A Controlled Experiment on Reusability Component Evaluation: Demographics Results

Suryani Ismail, Fatihah Mohd, Masita, Masila Abdul Jalil

Abstract*: Software component reuse (SCR) is considered as an important solution to software engineering problems. There is a wide benefit of SCR to improve the productivity and the quality of software development (SD). Many organizations have benefited from using reusable components in reducing the time and cost of software development. Our objective is to evaluate and validate the reliability of the component reusability for component based software development (CBSD). To achieve this objective, we systematically designed a controlled experiment using human subjects among 20 experts working in SD. The survey, conducted contains 2 sections. Section A is to be answered by respondents before the experimental tasks begin, while section B contains the results of user evaluation and their experience of using the given Java components. In this study, a finding of section A is presented. It contains mostly about questions about the user background of software engineering processes. It is targeted to collect some information regarding the respondent's background such as: working experience and some aspects related to their familiarity of software engineering tasks. Among the findings are the followings: (i) A total of 20 respondents is the expert in software engineering: system analyst 5%, lecturer 50%, and postgraduate student considered as a researcher is 45%. (ii) The majority of the experts were working more than 10 - 20 years (45%), followed by less than 5 years (30%), 5 to 10 years (20%) and 21 to 30 years (5%). (iii) In term of working experience, most of the experts (39%) were average and substantial in their work experience, and only 7% of the experts had none experience in their job. The results of our survey will be of interest to software development professionals. It will benefit as a guide for users to develop the new component for reuse and also help user to choose the components for reuse in a new software development.

***Index Terms*: Component based software development (CBSD), controlled experiment, software component reuse (SCR), software engineering.**

I. INTRODUCTION

Component Based Software Development (CBSD) is a procedure of software development that focuses the design and building systems using design of reusable software components [1]. The developers may use existing software components with a small or without any modification to reduce the development times and mostly focuses on objects that can be simply reusable and relationships among the objects [2]. In software reuse (SR) formerly written software either in the form of specifications, designs or codes is used [3]. Nowadays, software reuse is widely used by software development since it's capable to reduce risk, time and costs, higher production and quality, improve resources uses and maintenance [4]. Software engineers apply software reuse in all phases of software development, particularly in design, implementation and testing. In academic, software reuse and component reuse assist students in learning programming and system development [5].

Software reuse is defined as the use of existing software artifacts to build new software [6]. Software reuse could also be as the software development procedure for producing software systems using present software artifacts rather than the construction software system from scratch [7]. Another study, defined software reuse is the reuse of executable codes such as code fragments, subroutines or modules as a component reuse [8]. Component evaluation approaches (CEA) are mostly applied to evaluating the component reuse; in order to recognize the component reusability according to their quality, originality, and reusability. The evaluation is also to measure the reusability of components in order to realize the reuse of component effectively and to identify the best components in terms of their reusability. Component evaluation has two kinds of method: (i) component certification that is done by an independent player to deliver a trustworthy valuation of the component, and (ii) component evaluation that is implemented by a system development association [9].

In a field of software engineering (SE), a researcher applies a controlled experiment using human participant as an empirical method to evaluate and validate the research finding. This experiment is used to compare a type of techniques, methods, and working procedures. Researchers use the experiment because they can plan and design the study to verify high validity [10]-[12]. A practical methodological guidance on designing and running controlled experiments involving human participants [13]. One of the key components in designing a tool evaluation experiment is demographic measurements by survey and interview. In a controlled experiment, this data can be gathered before or after a task to better understand the level of expertise, their background, and their experience with certain tools.

A study done by [14] was aimed to characterize the SE practices and to provide a view on the latest SE techniques, tools and metrics employed by practitioners. The finding of the study gives benefits both SE experts and researchers to identify the strength and weakness and motivating more cooperation both academia and industry. In this study, they gather the profiles and demographics of the practitioners taking part in the survey, e.g., academic qualification,

Revised Manuscript Received on February 11, 2019.

Suryani Ismail, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030, Kuala Terengganu, Terengganu, Malaysia.

Fatihah Mohd, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030, Kuala Terengganu, Terengganu, Malaysia.

Masita @ Masila Abdul Jalil, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030, Kuala Terengganu, Terengganu, Malaysia.

professionals, industry sectors of the companies, and project sizes. The selected parameters are an important to ensure external validity of results.

