IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 10, Issue.6, June 2022 E-ISSN: 2347-2693

Employing Design and Development Research (DDR) Approches in Traceability Model For Test Effort Estimation To Support Software Change Management

Mazidah Mat Rejab[1*], Mohd Ridhuan Mohd Jamil[2]

[1]Faculty of Computer Science & Information Technology, Universiti Tun Hussien Onn, Johor, Malaysia
[2]Faculty of Human Development, Universiti Pendidikan Sultan Idris, Perak, Malaysia

[*]*Corresponding Author: mazidah@uthm.edu.my, Tel.: +(07) 453 3601*

DOI: https://doi.org/10.26438/ijcse/v10i6.18 | Available online at: www.ijcseonline.org

Received: 21/May/2022, Accepted: 10/Jun/2022, Published: 30/Jun/2022

***Abstract*—**In the last decade, the management of software projects has become a challenging task. The latest published figures on the status of software projects indicate a large failure rate, which has created a crucial challenge for project managers. In software maintenance, the impact of software changes is an important aspect due to the evolving environment of the software development life cycle. Many of the current traceability approaches and tools are devoted to and restricted to high-level objects such as specifications but fewer capabilities are made available to handle lower-level artefacts such as classes and codes. While test effort estimation has been in place for decades, it remains a major challenge for software project management to make accurate estimates and, ultimately, to successfully complete the software project. This article proposed a novel traceability model for test effort estimation to support software change management employing Design and Development Research (DDR), which may assist software project managers in making more informed decisions on software change management. In this paper will show two phase Fuzzy Delphi Method (FDM) and Nominal Group Technique (NGT) result. The both results in FDM and NGT showed that the key components and elements are located at acceptable level and can be applied whilst the score of more than 70% is achieved. Hence, the evaluation results proved that the proposed model and its prototype are acceptable and significant to support software change management.

***Keywords*—**Design and Development Research (DDR), Traceability Model, Test Effort Estimation, Change Management

I. INTRODUCTION

In general, software maintenance entails modifying the software product after it has been delivered to end customers. It is a bigger task that includes error correction, performance enhancement, feature enhancement, and the removal of old functionality. Consequently, it applies not only to programming, but also to other aspects of the software lifecycle, such as software specifications, testing, and development. Rajlich and Bennett [1] distinguish between three developmental stages: evolution, maintenance, and transformation. The maintenance phase concludes once the development of the core system is complete.

In the maintenance process, handling changes through a traceability link would ensure that the objects involved, i.e., functional and non - functional criteria, design model and part, and test artifacts are modified accordingly as a new change has been implemented. Handling changes through a traceability link in the maintenance process ensures that the objects involved, i.e., functional and non-functional criteria, design model and part, and test artefacts, are adjusted appropriately as a new change is applied. Current traceability approaches demonstrated insufficient coverage of traceability relationships, but improvements are possible. Change requests can be submitted at any stage of the SDLC. [2], [3]. It is necessary to manage software changes in order to meet the changing needs of the customer and thus to satisfy them. [1], [4].

Implementing software changes within the software maintenance process requires an understanding of the effects of software objects as well as the implications of software change activities. Accepting too many adjustments may result in a project delay and cost overrun. However, rejecting too many changes may lead to client dissatisfaction. As a result, dealing with the ever needs and making sound judgments about the progress of the software project is crucial for the software project manager. The expectation of effort change during software maintenance is one of the inputs that can assist and guide the software development project manager in making the appropriate decision.

Due to a lack of support decision for a software project manager can give a decision whereby the impact of cost and time during the change request in the maintenance phase, it is essential to expand the traceability model with a test effort estimation [5]. Lehtinen [6] define a software project failure means a recognizable inability to succeed in the cost, schedule, scope, or quality goals of the project. The decision to accept or reject the change request will be a process that is very complex or difficult. This decision will involve several variables under dynamically changing requirements (LOC involved is large); project managers faced the challenges in support decisions where involved in the maintenance phase that will be the change request with large LOC, time, and financial constraints. In addition, the software project manager is required to estimate the effort right after the change is implemented. The other challenge when there are changes to their software is the need to review these changes, which has a significant impact during the maintenance phase.

Software Project Manager required to estimate the effort right after the change is implemented. Effort estimation and impact analysis are two (2) processes that are important in supporting the Software Project Manager's decision. The problem to be resolved by this study is whether the software traceability approach with effort estimation will be able to efficiently support software changes in the maintenance phase. It is expected that the improved change management tools will allow a dynamic change of traceability structure in response to changes in operations such as addition, insertion, and deletion with an estimate of the cost dan time during the changes. Whereas, the objective of the effort estimation is to estimate the amount of work and time required in implementing the particular changes. [7],[8],[9].

The further paper is assembled as follows, Section II contains related work, Section III explain the methodology with flow chart, Section IV describes results and discussion, and Section V concludes research work and benefit of this studies.

II. RELATED WORK

Few researches have indicated the need for change impact analysis integration with the effort estimation. All the related researches [2],[10] ,[11],[12],[13] ,[14] pointed out that a mean to measure the size of code after the change is needed.

According to an evidence-based study conducted by Nurmuliani et al.[2], several change request attributes have a direct effect on the needed work estimation forecast to make that change. Change request type and change requirements are the identified change request attributes. Furthermore, Nurmuliani et al. [2] claimed that the most significant issues in the were the lack of a formal impact analysis approach to support software changes for work estimation and the lack of a traceability model for the relationships between requirements and classes.

