of this study shows a decreasing pattern in the software reuse research area throughout the year on January 2009 until December 2019. On top of that, this study has shown that quality is recorded at 292, evolution and tools at 223, management at 193, functionality at 129, methodology at 107 and interaction at 52.

There is a vast opportunity for future research on functionality, methodology and interaction. There is also interesting topics like code clone detection, component libraries, security, reliability and safety in software reuse research that need to be addressed. As future work, a further investigation will be conducted to determine which are the main objectives of applying software reuse, which application domains has software reuse been applied and to extend the topics on code clone detection, component libraries, security, reliability and safety in software reuse research. In conclusion, this study can serve as the foundation to identify gaps in software reuse research and to discover areas where further research of the existing studies should be carried out.

ACKNOWLEDGMENT

This research is fully supported by FRGS grant, vot number 59504. The authors fully acknowledged Ministry of Higher Education (MOHE) and Universiti Malaysia Terengganu for the approved fund, which makes this important research viable and effective.

REFERENCES

1. M. D. McIlroy, J. Buxton, P. Naur, and B. Randell, "Mass-Produced Software Components," in *Proceedings of the 1st International Conference on Software Engineering, Garmisch Pattenkirchen, Germany*, 1968, pp. 88–98.
2. C. W. Krueger, "Software Reuse," *ACM Comput. Surv.*, vol. 24, no. 2, pp. 131–183, 1992.
3. W. Frakes and C. Terry, "Software Reuse: Metrics and Models," *ACM Comput. Surv.*, vol. 28, no. 2, pp. 415–435, 1996.
4. B. Jalender, A. Govardhan, and P. Premchand, "Designing Code Level Reusable Software Components," *Int. J. Softw. Eng. Appl.*, vol. 3, no. 1, pp. 219–229, 2012.
5. R. J. Leach, *Software Reuse: Methods, Models and Costs*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1996.
6. W. W. Agresti, "Software Reuse: Developers' Experiences and Perceptions," *J. Softw. Eng. its Appl.*, vol. 04, no. 01, pp. 48–58, 2012.
7. A. Capiluppi, K.-J. Stol, and C. Boldyreff, "Software Reuse in Open Source: A Case Study," *Int. J. Open Source Softw. Process.*, vol. 3, no. 3, pp. 10–35, 2011.
8. B. Jalender, A. Govardhan, and P. Premchand, "A Pragmatic Approach to Software Reuse," *J. Theor. Appl. Inf. Technol.*, vol. 14, no. 2, pp. 87–96, 2010.
9. R. Keswani, S. Joshi, and A. Jatain, "Software Reuse in Practice," pp. 159–162, 2014.
10. J. Sametinger, *Software Engineering with Reusable Components*. 2013.
11. I. Sommerville, *Software Engineering*, 10th ed. Harlow: Pearson Education, 2016.
12. S. Wagner and M. Ruhe, "A Systematic Review of Productivity Factors in Software Development," 2018.
13. L. Antovski and F. Imeri, "Review of Software Reuse Processes," *Int. J. Comput. Sci. Issues*, vol. 10, no. 6, pp. 83–88, 2013.
14. W. B. Frakes and K. Kang, "Software Reuse Research: Status and Future," *IEEE Trans. Softw. Eng.*, vol. 31, no. 7, pp. 529–536, 2005.
15. E. S. De Almeida, A. Alvaro, D. Lucrédio, V. C. Garcia, and S. R. de Lemos Meira, "A Survey on Software Reuse Processes," in *IRI-2005 IEEE International Conference on Information Reuse and Integration, Conf, 2005.*, 2005, pp. 66–71.
16. T. Vale, I. Crnkovic, E. S. De Almeida, P. A. D. M. Silveira Neto, Y. C. Cavalcanti, and S. R. D. L. Meira, "Twenty-Eight Years of Component-Based Software Engineering," *J. Syst. Softw.*, vol. 111, pp. 128–148, 2016.
17. J.-G. Schneider and J. Han, "Components–the Past, the Present, and the Future," in *Workshop on Component-Oriented Programming*, 2004.
18. J. L. Barros-Justo, F. B. V Benitti, and S. Matalonga, "Trends in Software Reuse Research: A Tertiary Study," *Comput. Stand. Interfaces*, vol. 66, no. December 2018, p. 103352, 2019.
19. K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, 2008, vol. 8, pp. 68–77.
20. P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, 1st ed. Wiley Publishing, 2012.
21. C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and A Replication in Software Engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, p. 38.
22. B. Kitchenham and P. Brereton, "A Systematic Review of Systematic Review Process Research in Software Engineering," *Inf. Softw. Technol.*, vol. 55, no. 12, pp. 2049–2075, 2013.
23. D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical Evidence in Global Software Engineering: A Systematic Review," *Empir. Softw. Eng.*, vol. 15, no. 1, pp. 91–118, 2010.
24. B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, vol. 4. CRC press, 2015.
25. N. Bin Ali and M. Usman, "Reliability of Search in Systematic Reviews: Towards a Quality Assessment Framework for the Automated-Search Strategy," *Inf. Softw. Technol.*, vol. 99, pp. 133–147, 2018.
26. N. Bin Ali and K. Petersen, "Evaluating Strategies for Study Selection in Systematic Literature Studies," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2014, pp. 45:1--45:4.
27. L. Chen, M. Ali Babar, and H. Zhang, "Towards an Evidence-Based Understanding of Electronic Data Sources," 2010.
28. M. Turner, "Digital Libraries and Search Engines for Software Engineering Research: An Overview," *Keele Univ. UK*, 2010.
29. N. Bin Ali and M. Usman, "Reliability of Search in Systematic Reviews: Towards A Quality Assessment

