Indexed in Clarivate Analytics WoS, and ScienceOPEN

E-B Environment - Behaviour Proceedings Journal

ITMAHuB2023

https://kppim.uitm.edu.my/research/conferences/itmahub-2023

2nd International Conference on Information Science, Technology, Management, Humanities, and Business

Hotel UiTM, Shah Alam, Malaysia, 26-27 Sep. 2023

Organised by College of Computing, Informatics and Mathematics, UiTM, Malaysia

How Freelance Developers Embrace Software Reuse Practice? A perspective analysis using clustering techniques

Mohd Akmal Faiz Osman[1], Tuan Norhafizah Tuan Zakaria[2], Khalid Abdul Wahid[3], Mohamad Noorman Masrek[4]

[1] Universiti Teknologi MARA Cawangan Kelantan Kampus Machang, Jalan Bukit Ilmu, 18500, Malaysia
[2] Universiti Teknologi MARA Cawangan Negeri Sembilan Kampus Kuala Pilah, Pekan Parit Tinggi, 72000, Malaysia
[3] Universiti Teknologi MARA Cawangan Kelantan Kampus Machang, Jalan Bukit Ilmu, 18500, Malaysia
[4] Universiti Teknologi MARA Cawangan Selangor Kampus Puncak Perdana, Seksyen U10, 40150, Malaysia

akmalfaiz@uitm.edu.my, tuannorfaizah@uitm.edu.my, mnoorman@uitm.edu.my, awkhalid@uitm.edu.my,
Tel: 09-9763027

Abstract

The advancement of ICT technologies has modernized software development practices, thus presenting opportunities for developers to reuse software components in developing new software. These practices allow developers to use reliable reusable software components to quickly develop software. However, not much research has reported how freelance developers capitalize on these practices, despite the rapid rise of freelancing development. This research investigates how freelance developers embrace software reuse practices to achieve project success. A survey that gathers 351 responses from freelancers was analyzed using clustering techniques, revealing high levels of self-efficacy and satisfaction towards deploying software reuse practices and highlighting the need for software development firms to embrace this method.

Keywords: Software Reuse Practice, Freelance Developers, Clustering Techniques, Malaysian Developers.

DOI: https://doi.org/10.21834/e-bpj.v9iSI18.5471

1.0 Introduction

Software reuse practices are defined as the development of new software using existing software components, which consists of the source code, design, frameworks, components, and libraries (Barros, Pinciroli, Malatonga, & Martinez, 2018). The objective of software reuse practices is to develop software using proven, reliable, less-bugs, and robust software components because these components have gone through rigorous testing phases by previous developers (Restrepo, Monslave, Mazo, Vallejo, & Correa, 2022). The advancement of ICT has created new opportunities for developers to capitalize on the fast-growing availability of reusable assets via open source or subscription (Makitalo, Taivalsaari, Kiviluoto, Mikkonen, & Capilla, 2020). Software reuse practices have also evolved, as Makitalo et al., (2020) redefined it as opportunistic reuse that involves the development of software using software components that were originally not made to work together due to the advancement of ICT and software development platforms (Zhou, 2020). Although certain developers have overlooked reuse practices due to a lack of expertise (Bakar & Kasirun, 2014), incomplete documentation (Jusoh, Gorment, Nor, Nor, & Muhammad, 2017; Ahmad, Ubaidullah, & Lakulu, 2014), and difficult to maintain (Baharom, Deraman, &

eISSN: 2398-4287 © 2024. The Authors. Published for AMER and cE-Bs by e-International Publishing House, Ltd., UK. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/). Peer–review under responsibility of AMER (Association of Malaysian Environment-Behaviour Researchers and cE-Bs (Centre for Environment-Behaviour Studies), College of Built Environment, Universiti Teknologi MARA, Malaysia.
DOI: https://doi.org/10.21834/e-bpj.v9iSI18.5471

Hamdan, 2014), recent studies have shown that freelance developers have the expertise, experience, knowledge, skills, and self-efficacy to deploy this method (Osman, Masrek, & Wahid, 2022), thus offering attractive solutions for software development jobs. In the post-COVID-19 era, more developers tend to work as freelancers due to high salaries and work flexibility (Rauf, Petre, Tun, Lopez, & Nuseibeh, 2023). Renowned as highly skilled and wealthy experience across software development phases (Osman et al., 2022), the creative and innovative solutions offered by freelancers were too good for software development organizations to overlook (Sison & Lavilles, 2018). This is a crucial aspect of development, as Rangasamy, Negaraj, and Nandhakumar (2021) highlighted that nowadays, software must have high quality to cater to many users and achieve development efficiency to market the software quickly. However, questions remain on how freelance developers embrace software reuse practices in terms of their ability and satisfaction since there is limited research in the literature that explains this phenomenon. Hence, this study objectively tries to bridge this gap by investigating how freelance developers embrace software reuse practices to achieve project success by using a survey questionnaire and clustering analysis.