Table 1.Evaluation Of Traceability Model Integrating With Effort Estimation

Model & Author	Traceability Model	Impact Analysis	Regression testing	Test case Coverage	Change Management	Effort Estimation
JavaCodeCoverage Lingampally et al. (2007)	No	Yes	No	No	Yes (Before Change)	No
CATIA (Suhaimi (2006)	Yes	Yes	No	No	Yes	No
GRAYZER Faizah et al. (2012)	Yes	No	No	No	No	No
COCHCOMO Mehran (2013)	No	Yes	No	Yes	No	Yes
CEPM, Sufyan (2016)	Yes	Yes	No	Yes	No	Yes
HYCAT Shahid (2016)	Yes	Yes	Yes	Yes	Yes	No

The approaches that are similar to this research were evaluated on the basis of the evaluation criteria mentioned above. The results of the tests are shown in Table 1. This table shows that there is no current method specifying all of the above-mentioned assessment criteria. For the proposed model, the researcher claims that none of the methods helped with the calculation of test effort before and after adjustment. This will help to obtain modified traceability data, which is very useful for maintenance operations.This research study is inspired by the above-mentioned approaches. Based on some limitations from one over another, as shown in Table 1 this research study has determined to adopt some of the features or criteria from the existing approaches and created a new traceability approach to maintaining the involving artefacts effectively from the testing point of view instead of a requirement.

III. METHODOLOGY

Design and Development Research (DDR) approach was used in this study to produce a traceability model focusing on software change and test effort estimation. This research is classified into three (3) main phases: need analysis, design, and development, and lastly, Evaluation. Table 2 shows the research design phases of the study based on the DDR approach. The DDR approach was first proposed by Richey and Klein in 2007 [14] and is currently being applied in educational research to test theory and validate its practicality. In this study, the implementation of DDR was selected to describe the design and development of the traceability model with test effort estimation to support software changes in the maintenance phase.

DDR was used in this study because it is a systematic study of design, development, and evaluation processes with the goal of establishing an empirical basis for the creation of instructional and non-instructional products and tools, as well as new or improved models that govern their development [15]. Table 2 depicts the study's research design phases based on the DDR technique.

Table 2 Studies based approach to DDR (Richey&Klein,2014) [16]

Phase	Method
Phase 1: Need Analysis	Literature Review
Phase 2: Design & Development	Literature Review, Fuzzy Delphi Method
Phase 3: Evaluation	Modified Nominal Group Technique

A. Fuzzy Dephli Method (FDM)

This phase is to evaluate the proposed model using FDM with twelve(12) expert review. The result of this phase, if rejected from the expert review, will do the updated version of the model.

The Fuzzy Delphi Method (FDM) is a measuring instrument re-branded based on the Delphi technique [17]. The FDM was as an effective measuring tool to solve problems that have uncertainty for a study with the decision made based on the analysis of combining theory with the FDM [17]. Generally, the Delphi technique is used to review and gather opinions as a source of information or to form agreements[18].

Previous literature showed that the FDM is a combination of the conventional Delphi (classic) method and the fuzzy set theory (Fuzzy), which is an expansion of the classical set of theory where the elements in the set were evaluated using the binary set (Yes or No) that allows for a systematic comparison of each item under evaluation. Ragin [19] reported that the fuzzy numbering value is in the range of 0 and 1.

Previous literature state that the Delphi method is a technique and approach used to explore and gather opinion from groups of experts in a structured [18]. However, there is also a weakness in this method where Siraj [20] argues that the reliability of the data is questionable if researchers fail to choose the right expert. She also added that another possible weakness is that the researcher and expert become bored because the iteration of Delphi takes a long time to finish. Such problems found by Bojadziev [21] indicate that the most obvious flaws in the analysis include the methodology of the Delphi process, including a long-lasting testing and iteration period in which leakage and loss of knowledge have occurred. However, the selection of a sufficient number of experts can also be influenced because a limited number of experts have not been able to quantify a major problem [20]. In other words, the opinion of the overall selected experts is unlikely to represent the majority opinion of the experts.

B. Modified Nominal Group Technique (NGT)

The questionnaire was utilised during the evaluation phase of this technique to collect the thoughts as respondents, the opinions of industry and academic users who had engaged in software testing.. The Nominal Groups (NGT) technique is being utilised to evaluate this research.

For decision-making in face-to-face small group discussions, the Nominal Group Technique (NGT) was used. [22]. There are semi-structured and quantitative data collection approaches [23],[24]. In the current study, a more directive semi-quantitative data collection method and a qualitative strategy in which concept acceptance without evaluation (qualitative) was followed by a ranking of priority concepts were employed (quantitative).

When used as a method to assess product satisfaction, however, it can be completely quantitative, and this technique is known as NGT Modified [23]. In this study, the Nominal Group technique (NGT Modified) is utilised to evaluate user satisfaction with this model based on software testing user perceptions. NGT Modified was used in this study to identify the model's usability based on the user's perception of the programme, as well as the acceptance by experts and percentage of each major component, element, and importance of the model's elements. In retrospect, the purpose of employing NGTs is to contribute to and generate problem-solving ideas.. [26].