There are three demographic profiles about the subjects' abilities: a general programming experience, the participant's experience with Java, annotations and Net Beans, and English level of the participant [15]. The other study, covered demographic data in term of professional, academic backgrounds [16], year of work experience [17], years of professional programming experience, the size of the typical software development team, and programming languages typically used [18]. From the past studies, it can be found that the demographic profiles are essential to evaluate and validate the software quality.

Thus, it this study, the component reusability evaluation approach (CREA) was evaluated using a controlled experiment applying human subject. The analysis and interpretation of the research results in the respondents' profile are presented in this study. Evaluating the reliability of the proposed metrics using statistical analysis result of controlled experiment is also presented. The remainder of this article is structured as follows. The experimental methods for the study are presented in Section 2. In Section 3, the experiment's results, summarizes the findings and the lessons learned are discussed and analyzed. Finally, Section 4, concluded the study.

II. METHODOLOGY

A. Data Collection

The reusability metrics of the CREA was presented in the past study, covered five metrics for component reusability evaluation, namely the value of Component Documentation Level (VCDL) for measuring the documentation level, Value of Component Observability (VCO) for measuring observability, Value of Component Customizability (VCC) for measuring customizability, and value of Component Fan-in (VCFi), and Value of Component Fan-out (VCFo) for measuring external dependency of the component [19], [20].

In this study, the evaluation of the research results on the reusability metrics of the CREA was executed. To validate the CREA, a controlled experiment was conducted using human subjects. The subsequent section presents further descriptions on the controlled experimental in the SE research and practice. The questionnaires were distributed to 20 respondents and were selected based on their knowledge in SE. The respondents are ten lecturers and nine post graduate students from School of Informatics and Applied Mathematics (SIAM), University Malaysia Terengganu (UMT) and one staff from the Application Development Section from the Information Technology Management Center, University Malaysia Terengganu (UMT). Most of them have an experience in the software engineering field. All of them have the background in Object Oriented Design and Java programming language. These factors are very important to retain their high motivation to contribute in the experiment until it's done.

B. Data Analysis

The data were analyzed using Statistical Package for Social Science (SPSS) version 20. The analysis included the reliability test, demographic and interpretation of the respondents as described in the following sections.

C. Questionnaire

This experiment was targeted to record the dependent variables based on their corresponding independent variables. In order to obtain the evaluation results from the respondents in examining the four Java components, a set of questionnaires was distributed to them. The questionnaires are specifically formulated to support user evaluation of this controlled experiment. The respondents were required to fill-in the questionnaires during this session. The questionnaire is contained in two sections. Section A is to be answered by respondents before the experimental tasks begin, while section B contains the results of user evaluation and their experience of using the given Java components. In this study, a finding of section A is presented. It contains mostly about the user background of software engineering processes. It is targeted to collect some information regarding the respondent's background such as: working experience and some aspects related to their familiarity of software engineering tasks (see Appendix).

D. Reliability Test

Reliability is the degree of accurateness in an experiment. Cronbach's alpha is the most commonly used in measurement of reliability. It is used to recognize the construct of internal validity used in a research. To recognize the threshold for acceptance of an internal validity, the reliability levels recommended by [21] were used in this research. Table 1 shows the reliability levels for Cronbach's alpha in categorizing whether the internal validity is accepted.

Table 1: Reliability levels of Cronbach's Alpha

Reliability Level	Acceptance
Below 0.6	Unacceptance level
0.6 – 0.7	Low acceptance level
0.8- 0.9	Moderate to high acceptance level
Above 0.9	High acceptance level

III. RESULTS AND DISCUSSION

The results of the experiment are reported in this section. It presents the collected data based on the demographic information of the respondents and the outcomes according to the analysis of the collected data.

A. Reliability

Reliability test using Cronbach's alpha showed that VCFi has the lowest acceptable value, followed by VCDL, VCO, and VCC, while VCFo has the highest acceptance value. The results of the reliability test, as shown in Table 2, indicate that all values are above 0.8, and most of the values are above 0.9. Henceforth, the results prove that the questionnaire is acceptable as a reliable measurement instrument for the survey.

Table 2: Reliability statistics test

Variable	Cronbach's Alpha
VCDL	0.911
VCO	0.938
VCC	0.968
VCF_i	0.818
VCF_o	0.977

B. Demographics Interpretation of the Respondents

The following information represents the demographics of the respondents in terms of their background, experience in SE practices, motivation participant level, as well as their exposure, knowledge and familiarity with the software component used in the case study. These data were verified and analyzed. Table 3 shows the percentage of respondents' background. Overall, from 20 participants; only 5% are system analysts, 45% (25 % master students and 20 % PhD students) have previous background in the software development field and research assistants. The remaining 50 % of the respondents are lecturers.