2.0 Literature Review

Knowledge reuse theory by Markus (2001) explains the importance of reusing existing knowledge to embrace innovation and increase problem-solving efficiency (Zhao, 2020). Knowledge reuse theory has been widely used to investigate the impacts of knowledge reuse on individuals and organizational success (Majchzrak et al., 2004). In the case of software reuse practices, Zhao (2020) has extended knowledge reuse theory in the context of freelance software development, in which the author postulated that freelance developers who make a living solely on a software development project basis need to be creative and innovative in replicating or innovating the existing software reusable assets, such as the source codes, designs, modules, libraries, and components to not only offer high-quality software to their clients but also in quicker time. Other than Zhao's (2020) studies, the studies of Allen and Parsons (2010), Haefliger et al., (2008), and Kyriakou et al., (2017) also have applied the knowledge reuse theory in the context of software reuse practices.

In 2019, Capilla, Gallina, Cetina, and Favaro (2019) investigated the current trends of software reuse by conducting a survey among software practitioners across Europe, the United States, and Asia. Capilla et al., (2019) used the survey that was first created in 1993 by Frakes and Fox (1993) to identify whether reuse practices are still well alive. Capilla et al., (2019) found that reuse practices have greatly evolved upon comparing the responses from Frakes and Fox (1993). Capilla et al., (2019) also found that 76% of developers preferred reuse practices, while 71% agreed that there were no license and legal issues and 88% agreed that they had their reuse repositories for future projects. The findings reported by Capilla et al., (2019) have shown that reuse practices are highly regarded by modern developers. However, the respondents in this study were not freelance developers, who might have different points of view.

Furthermore, Beydoun, Hoffmann, and Garcia, (2020) investigated the success factors that underpin software reuse practices. From the case study and knowledge level analysis, Beydoun et al., (2020) revealed 5 factors that must be considered when applying reuse practices, namely software-related components, domain components, libraries, used, methodology used, and the developers' knowledge and experience. However, Beydoun et al., (2020) did not provide empirical evidence to support the framework.

Makitalo et al., (2020) introduced opportunistic reuse terms and conducted mixed-method studies to understand how developers apply new forms of software reuse when developing real-world software systems. The survey and interviews were conducted with developers across Brazil, Denmark, Finland, Italy, Spain, and Sweden. Makitalo et al., (2020) found that the number of reusable components that are readily available is overwhelming and increasing. Moreover, this study has contributed to unearthing the new opportunistic reuse practice employed by professional developers across the world to provide creative and innovative solutions. Hence, further research is needed to study these new, opportunistic reuse platforms in different contexts.

Meanwhile, Kruger and Berger (2020) investigated the cost factor of software reuse strategies by combining interviews with systematic literature reviews. Kruger and Berger (2020) found that the development process and cost greatly influenced the reuse strategies employed by developers. The findings also revealed that reusable assets come in many forms (Makitalo et al., 2020), and can be used by the project budget and nature. Ultimately, Kruger and Berger (2020) suggested that further studies be conducted to provide more conclusive evidence in understanding this phenomenon.

More recently, Chen, Badampudi, and Usman (2022) investigated software reuse practices among developers in small and medium enterprises in Sweden. By referring to the study of Makitalo et al., (2020), Chen et al., (2022) characterized reuse practices into participatory and opportunistic reuse and found that these practitioners deployed both practices following the nature of the projects (Kruger and Berger, 2020). Additionally, as reported by Chen et al., (2022), software reuse leads to better product quality with less development and market time. The findings reported by Chen et al., (2022) further reflect the evolvement of software reuse practices, which requires further investigation in the context of Malaysian freelance developers.

Based on the studies above, there are existing gaps that need to be bridged with the perception of software reuse practices by Malaysian freelance developers. The findings are expected to help to validate the frameworks proposed by Kruger and Berger (2020) and Beydoun et al., (2019). As such, this study assists in continuing the trends of reuse practices investigated by Capilla et al., (2019), while providing the state-of-the-art practice of software reuse in Malaysia to complement the study by Chen et al., (2022).