The respondents directly participated in this study to provide information in the usability evaluation phase are referred to as experts in this study. For academic background, the reply should be the one (1) who has taught software engineering curriculum and is involved with the SE project. In the case of industry, it should be a practitioner involved in a SE project.

The number of participants is highly subjective because it is determined by the study's design and outcome. Previous research used a diverse sample size. Allen et al. [26] proposed a total of 9 to 12 experts for an NGT study. Meanwhile, Harvey and Holmes [27] stated that an ideal review panel would have 6 to 12 members. In their studies, Dobbie et al. [23], Perry Linsley [24], and Williams et al. [28] used 30 to 40, 36, and 92 respondents, respectively. NGT is a method for generating ideas and identifying problems. As a result, 35 respondents from higher learning institutions and enterprises with a background in software engineering were chosen for this study.

IV. RESULTS AND DISCUSSION

The suggested HYTEE (Hybrid Software Change Management Tool with Test Effort Estimation) is composed of many connecting subsystems and interfaces. Several subsystems need the support of other tools integrated into the system. This HYTEE system is divided into the architecture of software traceability, the process of analyzing changes, use case diagrams, modules, and operations to fully understand the design process.The purpose of the HYTEE Model Validation using FDM is to get confirmation from the expert about the element in this model. The Fuzzy Delphi Method (FDM) data analysis study is based on the conditions of the triangular fuzzy number and defuzzification process. Terms for triangular fuzzy number are the threshold value (d), and the percentage of expert consensus where the threshold value (d) for each item (components and elements) measured should be less than or equal to 176 at 0.2 [29] and the percentage of expert group consensus must be more than or equal to 75.0 percent [30].

Threshold value (d) will be analyzed using Microsoft Excel. For the defuzzification process, there is also a condition only the value score fuzzy (A) must be more than or equal to the value of α-cut, which is 0.5 . [31]. The value of the same fuzzy score analyzed using Microsoft Excel.

This aims to evaluate the effectiveness of the proposed HYTEE model to show that it supports the evaluation and analysis of hybrid coverage through change dissemination to support software maintenance activities during regression testing. To achieve this goal, this study used one case study that matches the HYTEE model as well. A controlled experiment with a Model tool was conducted in that case study. Quantitative evaluations are then calculated based on the controlled experiment's scoring outcome using Nominal Group Techniques. The final section addresses the findings of the assessment and how the alternative solution would accomplish this research objective.

A. Model Validation using Fuzzi Delphi Method (FDM)

As part of the design process for this model, 12 experts have been identified to view, discuss, evaluate, and validate the key components required by the HYTEE Model. Each of the expert panels met at a meeting to see the needs of the seven (7) main components of the model. It is very important to discuss whether each of the main components is accepted or rejected and certified on the basis of the consensus of a group of experienced experts directly in the context of the study.

1) FDM Result by Expert Review

The findings of this phase are to involve the HYTEE usability evaluation model. There are 6 tables contained in this phase that involves the analysis of fuzzy Delphi (the threshold),This phase is the process of determining elements in HYTEE Models in value by the expert group. The results of the analysis carried out, and the findings have met the conditions contained in the triangular fuzzy number and defuzzification process. For the conditions contained in the triangular fuzzy number, it involves a threshold value (d) and the percentage of agreement of the expert group. The threshold value (d) obtained must be less than or equal to 0.2 [17] .while the percentage of the expert group agreement also must be more than 75% [17]. For the defuzzification process, the terms used are the α-cut of the score (A) obtained must be equal to or greater than 0.5. It is described as the argument stated by Mohd Ridhuan et al. [17] and Bodjanova [32].

Table 3. Change Management Component

No. Item/ Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		Threshold value, d	*Percentage of Consensus Expert Group,%*	*m1*	*m2*	*m3*	*Score Fuzzy (A)*	
1	B1	0.174	91.7%	0.617	0.783	0.917	0.772	ACCEPT
2	B2	0.206	91.7%	0.750	0.883	0.950	0.861	ACCEPT
3	B3	0.234	91.7%	0.633	0.792	0.908	0.778	ACCEPT
4	B4	0.161	66.67%	0.583	0.758	0.908	0.750	REJECT
5	B5	0.147	100.00%	0.600	0.775	0.925	0.767	ACCEPT

Table 4. Traceability Support Component

No. Item/ Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		The threshold value, d	*Percentage of Consensus Expert Group,%*	*m1*	*m2*	*m3*	*Score Fuzzy (A)*	
1	C1	0.219	91.67%	0.717	0.858	0.942	0.839	ACCEPT
2	C2	0.174	100.00%	0.767	0.900	0.967	0.878	ACCEPT
3	C3	0.196	100.00%	0.700	0.850	0.950	0.833	ACCEPT
4	C4	0.174	100.00%	0.767	0.900	0.967	0.878	ACCEPT
5	C5	0.190	100.00%	0.667	0.825	0.942	0.811	ACCEPT

Table 5. Regression Testing Component

No. Item/ Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		The threshold value, d	*Percentage of Consensus Expert Group,%*	*m1*	*m2*	*m3*	*Score Fuzzy (A)*	
1	D1	0.109	83.33%	0.567	0.750	0.917	0.744	ACCEPT
2	D2	0.220	91.67%	0.767	0.892	0.942	0.867	ACCEPT
3	D3	0.211	91.67 %	0.650	0.808	0.925	0.794	ACCEPT
4	D4	0.1 91	91.67%	0.6 17	0.783	0.917	0.772	ACCEPT