Table 3 also shows the summary of respondents' previous experience. 45% of the respondents have 11-20 years' experience, 30% respondents have less than 5 years' experience while 20% respondents have 5-10 years' experience. Only 5% respondents have more than 20 years of experience. The respondent's experience was a factor that helped the smooth running of the experiment. Overall, it can be concluded that 50% of the respondent that participate in this controlled experiment have experience more than 11 years in CBSD. We believed that the longer the exposure in CBSD, the more knowledgeable the respondents. Thus, producing more reliable results of the controlled experiment conducted that can support in measuring the reliability of CREA.

Table 3: Respondents' working background and experience

Categories	Percent of Respondents (N=20)
Position, n (%)	
System Analyst	5
Master Student	25
PhD Student	20
Lecturer	50
Experience (years)	
Less than 5 years	30
5-10 years	20
11-20 years	45
21-30 years	5

C. Work Experience in Software Development

The summary of the respondents' prior experience percentage is presented in Table 4. As for the first characteristic, it is observed that 50% respondents have average experience, followed by 40% respondents with substantial experience, then 5% respondents have little and professional experience respectively. There were no respondents that have no experience in practicing Software Engineering theories ranging from the initial phase of requirements, analysis, design, implementation, testing, delivery phase until software maintenance and component reuse in overall. This evidence reflected in the results on respondents' experience as described in Table 3. All the respondents have an experience in software engineering practice.

The frequency of collected data on respondents' former experience, particularly in software development activities; namely programs coding during implementation phase is shown in the second characteristic. About 50% of the respondents have an average level of experience in software development while a little less (45%) claimed that they have substantial experience. The rest of the respondents (5%) are respondents in the expert category in software development. In contrast, no subject acknowledged that they have either none or little experience in software developments. These findings agree with the respondents' working experience as software developers and engineers, as well as respondents' exposure to development projects in performing tasks during their work and study.

The third characteristic present statistics on the respondents' previous experience in object oriented design. 0% or none of the 20 respondents have professional experience in object-oriented design. 50% have substantial experience, 25% have average experience, while only 10% have little experience. Surprisingly, 15% of the respondents have not had any previous experience in object-oriented design. These findings are important to conclude whether previous experience being involved in object-oriented design correlates and influences the causal relationships with software component reuse.

Subsequently, the forth characteristic of the respondents' prior experience is involving the management of software component reuse, including maintenance and application in software development. Findings show that 30 % respondents have an average or substantial experience, in dissimilarity to 20% having less experience. Another 10% admitted that they have no particular experience, while the balance 10% of the respondents declared that they have professional experience in software component reuse.

The fifth characteristic presents the overall frequency of respondents' experience in performing software component reuse in component based software development. It is

observed that no respondents claimed that they have professional experience in performing software component reuse. In contrast, 30% admitted that they substantial experience, whereas only 10% have no experience, while another 20% acknowledged that they have little experience in the performing software component reuse. The other 40% have an average practice in the performing software component reuse. This information is essential to logically relate whether respondents' former experience in performing software component reuse can be related to performing the task in the reuse of sample component in simple Java development. As a conclusion, all respondent' prior experiential characteristics, covers: (1) experience in software engineering practice, (2) experience in software development, (3) experience in object-oriented design using software component reuse, (4) experience in software component reuse management/ maintenance/ application, and (5) experience in performing software component reuse, are important to support controlled experiment for CREA.

Table 4: Respondent's prior experiences

Experience (Range)	Percent (%)
1. Experience in software engineering practice	
None	0
Little	5
Average	50
Substantial	40
Professional	5
2. Experience in software development	
None	0
Little	0
Average	50
Substantial	45
Professional	5
3. Experience in object-oriented design using software component reuse.	
None	15
Little	10
Average	25
Substantial	50
Professional	0
4. Experience in software component reuse management/ maintenance/ application	
None	10
Little	20
Average	30
Substantial	30
Professional	10
5. Experience in performing software component reuse	
None	10
Little	20
Average	40
Substantial	30
Professional	0

D. Software Artefacts

Table 5 presents the descriptive statistics on the respondents' previous experience with related software artefacts while performing software component reuse. From the schedule, three types of software artefacts (source code, design, and class diagram), it was found that 5% of the respondents have been involved specifically with design and class diagram artefacts during software component reuse. Followed by, source code and class diagram with 10%. On the other note, 15% of the respondents' are experienced with class diagram whereas 25% respondents' acknowledged that they had some experience using source code and the associated design. Finally, the other 40% of the respondents' have experience involving source codes which are the most used artefacts in performing software component reuse. In

conclusion, the higher the percentage of the involvement with in performing software component reuse, the easier the respondent to perform the software component reuse in software development in this experiment.