3.0 Methodology

In software reuse practices research, surveys using questionnaires have been regarded as the dominant method to collect data on individual developers (Zhou, 2020), and to perform statistical analysis. A questionnaire that consists of 88 questions was distributed to

freelance developers of Malaysian nationality. Data was collected online using Google Forms and through physical paper-based distribution. The questionnaire has gone through pre-testing with 5 software practitioners with a minimum of 5 years of experience in software reuse and 10 years in software development. Subsequently, the questionnaire underwent pre-testing with 5 academicians with a minimum of PhD qualifications and 5 years of experience in quantitative research to complete the instrument validity process. Given the freelance developers sampling frame is difficult to determine because these developers are scattered all over Malaysia, that consists of part-time freelancers, full-time freelancers, or individuals that have experience as freelance developers, therefore the non-probability convenience sampling method was chosen to ease data collection. However, the sample chosen must meet the requirements of possessing Malaysian nationality, having a minimum of 3 years of experience as a freelance developer, have a minimum of 5 years of experience in software development, and having experience in practicing software reuse. Data for this research were analyzed using clustering techniques that include six clustering techniques, namely K-Means clustering, K-Means clustering with PCA, single linkage, complete linkage, DBScan, and DBScan with PCA. These clustering techniques were chosen to provide different angles of analysis to support the analysis of the same data using the Structural Equation Modelling (SEM) technique, which will be presented in another study. This research comprises five essential steps: data collection, data preprocessing, application of clustering algorithms, and model evaluation.

3.1 Data Collection

The dataset utilized for this research was obtained through a comprehensive survey conducted among Freelance Developers in Malaysia, encompassing various locations nationwide. For the study, this research only explains 16 questions out of 88, which are categorized into demographics (gender, age, academic qualifications, scope of development job, certificates possessed), experience (years of experience in software development, years of experience as freelancer, number of training attended, number of years in software reuse practices), self-efficacy (ability to search reuse assets, ability to understand reuse asset, ability to manage reuse assets, ability to modify reuse assets, ability to develop new software using reusable asset) and satisfaction. The data are explained to capture the current state of how Malaysian freelancers embrace software reuse practices. Table 1 depicts the acquired dataset.

Table 1. Dataset of variables

Gender	Age	Qual.	Dev. Scope	Cert.	Dev. Exp	Freelance Exp.	Training Attended	Reuse Exp.	Satis-faction	Search Asset	Undstd. Asset	Manage Asset	Modify Asset
1	3	4	1	2	5	5	4	4	4	5	5	4	4
1	1	1	3	2	4	4	5	5	5	5	4	4	4
1	3	3	1	2	5	5	5	5	5	5	4	5	5
1	3	3	1	2	4	4	4	5	5	4	4	4	4
1	3	2	1	2	5	5	4	5	4	4	4	4	4
…	…	…	…	…	…	…	…	…	…	…	…	…	…
1	3	2	3	1	5	4	5	4	4	5	4	4	4
1	3	2	4	2	5	4	4	5	5	5	4	4	4
1	3	2	3	2	4	4	3	4	4	4	4	4	4
1	3	2	3	1	5	4	5	5	4	5	5	4	4
2	3	2	3	1	4	4	5	4	5	4	4	4	4

3.2 Data Preprocessing

Data preprocessing generally involves data cleaning to ensure the absence of missing values, and all data points are expressed in numerical form. This research centers on evaluating how freelance developers embrace software reuse practices that involve their perceived ability, experience, and satisfaction.

Correlation analysis using a heatmap within the Jupyter Notebook environment was employed. This analytical approach is aimed at discerning the factors influencing the level of satisfaction with software reuse among respondents. Accordingly, by employing color gradients to represent correlation coefficients, the heatmap facilitates a clear visualization of the strength and direction of linear relationships between diverse attributes and the focal variable of satisfaction. This method also yields valuable insights into the positive and negative associations between variables and developers' contentment with software reuse.

The correlation analysis, facilitated by the heatmap, serves as a pivotal phase during data exploration and attribute selection endeavors. Furthermore, by furnishing both a visual representation and quantitative assessment, this analysis aids in the identification of meaningful interrelationships among variables. This groundwork now paves the way for subsequent model creation and analysis, guiding the trajectory toward developing predictive models. Ultimately, these insights can enable informed decisions to amplify software reuse development practices and satisfaction among freelance developers. Table 2 shows the correlation values between the attributes and the level of satisfaction with software reuse, while Figure 1 illustrates the heatmap.

Table 2. Correlation values between attributes and the level of satisfaction with software reuse

Variable	Correlation Value
Ability to search reusable assets	0.5
Ability to understand reusable assets	0.39
Ability to help others regarding reusable assets	0.36
Ability to manage reusable assets	0.35
Ability to develop new software using reusable assets	0.35

Ability to modify reusable assets	0.33
Number of years in using reusable assets	0.31
Number of years in software development	0.22
Number of years as a freelancer	0.21
Number of training attended	0.14
Highest academic qualification	0.03
Age	0.02
Certificates Possessed	0.01
Scope of Freelancing	0.00
Gender	-0.02

Table 2 presents the correlation coefficients among different attributes and the satisfaction level. These coefficients, depicted as numerical values, elucidate the intensity and orientation of the linear connections between individual attributes and the focal variable, satisfaction. Positive correlation coefficients signify a direct connection, wherein an escalation in one attribute corresponds to heightened satisfaction. Conversely, negative correlation coefficients denote an opposing link, implying that elevated values in a specific attribute align with reduced satisfaction concerning software reuse. Subsequently, these correlation values are visualized through a heatmap, as showcased in Figure 1.