Table 6. Test Effort Estimation Component

No. Item/ Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		The threshold value, d	*Percentage of Consensus Expert Group,%*	*m1/*	*m2*	*m3*	*Score Fuzzy (A)*	
1	E 1	0 .196	100.00 %	0.7 00	0.850	0.950	0.833	ACCEPT
2	E 2	0 .174	100.00 %	0.767	0.900	0.967	0.878	ACCEPT
3	E 3	0 .191	91.67 %	0.617	0.783	0.917	0.772	ACCEPT
4	E 4	0 .206	91.67 %	0.750	0.883	0.950	0.861	ACCEPT
5	E 5	0 .220	91.67 %	0.683	0.833	0.933	0.817	ACCEPT

Table 7. Report Component

No. Item / Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		The threshold value, d	Percentage of Consensus Expert Group,%	m1	m2	m3	Score *Fuzzy (A)*	
1	F 1	0.147	100.00 %	0.800	0.925	0.975	0.900	ACCEPT
2	F 2	0.161	66.67 %	0.583	0.758	0.908	0.750	REJECT
3	F 3	0.070	83.33 %	0.517	0.708	0.892	0.706	ACCEPT

Table 8. Graphical User Interface Component

No. Item / Element		Triangular Fuzzy Numbers		Defuzzification Process				The Consensus of Experts
		The threshold value, d	*Percentage of Consensus Expert Group,%*	*m1*	*m2*	*m3*	*Score Fuzzy (A)*	
1	G 1	0.211	91.67%	0.650	0.808	0.925	0.794	ACCEPT
2	G 2	0.174	100. 00%	0.767	0.900	0.967	0.878	ACCEPT
3	G 3	0.206	91.67%	0.750	0.883	0.950	0.861	ACCEPT
4	G 4	0.191	91.67%	0.617	0.783	0.917	0.772	ACCEPT

In conclusion that all the experts agree that all elements of the overall study stand at HYTEE model status suitable to be used and implemented by consensus expert study.

B. Model Evaluation using Nominal Group Technique (NGT)

Usability evaluation is the process of determining a developed product's applicability and suitability. According to Mack and Sharples[33], the usability of a functional product is a measurement of its capabilities based on its product development goals. In this research study, the measuring of product usability is based on the presentation of a HYTEE model. Respondents will have earned a degree in software engineering, worked as a software tester in the industry, and have at least 5 years of real-world experience in software engineering.

Following that, the responder will complete a questionnaire to assess the usability of the HYTEE model. This respondent will be given a questionnaire to evaluate the model's usability. This technique is acceptable and consistent with Millano and Ullius' [34] argument that the evaluation should be based on the user's enjoyment and perception of a generated model. As a result, it indicates that usability evaluation is just as important in determining if the intended and produced model is capable and suitable for fulfilling model production goals. The researcher employed the Modified Group's Nominal (NGT) techniques to examine the perception and satisfaction with the HYTEE

Model in order to assess its applicability. A total of 35 people were picked and divided into five (5) small groups of seven (7) people each, namely groups A, B, C, D, and E. A census of the people who participated in this usability procedure demonstrates that the numbers are insignificant. This is because researchers are only interested in responders with experience in software testing and maintenance.

The feasibility of selecting the participants was consistent with the reasons advanced by Dobbie et al. [23], who used a notional group technique to examine the model's efficiency (Modified NGT).The participant in this model's usability study is asked to express an opinion and to translate it into the form of the provided usability questionnaires. The relevance of each person's agreement and suitability on the Likert scale will offer a score rating for each item evaluated. The score value will be converted into a percentage to represent the data of any properly assessed item, which may or may not be used to determine the applicability of the HYTEE Model.

Furthermore, the group score must be equal to or greater than 70.0 percent. According to Deslandes et al. [35] and Dobbie et al. [23], a criteria for the nominal group technique (NGT) is that each participant percentage equals or exceeds 70%. A participant completed seven (7) sections of a questionnaire in order to conduct an evaluation of the HYTEE model. Parts A through G address participant demographics, change management, traceability support, regression testing support, test effort estimation, report generation, and the HYTEE Model's graphical user interface, in that order.

Table 9. Demographic Respondents

No.	Item	Detail	Frequency	Percent (%)
1	Sex	Male	14	40
		Female	21	60
2	Category	Academician	5	14
		Public Sector	22	62.8
		Private Sector	8	22.8
3	Working Post	Lecture	3	8.5
		IT Officer	19	54
		Engineer	2	5.7
		Project Manager	3	8.5
		Other	8	22
4	Experience in Software Development Project	Less 1 Year	1	2.8
		1-3 Year	8	62.8
		4-6 Year	1	2.8
		More than 6 Year	25	71.4
		No experience at all	0	0
		Other	0	0
5	Involved in software maintenance	Yes	30	85
		No	5	15
6	If yes, Year of experience in Software Maintenance	Less 1 Year	0	0
		1-3 Year	6	17.1
		4-6 Year	3	8.5
		More than 6 Year	21	60
		No experience at all	5	14
		Other	0	0
		Total	N = 35	100.0

The participant responded to seven (7) sub-sections, including gender, career background, job, experience in a Software Development Project, involvement in software maintenance, years of experience in Software Maintenance, and product and type of work involved in a maintenance project, according to Table 9. In the gender subsection, 21 participants, or 60%, were women, and 14 participants, or 40%, were men. The Academic sector employs five (5) individuals at a 14 percent rate. The Public Sector employs the most people (22 people) at a rate of 62.8 percent, while the Private Sector employs eight (8) at a rate of 22.8 percent. IT officers have the most participants in the post category, with 19 people, or 54%, followed by a variety of positions such as programmers, technical managers, and business analysts, with 8 people, or 22%.