Framework for the Automated-Search Strategy," *Inf. Softw. Technol.*, vol. 99, pp. 133–147, 2018.

30. R. Anguswamy, "A Study of Factors Affecting the Design and Use of Reusable Components," pp. 2–9, 2013.
31. J. L. Barros-Justo, F. B. V. Benitti, and S. Matalonga, "Trends in Software Reuse Research: A Tertiary Study," *Comput. Stand. Interfaces*, Apr. 2019.
32. F. Dantas and A. Garcia, "Software Reuse versus Stability: Evaluating Advanced Programming Techniques," *Proc. - 24th Brazilian Symp. Softw. Eng. SBES 2010*, pp. 40–49, 2010.
33. E. S. De Almeida, A. Alvaro, D. Lucrédio, V. C. Garcia, and S. R. De Lemos Meira, "A Survey on Software Reuse Processes," in *Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration, IRI - 2005,* 2005.
34. J. P. Gibson, "Software Reuse and Plagiarism: A Code of Practice," in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 2009.
35. L. Heinemann, F. Deissenboeck, M. Gleirscher, B. Hummel, and M. Irlbeck, "On the Extent and Nature of Software Reuse in Open Source Java Projects," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6727 LNCS, pp. 207–222, 2011.
36. R. Holmes and R. J. Walker, "Systematizing Pragmatic Software Reuse," *ACM Trans. Softw. Eng. Methodol.*, 2012.
37. B. Jalender, D. A. Govardhan, and D. P. Premchand, "Breaking the Boundaries for Software Component Reuse Technology," *Int. J. Comput. Appl.*, vol. 13, no. 6, pp. 37–41, 2011.
38. M. Jha and L. O'Brien, "A Comparison of Software Reuse in Software Development Communities," *2011 5th Malaysian Conf. Softw. Eng. MySEC 2011*, pp. 313–318, 2011.
39. C. W. Krueger, "Software Reuse," *ACM Comput. Surv.*, 1992.
40. A. Lampropoulos, A. Ampatzoglou, S. Bibi, A. Chatzigeorgiou, and I. Stamelos, "REACT - A Process for Improving Open-Source Software Reuse," no. September, 2018.
41. J. J. Marshall, R. R. Downs, and C. A. Mattmann, "Software Reuse Methods to Improve Technological Infrastructure for e-Science," 2019.
42. I. J. Mojica, B. Adams, M. Nagappan, S. Dienst, T. Berger, and A. E. Hassan, "A Large-Scale Empirical Study on Software Reuse in Mobile Apps," *IEEE Softw.*, 2014.
43. A. Shatnawi, A.-D. Seriai, H. Sahraoui, and Z. Al-Shara, "Reverse Engineering Reusable Software Components from Object-Oriented APIs," *J. Syst. Softw.*, pp. 442–460, 2017.
44. M. Sojer and J. Henkel, "Code Reuse in Open Source Software Development: Quantitative Evidence, Drivers, and Impediments," *J. Assoc. Inf. Syst.*, vol. 11, no. 12, pp. 868–901, 2010.
45. S. K. Soora, "A Framework for Software Reuse and Research Challenges," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 10, pp. 441–448, 2014.