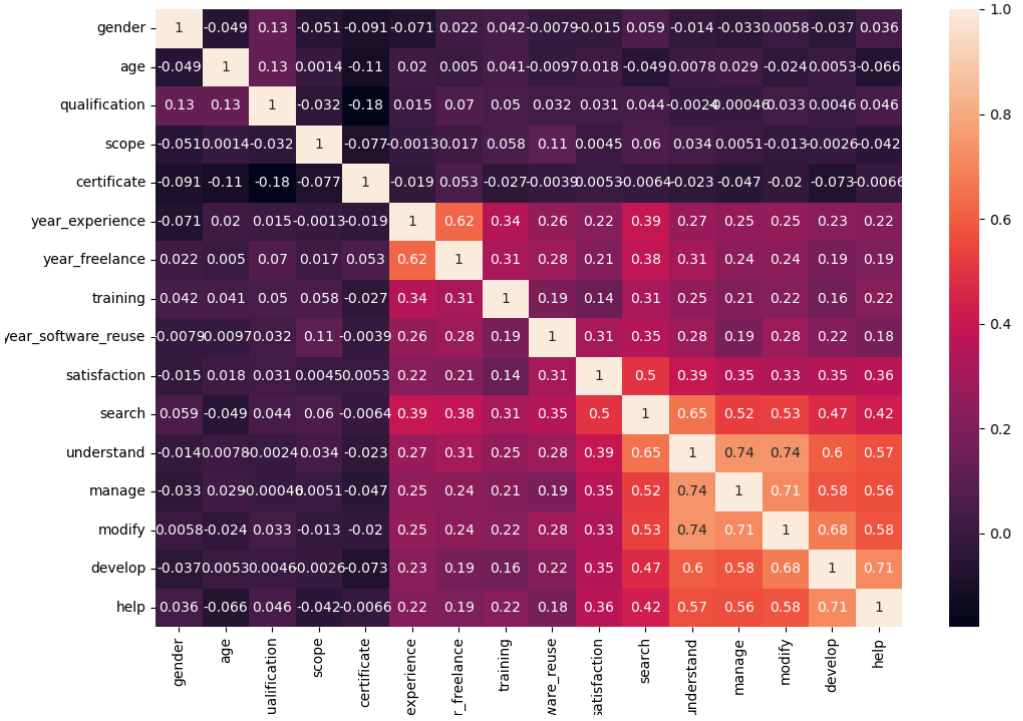

Figure 1. Heatmap displaying the correlation of attributes with the level of satisfaction with software reuse

The heatmap visually represents the connections between diverse variables and their influence on satisfaction. Warmer color tones denote robust positive correlations, while cooler shades indicate weaker or negative correlations. This graphical tool facilitates selecting features by accentuating crucial attributes that showcase noteworthy correlations with satisfaction. Furthermore, it aids in pinpointing the pivotal factors that impact satisfaction within the examined dataset. Guided by insights gleaned from the correlation heatmap, the ten most influential variables were included for subsequent analysis. These variables encompass self-efficacy ("search," "understand," "help," "manage," "develop," "modify,"), experience ("year_software_reuse," "year_experience," "year_freelance," "training"). Moreover, these selected attributes constitute the final choices for employment within the clustering algorithms.

3.3 Clustering Models and Model Evaluation

Next, multiple clustering models were created to group variables based on the satisfaction level. K-Means, K-Means with Principal Component Analysis (PCA), Hierarchical Clustering using Single Linkage, Hierarchical Clustering using Complete Linkage, DBScan, and DBScan with PCA were employed. The performance of each clustering algorithm was then assessed using various evaluation metrics, including the Within Cluster Sum of Squares (WCSS), Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index. These metrics provide valuable insights into the effectiveness and quality of the clustering solutions, aiding in the selection of algorithms for satisfaction and software reuse practices. Table 3 presents the WCSS, Silhouette Coefficient, Calinski-Harabasz Index,

and Davies-Bouldin Index values obtained from various clustering models. These metrics indicate the clustering algorithms' performance in segmenting the Malaysian freelance developer survey dataset. Subsequently, by analyzing these evaluation measures, valuable insights into the efficacy and appropriateness of each clustering approach for satisfaction and analysis were gained.