Three (3) people contributed 8.5 percent to Lecturer and Software Project Manager, respectively, while two (2) engineers contributed 5.7 percent. According to the examination, all participants in the usability evaluation phases are professionals, with a minimum of IT officer rank. A prior profession in the same subject is quite beneficial because it indicates an individual's level of knowledge in the field of education. This is congruent with Swanson and Holton's [36] stance, which holds that one (1) is considered an expert when one (1) possesses knowledge and abilities in a certain field. The next section discusses the Software Development Project Experience.

According to Table 9, all study participants had prior experience in software development. According to Berliner [37],[38], a person with more than five (5) years of experience in a field is deemed professional and competent. As a consequence, a total of 25 research participants (71.4%) had more than six (6) years of software development experience.

However, 1 (2.8%) have less than one (1) year of experience, 8 (62%) have fewer than three (3) years of experience, and just 1 (2.8%) have 4 – 6 years of knowledge in software development. Concerning software maintenance expertise, the majority of participants, or 21, had more than six (6) years of experience, six, or 17.1 percent, had one to three years of experience, three, or 8.5 percent, had four to six years of experience, and the remaining five (5) had no experience in the field of computing.

1) NGT Result by User Perseption & Satisfaction

The 35 respondents selected with some work experience were chosen as representatives of them as software practitioners in this evaluation. Based on the assessment, it was found that the proposed models met their requirements and satisfied users. The results of the evaluation indicate that the proposed model and it's software traceability with effort estimation have achieved some positive efficiency and a remarkable understanding of current approaches.

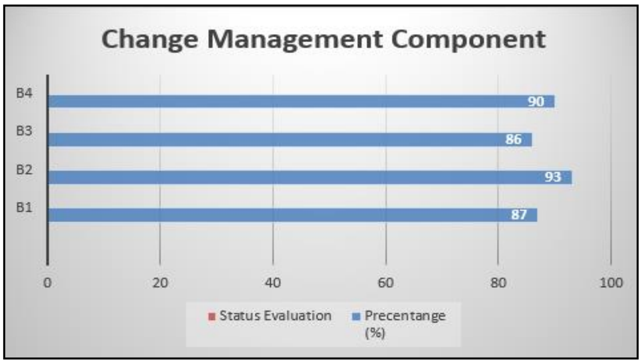

Figure 1. Results of Acceptance Change Management Component

Figure 1 demonstrates that the total score for component B1 in Change Management, based on the responses to the questionnaire, was 153, for an overall value of 87 percent. The overall score for component B2 was 163, representing a 93 percent grade. While B3 has a total score of 152 with 86 percent, B4 has a total score of 158 with 90 percent. According to the participant's opinions and evaluations, the outcome demonstrates that all of the HYTEE model's subcomponent were accepted with a score of at least 70%.

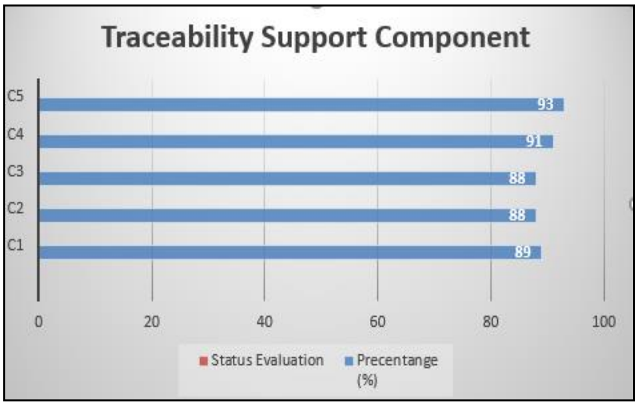

Figure 2. Results of Acceptance Traceability Support Component

Figure 2 demonstrates that the total score of the Traceability Support Component's C1 component, obtained from respondents' responses to the questionnaire, was 157, or 89 percent. The overall score for components C2 and C3 was 155, representing a 91 percent grade. While C4 has a total score of 160 and a percentage of 91 percent, C5 has a total score of 163 and a percentage of 93 percent. Based on the participant's opinions and evaluations, the result demonstrates that all of the HYTEE model's subcomponents with a score of 70 percent or more are approved.