Table 5: Related software artefacts

Description	Range	Percent
Related software artefacts that you have been involved with in performing software component reuse	Source Code	40
	Design	5
	Class Diagram	15
	Source Code and Design	25
	Source Code and Class Diagram	10
	Design and Class Diagram	5

Respondents' motivation in contributing in the experimentation was also investigated. Table 6 shows the frequency distribution of respondents' motivation towards participation in the controlled experiment. From the given questionnaires, 40% of the respondents claimed that they were highly motivated to participate in the experiment while the other 35% respondents' are well motivated, and the balance of 25% respondents' stated that they are fairly motivated. The findings are important to ensure the selection effect of internal validity is carefully controlled in conducting this controlled experimentation, and the analysis interpreted from this data is objectively concluded.

Table 6: Percentage of respondents' motivation of participation

Description	Range	Percent
Subject motivation in participating the experiment	Fairly	25
	Well	35
	Highly	40

The results of respondents' motivation can be revealed to their reasons for participating in the controlled experiment, as shown in Table 7. About 45% of the respondents participated to gain practical experience and to apply the related SE interested in user testing of the SE research projects. Meanwhile, 30% of them wanted to gain practical experience while 20% felt the need to apply the related SE theories and knowledge from their work. The remaining 5% of the respondents have personal reasons to that motivated them to participate in this experiment. The important purpose of this question is to observe the respondents' exposure and their familiarity on the sample component under investigation in performing the required experimental tasks.

Table 7: Percentage of respondents' reasons of participation

Participation Reasons	Percent
To gain practical experience	30
To apply the related SE	20
To gain practical experience and to apply the related SE	45
Other	5

Table 8 shows the descriptive statistics of the respondents' knowledge of selected software components. The majority of the respondents (65%) are well familiar with the software components while 25% conceded that they are fairly familiar with the components. Interestingly, 5% of the respondents disclosed that they highly understand and exposed to the software component's environment, while the remaining 5% admitted that they have a poor understanding of software components. In this research viewpoint, the higher the respondent's understanding of the case study, the results for the controlled experiment conducted is more precise for evaluating CREA.

Table 8: Percentage of respondents' knowledge in sample component

Description	Range	Percent
Understanding sample component case study; its environment and functionality	Poorly	5
	Fairly	25
	Well	65
	Highly	5

In general, it can be concluded that all respondents are considerately familiar and exposed to the software components case study, either in terms of the system development perspective, or on the component reuse. As a result, it could be established that the effects of maturation and instrumentation of the internal validities have been carefully held and reduced in conducting this controlled experimentation, and the analysis and interpretation of the has been fairly conducted.

IV. CONCLUSION

A survey gathered responses from 20 practitioner software engineers working in the software development. According to our demographic data, 50% of the respondent that participate in this survey have work experience more than eleven years in

CBSD. It can be concluded that, the longer year exposure in CBSD, the more knowledgeable the respondents. Thus, producing more reliable results of the survey conducted can support in measuring the reliability of CREA.

APPENDIX

Questionnaire
Section A: Personal Details and Working Background

Position: _____

Qualifications:
(PhD in Computer Sc/ Master in SE/ Bachelor in Comp.Sc)
Working Experience. (Tick √ **ONE** answer only in the box)

	a.	Less than 5 years
	b.	5 -10 years
	c.	11 -20 years
	d.	21 – 30 years
	f.	More than 30 years

Please circle your responses to the following statements based on the scales below.
(Choose ONE answer only)

1. How do you rate your experience in software engineering practice?

None	Little	Average	Substantial	Professional
1	2	3	4	5

2. How do you rate your experience in software development?

1	2	3	4	5

3. How do you rate your experience in object-oriented design using UML modelling?

1	2	3	4	5

4. How do you rate your experience in software component reuse management/maintenance/application?

1	2	3	4	5

5. How do you rate your experience in performing software component reuse in component based software development (CBSD)?