4.0 Findings

First, the effectiveness of each clustering algorithm was evaluated using several evaluation metrics, including the WCSS, Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index. Table 3 displays the evaluation results for each algorithm applied to this research dataset.

Table 3. Evaluation metrics for each clustering algorithm

Technique	WCSS	Silhouette Coefficient	Davies-Bouldin Index	Calinski-Harabasz Index
K-Means	710.24	0.22	1.67	102.98
K-Means PCA	1542.85	0.27	1.32	281.88
Hierarchical Single	265.52	0.31	0.52	3.25
Hierarchical Complete	416.55	0.27	1.41	129.30
DBSCan	136.77	-0.02	1.59	9.31
DBSCan PCA	794.64	0.04	1.76	43.27

The evaluation metrics employed in this analysis offer distinct viewpoints on the performance of various clustering techniques. The Within-Cluster Sum of Squares (WCSS) metric gauges the compactness of clusters, with lower values indicating more favorable outcomes. Meanwhile, the Silhouette Coefficient measures the clustering quality by assessing the separation between clusters, with a preference for higher values. The Davies-Bouldin Index quantifies the relationship between within-cluster scatter and between-cluster separation, favoring lower values indicative of well-defined clusters. Similarly, the Calinski-Harabasz Index evaluates the balance between cluster separation and within-cluster scatter, with higher values suggesting well-distinguished clusters.

Based on the WCSS values, DBSCAN demonstrated the lowest score (136.77), thus indicating superior cluster compactness. In contrast, K-Means PCA yielded the highest value (1542.85), which implies fewer compact clusters. Additionally, regarding the Silhouette Coefficient, Hierarchical Single achieved the highest score (0.31), thereby signifying better cluster separation, while DBSCAN recorded a negative coefficient (-0.02), which possibly implies overlapping or ill-defined clusters.

Based on the analysis of the Davies-Bouldin Index, Hierarchical Single obtained the lowest value (0.52), thus reflecting well-separated and compact clusters, while DBSCAN PCA produced the highest value (1.76), which suggests less distinct clusters. About the Calinski-Harabasz Index, K-Means PCA acquired the highest value (281.88), which is indicative of well-defined clusters, whereas Hierarchical Single garnered the lowest value (3.25), thereby pointing to less well-defined clusters. Considering the synthesis of these evaluation metrics, the Hierarchical Single technique emerges as a strong contender, performing well across the Silhouette Coefficient, Davies-Bouldin Index, and Calinski-Harabasz Index. DBSCAN also shines in terms of low WCSS, which indicates compact clusters. Additionally, K-Means PCA demonstrates the highest Calinski-Harabasz Index, signifying well-differentiated clusters.

Overall, when evaluating satisfaction levels through software reuse among Malaysian freelance developers, the assessment of all these evaluation metrics collectively suggests that the Hierarchical Single technique is the most suitable approach.

Next, mean satisfaction levels and variable scores were compared across various clustering techniques, focusing on attributes that exhibit a high correlation and those displaying a low correlation. Table 4 presents the mean satisfaction levels and attribute scores in different clustering techniques for high-correlation attributes, while Table 5 depicts the attribute scores for low-correlation attributes.

Table 4. Mean satisfaction levels and variable scores for each clustering technique

Technique	High-Correlation Variables										
	Satis-faction	Search Asset	Undstd. Asset	Help Others	Manage Asset	Develop new s/w	Modify Asset	Reuse Exp.	Devel. Exp.	Freelance Exp.	Training Attended
K-Means	4.50	4.88	4.94	4.83	4.85	4.93	4.87	4.37	4.57	4.40	4.36
K-Means PCA	4.54	4.90	4.93	4.90	4.87	4.94	4.90	4.41	4.65	4.48	4.41
Hierarchical Single	4.14	4.15	4.09	3.94	4.02	4.08	4.07	4.18	4.38	4.19	4.15
Hierarchical Complete	4.14	4.15	4.09	3.94	4.02	4.08	4.07	4.18	4.38	4.19	4.15
DBSCan	4.70	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
DBSCan PCA	4.20	4.39	4.34	4.12	4.16	4.30	4.30	4.28	4.49	4.28	4.28

Table 5. Satisfaction levels and attribute averages in various clustering techniques (low-correlation)

Technique	Low-Correlation Variables					
	Satis-faction	Quali-fication	Age	Certificates	Devel. Scope	Gender
K-Means	4.50	2.04	2.46	1.64	1.67	1.10
K-Means PCA	4.54	2.04	2.68	1.65	1.49	1.08
Hierarchical Single	4.14	2.01	2.47	1.66	1.75	1.10
Hierarchical Complete	4.14	2.01	2.47	1.66	1.75	1.10
DBSCan	4.70	2.30	2.60	1.70	1.60	1.00
DBSCan PCA	4.20	2.01	2.59	1.65	3.00	1.08