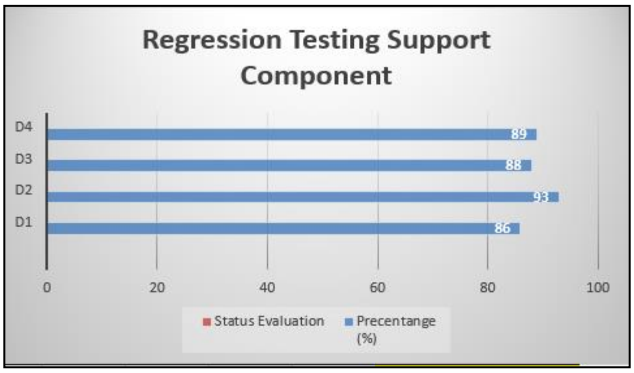

Figure 3. Results of Acceptance Regression Testing Support Component

Figure 3 demonstrates that the total score of the D1 Regression Testing Support Component, determined from respondents' responses to the questionnaire, was 153 with a value of 86%. The overall score for component D2 was 163, representing a 93 percent grade. D4 has a total score of 157 with 89 percent, whereas D3 has a total score of 154 with 88 percent. According to the participant's opinions and evaluations, the outcome demonstrates that all of the HYTEE model's subcomponents were accepted with a score of at least 70%. The outcome of the HYTEE Model for Regression Testing Support Component.

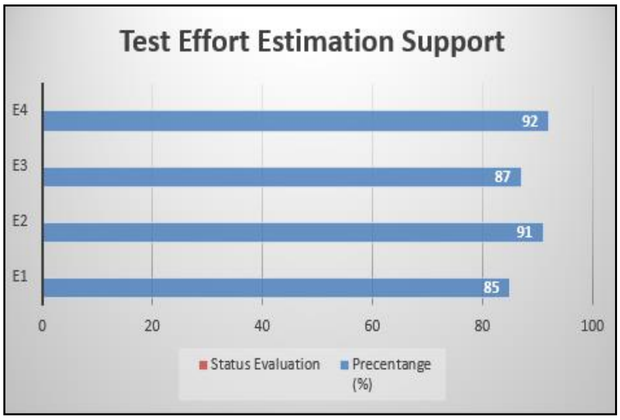

Figure 4. Results of Acceptance Test Effort Estimation Component

The Test Effort Estimation Support Component is depicted in Figure 4. The total score of the component E1 was 150, with a value of 85 percent, based on the views of the respondents as revealed by the results of the questionnaire. The overall score for component E2 was 160, representing a 91 percent grade. While E3 has a total score of 153 with 87 percent, E4 has a total score of 162 with 92 percent. According to the participant's opinions and evaluations, the outcome demonstrates that all of the HYTEE model's subcomponents were accepted with a score of at least 70%. Support Component for the HYTEE Model for Test Effort Estimation Output.

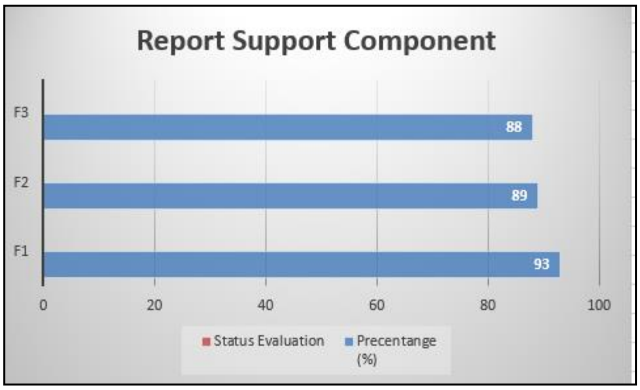

Figure 5. Results of Acceptance Report Supoort Component

Report Support Component is depicted in Figure 5. The total score of the components' F1 was 163, with a value of 93 percent, based on the responses of respondents to the questionnaire's findings. The component F2 total score was 157 with a value of 89 percent. While the final total score for component F3 is 155 and 88 percent, the result demonstrates that all component items in the HYTEE model scored above 70 percent and were accepted based on the participant's opinions and evaluations. HYTEE Model For Report Support Component Output.

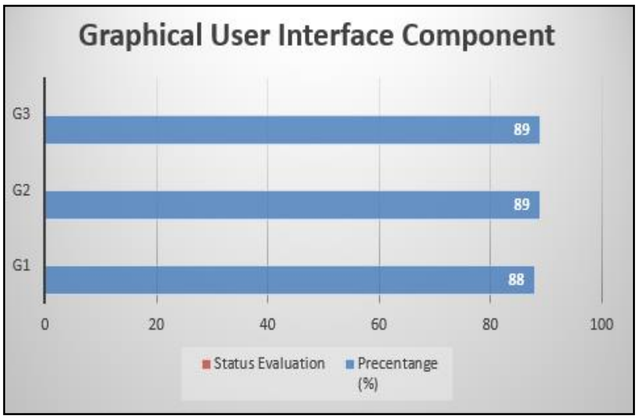

Figure 6. Results of Acceptance GUI Component

The Graphical User Interface Component is depicted in Figure 6. The G1 component's overall score was 155, with a value of 88 percent, based on the perspectives of survey respondents as revealed by the questionnaire's findings. The overall score for components G2 and G3 was 157, representing an 89 percent grade. According to the participant's opinions and evaluations, the outcome demonstrates that all of the HYTEE model's subcomponents were accepted with a score of at least 70%. Component of the HYTEE Model for Graphical User Interface. According to the analysis that can be summarised, the respondent agrees that all of the main components and elements inside the main component of each element for the accomplishment of the status model are adequate for the study participants to utilise HYTEE. The recommended method combines static and dynamic analysis to build a diversified set of software system item dependencies.