1	2	3	4	5

6. What kind of software artefacts that you have been involved with in performing software component reuse? (Tick √ all that applies OR fill in the blank if specified otherwise)

	a.	Source code
	b.	Design
	c.	Class diagram
	f.	Others (please specify) _____

Motivation and Domain Knowledge Background

1. In overall, kindly estimate how motivated you are to participate in this experimentation.

Not	Poorly	Fairly	Well	Highly
1	2	3	4	5

2. Please choose the specific reasons for the above answer. (Tick √ all that applies OR fill in the blank if specified otherwise)

	a. To gain practical experience involving the user testing in the SE research project
	b. To apply the related SE theories and knowledge from my coursework
	c. Others (please specify) _ _____ _____

3. In overall, kindly estimate how well you understood the Library system as the case study application; in terms of software component reuse for CBSD.

Not	Poorly	Fairly	Well	Highly
1	2	3	4	5

ACKNOWLEDGMENT

This study is fully supported by Fundamental Research Grant Scheme (FRGS), 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. R. S. Pressman, Software Engineering: A Practitioner's Approach. New York: McGraw-Hill, 2001.
2. A. I. Khan, N. Qayyum, and U. A. Khan, "An improved model for component based software development," Software Engineering, 2(4), 2012, pp. 138-146.
3. A. Ibraheem, A. Abdallah, and Y. Mohd, "Taxonomy, definition, approaches, benefits, reusability levels, factors and adaption of software reusability: a review of the research literature," Journal of Applied Sciences, 14(20), 2014, pp. 2396-2421.
4. V. R. Basili, L. C. Briand, and W. L. Melo, "How reuse influences productivity in object-oriented systems," Communications of the ACM, 39(10), 1996, pp. 104-117.
5. S. Ismail1, W. M. N. Wan Abdul Kadir, and N M Mohd Noor, "Reuse component object oriented programming can help in learning and development system," International Conference on Quality of Teaching and Learning, 2012.
6. B. Jalender, B, A. Govardhan, and P. Premchand, "Designing code level reusable software components," International Journal Software Engineering Application, 3, 2012, pp. 219-229.
7. C. W. Kruger, "Software reuse," ACM Computing Surveys, 24(2), 1992, pp. 132-138.

8.
9. S. Hong, and B. Koelzer, "A comparison of software reuse support in object-oriented methodologies," IRMA, 1995, pp. 1-18.
10. R. Land, A. Alvaro, and I. Crnkovic. "Towards efficient software component evaluation: An examination of component selection and certification," 34th Euromicro Conference Software Engineering and Advanced Applications, 2008, pp. 274-281.
11. C. Wohlin, and M. Höst. "Controlled experiments in software engineering," Information and Software Technology, 43, 2001, pp. 921-924.
12. P. Runeson, and M. Höst. "Guidelines for conducting and reporting case study research in software engineering," Empirical Software Engineering, 14, 2009, pp. 131-164.
13. H. Chennamsetty, Experimentation in global software engineering. Master thesis, Sweden: Blekinge Institute of Technology, 2015.
14. A. J. Ko, T. D. Latoza, and M. M. Burnett, "A practical guide to controlled experiments of software engineering tools with human participants," Empirical Software Engineering, 20, 2015, pp. 110-141.
15. V. Garousi, A. Coşkunçay, A. Betin-Can, and O. Demirörs, "A survey of software engineering practices in Turkey," Journal of Systems and Software, 108, 2015, pp. 148-177.
16. M. Sulír, "Sharing developers' mental models through source code annotations," Federated Conference on Computer Science and Information Systems, 2015, pp. 997-1006.
17. P. Pourali, and J. M. Atlee, An experimental investigation on understanding the difficulties and challenges of software modellers when using modelling tools. 2018, Available: https://cs.uwaterloo.ca/sites/ca.computer-science/files/uploads/files/cs-2018-03_0.pdf.
18. M. Kim, T. Zimmermann, R. DeLine, and A. Begel. "The emerging role of data scientists on software development teams," 38th International Conference on Software Engineering, 2016, pp. 96-107.
19. W. Snipes, A. R. Nair, and E. Murphy-Hill, "Experiences gamifying developer adoption of practices and tools," 36th International Conference on Software Engineering, 2014, pp. 105-114.
20. S. Ismail, W. M. N. W. A. Kadir, N. M. M. Noor, and F. Mohd, "Determining characteristics of the software components reusability for component based software development," Journal of Telecommunication, Electronic and Computer Engineering, 9, 2017, pp. 213-216.
21. S. Ismail, F. Mohd, M. A. Jalil and W. M. N. W. A. Kadir, "Development metrics measurement level for component reusability evaluation approach (CREA)," International Journal of Electrical and Computer Engineering.
22. R. A. Peterson, "Meta-analysis of Alpha Cronbach's coefficient," Journal of Consumer Research, 21(51), 1994, pp. 381–391.