Based on Table 4 and Table 5, clustering techniques using various variables related to the satisfaction level with software reuse among freelance developers were evaluated. The cluster with the highest satisfaction level for each technique was identified and the mean values of different attributes within that cluster were calculated. The values in Table 5 represent the mean values from the respective clusters with the highest satisfaction level for each technique. All techniques showed the highest satisfaction levels constituting 'agree' and 'strongly agree' with significant experience with software reuse assets. Meanwhile, the conclusions for the respondents who agreed and strongly agreed that they possessed a great deal of experience with software reuse assets are as follows:

Table 6. Summary of highest satisfaction clusters

Variable	Technique	Mean Value
Search (Ability to search for reusable software assets)	All techniques except DBScan	Agree
	DBScan	Strongly agree
Understand (Ability to understand the reusable software asset)	All techniques except DBScan	Agree
	DBScan	Strongly agree
Help (Ability to help other developers regarding reusable assets)	K-Means, K-Means (PCA), DBScan (PCA)	Agree
	DBScan	Strongly agree
Manage (Ability to manage reusable software assets for future use)	All techniques except DBScan	Agree
	DBScan	Strongly agree
Develop (Ability to successfully develop new software by using reusable software assets)	All techniques except DBScan	Agree
	DBScan	Strongly agree
Modify (Ability to modify reusable software assets to accommodate the software project requirement)	All techniques except DBScan	Agree
	DBScan	Strongly agree
Year software reuse (No. of years practicing software reuse in development projects)	All techniques except DBScan	11–15 years
	DBScan	More than 15 years
Year experience (No. of years involved in software/systems development)	All techniques except DBScan	11–15 years
	DBScan	More than 15 years
Year freelance (No. of years as a freelance developer)	All techniques except DBScan	11–15 years
	DBScan	More than 15 years
Qualification	All techniques	Bachelor's Degree
Age	All techniques	26–30 years old
Certificate	All techniques	No
Scope	All techniques except DBScan	Software Development
	DBScan	Smart-Phone Apps
Gender	All techniques	Male

Table 6 depicts an exhaustive elucidation of attributes and their corresponding mean values, which were derived from the clusters exhibiting the highest satisfaction levels for each utilized technique. The conclusions drawn from this analysis intricately unveil the contributing variables behind the utmost satisfaction levels specific to each technique. This comprehensive exploration also yields invaluable insights into the intricate landscape of software reuse satisfaction dynamics among Malaysian freelancers. Subsequently, these findings are leveraged to engender a holistic discourse on the overall analysis in the upcoming segment.

5.0 Discussion

In this segment, an exhaustive analysis of outcomes derived from diverse clustering techniques is embarked upon to harness and explore the intricate interplay between experience, self-efficacy, software reuse practices, and satisfaction. The primary objective of this comprehensive evaluation is to unveil the multifaceted relationships among variables, satisfaction levels, and the employed clustering approaches. The significance of the findings is to identify factors that contribute to software reuse adoption that enable freelance developers to develop software quickly and effectively. The result of practicing software reuse has been well documented in the literature, therefore, the factors identified will serve as important points to be looked upon in encouraging software reuse practices for both freelance developers and software development organizations. While this finding has provided direction on the verification of self-efficacy and experience as important variables, more training and practice on software reuse would increase software reuse practices. However, another factor such as project characteristics should be explored as the characteristics of freelance projects remain relatively scarce in the literature. Each clustering method delves into different facets of satisfaction and software reuse attributes. This thorough scrutiny consequently becomes the conduit to unearth invaluable insights into the subtle determinants that shape freelance developers'

contentment, experience, and self-efficacy toward software reuse practices. In essence, by meticulously dissecting attribute means, the dialogue evolves along three essential dimensions:

5.1. Satisfaction and Self-efficacy

Across all clustering techniques, the mean values for satisfaction-related attributes, including satisfaction levels and the ability to search, understand, help, manage, develop, and modify, were consistently high, ranging from 4.14 to 4.94. This finding counter-explains the studies by (Bakar & Kasirun, 2014; Ahmad, 2014; Baharom, 2014) in which freelancers have the capabilities of employing reusable software assets in development compared to developers who are currently working permanently in software development firms. These values generally indicate positive satisfaction among freelance developers regarding software reuse. The K-Means and K-Means PCA techniques yielded the highest mean values for most satisfaction variables, suggesting that these techniques could capture a higher level of satisfaction than others. The Hierarchical Single and Hierarchical Complete techniques produced slightly lower mean values, while the DBSCAN and DBSCAN PCA techniques had similar mean values to other techniques, albeit with slightly more variation. This suggests a generally positive satisfaction level among freelancers in Malaysia regarding software reuse. The explanation for this phenomenon is that freelancers have the experience, and self-efficacy toward software reuse; thus, the satisfaction levels are high.