V. CONCLUSION AND BENEFIT OF STUDY

The Design & Development Research that was chosen to conduct this study had made it possible to consider the research issues and to address the research questions. A model was developed based on the proposed approach to create links and dependencies within the software program involving the central artifacts; requirements, architecture, test cases, and code. The goal of this model is to create traceability from a test case requirement and to code so that it can recognize the possible impacts on other sections of the system. When such relationships are established, change management can be managed correctly.The proposed model was then refined, and links to traceability were further formed to the specific objects, which included the specifications, test cases, packages, classes, and methods. Software traceability was controlled vertically and horizontally. It has helped handle changes within any artefact and update data on coverage accordingly. The model used dynamic analysis to capture the interconnections between specifications, test cases, and code.

The prototype tool, HYTEE, which is based on our research, was exposed to the subjects by letting them use and evaluate its efficiency and effectiveness under a controlled laboratory experiment. Feedback and comments from users regarding its usefulness and support for software maintenance were taken into account. Some questionnaires were designed to determine if the prototype tool was helpful and effective to support software maintenance. The majority agreed on its overall usefulness.Even when examining the problems, there were tendencies for the researcher to depart from the original planning, even though some slight changes were made during the study path through the planning. Whatever the improvements, this final chapter has shown that all the work targets have been achieved. The consequence of this study is the creation and refining of the HYTEE model, which was a tool for selection and analysis, allowing an understanding of the outcome viewed from the aspect of change approach and measurement of effort.In addition, this study evaluated the impact on HYTEE efficiency of the decision strategy, function, and work-system life-cycle model. The outcome is an example of the strong impact of decision strategy and function on the HYTEE model. The analysis also shows that the proposed model constructs affect the entire program change and effort estimation method very strongly. The research is one of the first attempts at testing function and decision strategy in a life-cycle model of a working system based on software traceability with effort estimation.

REFERENCES

[1] Bennett, K. H., & Rajlich, V. T. (May). *Software maintenance and evolution: a roadmap*. Proceedings of the Conference on the Future of Software Engineering ACM pp **73-87, 2000.**

[2] Nurmuliani, N., Zowghi, D., & Williams, S. *Requirements volatility & its impact on change effort: Evidence based research n software development projects*. In *Verified OK*. University of South Australia. **2006.**

[3] Chen, C.-Y., and Chen, P.-C. *A holistic approach to managing software change impact*. Journal of Systems and Software, Vol ***82*** Issue **(12)**, pp **2051-2067**, **2009.**

[4] Finkelsteiin, A., & Kramer, J. *Software engineering: a roadmap*. Proceedings of the Conference on the Future of Software Engineering, pp. **3-22, 2000.**

[5] Chua, B., Bernardo, D., and Verner, J. *Criteria for Estimating Effort for Requirements Changes*. In R. O'Connor, N. Baddoo, K. Smolander & R. Messnarz (Eds.), Software Process Improvement Springer Berlin Heidelberg, Vol. **16**, pp. **36-46**,**2008.**

[6] Lehtinen, T. O. A., Mäntylä, M. V., Vanhanen, J., Itkonen, J., and Lassenius, C.). *Perceived causes of software project failures – An analysis of their relationships*. Information and Software Technology, pp **623-643. 2014.**

[7] Asl, M. H., and Kama, N. *A Change Impact Size Estimation Approach during the Software Development.* Paper presented at the Software Engineering Conference (ASWEC),pp **68-77. 2013.**

[8] Anooja A, Jameel Ahmad Qurashi, Sanjay Kumar, "*A Survey Study of Various Software Cost Effort Estimation in Perspective of India*", International Journal of Computer Sciences and Engineering, Vol**.7,** Issue**.1,** pp**.928-933, 2019.**

[9] Chua, B. *Rework Requirement Changes in Software Maintenance.* Software Engineering Advances (ICSEA), 2010 Fifth International Conference pp **252-258, 2010.**

[10] Neeraj kumar, Yogesh kumar, Rahul Rishi, "*Software Effort Estimation Techniques*", International Journal of Computer Sciences and Engineering, Vol.7, Issue.1, pp.139-142, 2019.

[11] Sharif, B., Khan, S. A., and Bhatti, M. W. *Measuring the*

Impact of Changing Requirements on Software Project Cost: An Empirical Investigation. IJCSI International Journal of Computer Science Issues. pp **170-174. 2012.**

[12] Singh, M., and Vyas, R. *Requirements Volatility in Software Development Process.* International Journal of Soft Computing, 2. **2012**

[13] Stammel, J., and Trifu, M. *Tool-supported estimation of software evolution effort in service-oriented systems.* Paper presented at the First International Workshop on Model-Driven Software Migration (MDSM 2011), pp **56. 2011.**

[14] Richey, R., & Klien, J. *Design and development research: Method, strategies and issues.* London: Erlbaum .**2007**

[15] Richey, R. C., & Klein, J. D. *Research on design and development.* Handbook of research for educational communications and technology .pp. **748–757. 2008**.

[16] Richey, R. C., & Klein, J. D. *Design and development research*. In Handbook of research on educational communications and technology pp. **141-150, 2014.**

[17] Jamil, M. R. M., Siraj, S., & Hussin, Z. Nurulrabihah Mat Noh., & Ahmad Arifin Sapar Pengenalan Asas Kaedah Fuzzy Delphi Dalam Penyelidikan Rekabentuk dan Pembangunan.*(Mohd Ridhuan Mohd Jamil, Ed.). Kuala Lumpur, Malaysia: Minda Intelek Agency*. **2017.**

[18] Muhammad Imran Yousof. *Using experts' opinions through Delphi technique*. Practical Assessment Research & Evaluation, Vol **12** No **4**, pp **1-8. 2007.**

[19] Ragin, C. C. *Qualitative comparative analysis using fuzzy sets (fsQCA).* In Configurational comparative analysis. London: Sage Publications. **2007**.