5.2. Experience and Training

The mean values for the attributes related to experience, such as the number of years of software reuse, experience in software development, and years of freelance work, did not vary significantly across the clustering techniques and remained within a similar range (e.g., 4.19 to 4.65). This indicates that the clustering techniques did not have a substantial impact on this variable. Similarly, the mean values for the training attribute were relatively consistent, ranging from 4.36 to 4.48 across the techniques. This suggests that the techniques did not strongly influence the perceived importance of training with satisfaction and software reuse.

5.3. Qualification and Demographics

The mean values for qualification, age, certificate, scope, and gender attributes exhibited minor variations across the clustering techniques. For example, the mean values for qualification ranged from 1.64 to 2.68, with K-Means PCA showing the highest mean value. Subsequently, the mean values for age ranged from 2.46 to 2.68, with DBSCAN PCA having the highest value, while the mean values for the certificate attribute were consistent across most techniques, hovering around 1.65, and the mean values for the scope attribute ranged from 1.49 to 3.00, with DBSCAN PCA showing the highest mean value. Finally, the mean values for gender ranged from 1.00 to 1.75, with Hierarchical Single and Hierarchical Complete techniques having the highest values. These results indicate that the clustering techniques did not strongly influence the qualification, age, certificate, and gender attributes. However, the scope attribute showed some variation, with DBSCAN PCA having a higher mean value than other techniques.

Overall, the analysis of variables provides insights into the impact of clustering techniques on various variables related to satisfaction and software reuse. The analysis also highlights the consistently high satisfaction levels among freelance developers in Malaysia and emphasizes the independence of specific variables such as self-efficacy, experience, training, qualification, and demographics from the clustering outcomes.

6.0 Conclusions, Implications & Recommendations

In this research, K-Means, K-Means PCA, Hierarchical Single, Hierarchical Complete, DBSCAN, and DBSCAN (PCA) were applied to evaluate the results using several metrics. Additionally, the mean values of various attributes were verified within each technique to gain further insights into their impact on satisfaction.

From the analysis of attribute scores, several key findings have emerged. Firstly, the satisfaction and software reuse attributes consistently indicated high satisfaction among freelancers regarding software reuse practices. This suggests that software reuse is positively perceived and valued in the freelance development context. Furthermore, the attributes related to experience, such as the number of years of software reuse, experience in software development, and experience as a freelancer, did not show substantial variations across the techniques. This indicates that the clustering techniques employed might not significantly influence these attributes. Lastly, the qualification and demographic attributes displayed minimal variations across the techniques. This implies that individual qualifications, age, certificate status, scope of development work, and gender were not strongly impacted by the clustering techniques, thus emphasizing their independence from the clustering outcomes.

In essence, these findings imply that software reuse practices must be improvised not only in the context of freelance developers but also full-time developers. The benefits of software reuse practices have been well-documented in the literature; however, research in the Malaysian context has found that traditional developers often overlook software reuse practices and tend to opt for developing from scratch, which typically results in a software crisis (Bakar & Kasirun, 2014; Ahmad, 2014; Baharom, 2014). Therefore, to improve software reuse practices at Malaysian software development firms, developers' self-efficacy and experience in software reuse must be nurtured through software reuse workshops, training, and demonstrations to further capitalize on these practices. Furthermore, Malaysian software firms must also have the expertise in developing reusable software assets that suit the context of Malaysian software specifications, complete with documentation, to enable effective sharing across development firms.

Nonetheless, it is important to note that this analysis has some limitations. Since the results are based on the specific dataset and clustering techniques used, other datasets or different clustering approaches may yield different results. Moreover, the investigation

using partial least squares is encouraged to support the findings. Additionally, the analysis did not consider causal relationships between attributes and satisfaction levels. Thus, future research exploring additional variables or factors that may influence satisfaction and software reuse among freelance developers will be beneficial. Conducting qualitative studies or incorporating domain-specific knowledge can also provide deeper insights into the factors affecting satisfaction and software reuse. The findings of the study indicate that the software development industry needs to educate, train, promote, and encourage software reuse practices among developers, whereby freelance developers need to constantly update their knowledge, skills, and expertise on current reusable assets to stay relevant and survive in the freelancing world.