[20] Siraj, S. *Kurikulum masa depan*. Penerbit Universiti Malaya. **2008**

[21] Bojadziev, G., & Bojadziev, M.. *Fuzzy Set For Business, Finance and Management.* Singapore: World Scientific Publishing Co. Pte. Ltd. **2007.**

[22] Nasurddin, A. M., & Osman, I. *Pengantar pengurusan*. Utusan Publications. **2006.**

[23] Dobbie, A., Rhodes, M., Tysinger, J. W., & Freeman, J. (). *Using a modified nominal group technique as a curriculum evaluation tool.* FAMILY MEDICINE-KANSAS CITY-, Vol **36**,pp **402-406, 2004.**

[24] Perry, J., & Linsley, S. *The use of the nominal group technique as an evaluative tool in the teaching and summative assessment of the inter-personal skills of student mental health nurses*. Nurse education today, Vol **26** No **4**, pp **346-353 2006.**

[25] Dang, V. H. *The Use of Nominal Group Technique: Case Study in Vietnam*. World Journal of Education, Vol **5** No **4**, pp **14-25, 2015.**

[26] Allen, J., Dyas, J., & Jones, M. *Building consensus in health care: a guide to using the nominal group technique*. British journal of community nursing, Vol **9** No **3**, pp **110-114, 2004.**

[27] Harvey, N., & Holmes, C. *A Nominal group technique: an effective method for obtaining group consensus*. International journal of nursing practice, Vol **18** No **2**, pp **188-194, 2012.**

[28] Williams, P. L., White, N., Klem, R., Wilson, S. E., & Bartholomew,P. *Clinical education and training: jousing the nominal group technique in research with radiographers to identify factors affecting quality and capacity*. Radiography, Vol **12** *No* **3**, pp **215-224, 2006.**

[29] Cheng, C. H., & Lin, Y. *Evaluating the best main battle tank using fuzzy decision theory with linguistic criteria evaluation*. European journal of operational research, Vol **142** No **1**, pp **174-186, 2002.**

[30] Hammons, J. O., & Murry Jr, J. W. *Management appraisal systems in community colleges: How good are they?*. Community College Review, Vol **24,** No **1,** pp **19-28, 1996.**

[31] Tang, C.W. and , Wu, C.T. *Obtaining a picture of undergraduate education quality: a voice from inside the university*, Springer. Higher Education, pp **269-286, 2010.**

[32] Bodjanova, S. *Median alpha-levels of a fuzzy numbe. Fuzzy Sets and Systems*, Vol.**157** no **7,** pp **879 – 891, 2006.**

[33] Mack, Z., & Sharples, S. *The importance of usability in product choice: A mobile phone case study*. Ergonomics, Vol **52** No **12,** pp **1514-1528, 2009.**

[34] Milano, M., & Ullius, D. *Designing powerful training: The sequential-iterative model***. 1998.**

[35] Deslandes, S. F., Mendes, C. H. F., Pires, T. D. O., & Campos, D. D. S. *Use of the Nominal Group Technique and the Delphi Method to draw up evaluation indicators for strategies to deal with violence against children and adolescents in Brazil*. Revista Brasileira de Saúde Materno Infantil, (2004b).Vol **10,** pp **29-37, 2010.**

[36] Swanson, R. A., & Falkman, S. K. *Training delivery problems and solutions: Identification of novice trainer problems and expert trainer solutions*. Human Resource Development Quarterly, Vol **8** No **4,** pp **305-314, 1997.**

[37] Berliner, D. C. *Expert teachers: Their characteristics, development and accomplishments.* Bulletin of Science, Technology and Society, Vol.**24** No **3,** pp **200-12. (2004a).**

[38] Berliner, D. C. *Describing the behavior and documenting the accomplishments of expert teachers*. Bulletin of Science, Technology & Society, Vol.**24** No **3**pp 2**00-212, 2004b.**

AUTHORS PROFILE

Mazidah Mat Rejab is Senior Lecturer at the Universiti Tun Husse in Onn's Faculty of Computer Science and Information Technology. She obtained a BSc and an MSc in Software Engineering from Universiti Teknologi Malaysia (UTM) respectively (UTM). The Razak Faculty of Technology and Informatics at Universiti Teknologi Malaysia awarded her a doctorate in Software Engineering. She has 12 years of expertise in the software engineering sector. She is a member of the Malaysia Board of Technologies (MBOT) for Professional Technologies.

Mohd Ridhuan Mohd Jamil is Senior Lecturer at Faculty of Human Development, Universiti Pendidikan Sultan Idris. He received BEng in Mehanical Engineering at Kolej Universiti Tun Hussein Onn (KUiTTHO) and Med in TVET Education at Universiti Tun Hussein Onn (UTHM). His PhD degree in Engineering Education from Institute for Advanced Studies, Universiti Malaya. He had 14 years teaching experience in Mechanical Engineering and Research Methodology