In conclusion, the analysis of attribute scores for each clustering technique has revealed consistently high satisfaction levels and positive perceptions of software reuse among freelancers. Additionally, the experience, training, qualification, and demographic attributes displayed relatively stable patterns across the techniques. Although this study has several limitations such as being dataset-specific and not establishing causal relationships, it offers valuable insights into the satisfaction and software reuse dynamics in Malaysia's software development industry. Future research can also expand these findings by exploring diverse attributes, employing various clustering techniques, and considering qualitative approaches, all of which may contribute to a more comprehensive understanding of the complex interactions between freelance developer experiences and software reuse satisfaction.

Acknowledgments

This research is partially sponsored by Universiti Teknologi MARA, Malaysia.

References

Ahmad, M. A., Ubaidullah, N. H., & Lakulu, M. (2014). Current Practices in Monitoring Software Development Process in Malaysia. *World of Computer Science and Information Technology Journal* (WCSIT), 4(5), 62–67.

Allen, G., & Parsons, J. (2010)."Is Query Reuse Potentially Harmful? Anchoring and Adjustment in Adapting Existing Database Queries. *Information Systems Research,* 4(2). 56-77.

Baharom, F., Deraman, A., & Hamdan, A. R. (2005). A Survey on the Current Practices of Software Development Process In Malaysia. *Journal of Information and Communication Technology*, 4, 57–76.

Bakar, N. H., & Kasirun, Z. M. (2014). Exploring Software Practitioners Perceptions and Experience in Requirements Reuse An Empirical Study in Malaysia. *International Journal of Software Engineering and Technology*, 1(2), 33–42

Barros, J. L., Pinciroli, F., Matalonga, S., & Martínez-Araujo, N. (2018). What software reuse benefits have been transferred to the industry? A systematic mapping study. *Information and Software Technology*, 103, 1–21.

Beydoun, G., Hoffmann, A., & Garcia, R.V. (2020). Towards an assessment framework of reuse: a knowledge-level analysis approach. *Complex Intelligent System*. 6, 87–95.

Capilla, R., Gallina, B., Cetina, C., & Favaro, J. (2019). Opportunities for software reuse in an uncertain world: From past to emerging trends. *Journal of Software Evolution and Process*. Special Issue Paper.

Chen, X., Badampudi, D., & Usman, M. (2022). Reuse in Contemporary Software Engineering Practices – An Exploratory Case Study in A Medium-sized Company. In e-*Informatica Software Engineering Journal*, vol. 16(1). 220110.

Frakes, W.B., & Fox, C.J. (1993). Sixteen questions about software reuse. *Communication ACM*. 38(6). 75-87.

Haefliger, S., Von Krogh, G., and Spaeth, S. (2008). Code Reuse in Open-Source Software. *Management Science*, 54(1). 180-193.

Jusoh, Y. Y., Gorment, N. Z., Nor, R. N. H., Nor, S. A. M., & Muhamad, S. (2017). A study on the current practices of software development in Malaysia. *3rd International Conference on Science in Information Technology:* Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech 2017. 716–721.

Krüger, J., & Berger, T. (2020). An empirical analysis of the costs of clone- and platform-oriented software reuse. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (ESEC/FSE 2020). Association for Computing Machinery, New York, NY, USA, 432–444.

Kyriakou, H., Nickerson, J. V., and Sabnis, G. (2017). Knowledge Reuse for Customization: Metamodels in an Open Design Community for 3d Printing.

Majchrzak, A. et al. (2004). Can Absence Make a Team Grow Stronger? *Harvard Business Review*, 82, 131-137.

Makitalo, N., Taivalsaari, A., Kiviluoto, A., Mikkonen, T., & Capilla, R. (2020). On opportunistic software reuse. *Computing*. 102.

Osman, M. A.F., Masrek, M. N., & Wahid, K. A. (2022). Malaysian freelance software developer development practice: a preliminary study. *Journal of Information and Knowledge Management*, Special Issue, 48-58.

Rangasamy, A., Nagaraj, V., & Nandhakumar, K. (2021). Software Reuse Management for better efficiency and turnaround time. *IEEE Technology & Engineering Management Conference - Europe* (TEMSCON-EUR), Dubrovnik, Croatia. 1-4.

Rauf, I., Petre, M., Tun, T., Lopez, T., & Nuseibeh, B. (2023). Security Thinking in Online Freelance Software Development. 13-24.

Restrepo, L., Monsalve, S., Mazo, R., Vallejo, P., & Correa, D. (2022). "Snapshot" of the State of Software Reuse in Colombia. *Revista Científica*. 44. 242-256.

Sison, R., & Lavilles, R. (2018). Software gigging: A grounded theory of online software development freelancing. *International Conference on Information Systems*, ICIS 2018, 1–17.

Zhou, J. (2020). Application developer's innovation performance on mobile platforms - Investigating the effect of module reuse. *Proceedings of the 24th Pacific Asia Conference on Information Systems:* Information Systems (IS) for the Future, PACIS 2020.