Requirement Traceability Model for Agile Development: Results from Empirical Studies

Adila Firdaus Binti Arbain, Dayang Norhayati Abang Jawawi, Wan Mohd Nasir Bin Wan Kadir, Imran Ghani

Abstract: Currently, it is crucial to develop software within the time frame given. Agile software development methodologies offer methods to develop a system in term of time and cost saving but has been criticized for not offering software quality management (a.k.a Non-Functional Requirement, NFR) properly. An empirical case study has been conducted used to find out the need of a traceability approach for NFR change impact in most of Agile software methodology (TANC). TANC is improved and further evaluated by using expert survey analysis method. Based on the results of the expert survey analysis TANC has been proven to fulfil the characteristics of the criteria that needed to be a traceability approach in Agile Software Development for tracing NFR change impact. Thus, this proves that TANC offered better way to trace change impact during the agile development process.

Index Terms: Requirement, Traceability, Empirical Study, Agile Software Development.

I. INTRODUCTION

Traceability approaches and methods have been applied in traditional software development processes such as waterfall [31] and model driven [29] and later started to be introduce in Agile software development projects [6] also. As a result, many researchers have done their researches on agile and traceability [18], [1], [12]. There are some researches that have started to create traceability models and techniques in various Agile software development model such as Scrum [11], [21], [4], [19], FDD [25] and AUP [14]. However, these established traceability techniques in Agile only support the functional requirements, not the NFRs [20]. Furthermore, some researches state that traceability is not compatible with extreme programming (XP) processes [27], [15], [28], [18]. There are many traceability models [2], [5] concept [9], [8] and mechanism [17] that have been proposed in NFR but not one of them are specifically addresses issues in agile-based projects.

Revised Manuscript Received on May 22, 2019.

Adila Firdaus Binti Arbain, Software Engineering Department, Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia

Dayang Norhayati Abang Jawawi, Software Engineering Department, School of Computing, UniversitiTeknologi Malaysia

Wan Mohd Nasir Bin Wan Kadir, Information Technology and Services, Indiana University of Pennsylvania

Imran Ghani, Information Technology and Services, Indiana University of Pennsylvania

However, the issues have not discussed in specifics and categories manner. Therefore, below are the discussions about the specific issues that covered in the coming section.

In fact, after doing intensive literature review on the traceability and Agile projects [10], there are a few main issues that are related to traceability approaches and Agile environment with respect to requirements change impact. Therefore, in this research, the main focus is the problem related to traceability of NFR in Agile projects. First, the focus of main issue is requirements change impact. The requirements change impact could be decomposed to a few sub issues that are propagation issue and consistency issues. In addition, the scope of these two main sub issues are specified in security and performance tracing. Propagation issue have been addressed by a few researchers [32], [7], [33], [3] but mainly the propagation approaches are used in relation to model based or object-oriented based (Chen and Chou, 1999). These approaches are not compatible with Agile process development because they lead to redundancy of traceability process flow. It means that the propagation techniques in most existing traceability is heavy weighted documents, time consuming and repeating flows. Then, another issue is inconsistencies [24], [22]. This issue is important too and must be considered in this research. The issues are that, when changes happen, then the team could track back to see which requirements are affected. The reason is due to the Agile software development environment that always accepting changes during development force the traceability approach must be systematically planned so that it could consistently update the traced and changed artefacts.

Next issue is adaptiveness. This issue is concerned with whether the traceability approaches are compatible with most Agile process development models. This reason is Agile software development that has many processes models such Scrum and FDD. In addition, these models differ by processes, documentation and project scope. In order to simplify the task of a developer, a traceability approaches must be designed to fit most of the Agile software development models. The traceability approaches must be flexible in terms of documentation (features or user stories) and the order of phases that could be fit based on the current Agile projects condition. In addition, there are some researches on NFR traceability techniques in Agile software development recently.

There are NFRs traceability techniques that discussed in Scrum [13], FDD [16], and general for Agile software development models [26] but there quite some unresolved issues and weaknesses in those techniques By looking each one of these issues, we can lead to develop a quality software by solving and tackling each of these problems with traceability approaches, TANC [37]. Besides that, the merge between traceability in NFR with Agile software development environment is difficult, due to definition of NFR that do not have any quantitative satisfaction criteria to measure such as User Acceptance Testing (UAT).

The rest of the paper is organized as follows: Sections 3 presents the detail about TANC Expert Analysis by ranking the TANC approach with 5 point of Likert Scale and run a statistical analysis using Kendall's coefficients of concordance and p-value for TANC. Lastly, Section 4 presents the conclusion of this study.

II. TRACEABILITY MODEL

Basically, TANC is composed of four main components, Strategic, Create, Maintaining and Using trace phase. Figure 10 depicts the order of decisions on how to use TANC in order to trace NFR. It starts with Strategic trace phase where the process of requirement components elicitation from Agile Information Model (AIM) such as user stories and Quality Agile Information Model (QAIM) such as security and performance features, that is used during Create trace phase. Then, create trace phase is where all the trace artefacts are created, trace link are defined and stored in Traceability Information Model (TIM). After Create trace phase, during software development iterations [23], Maintaining trace phase are executed where TIM are updated based on the progress of the development. If there is a need for change, then, the process goes to Using Trace phase. In this phase, TIM storage is used to trace which user stories and NFR affected while changes happen during the development.

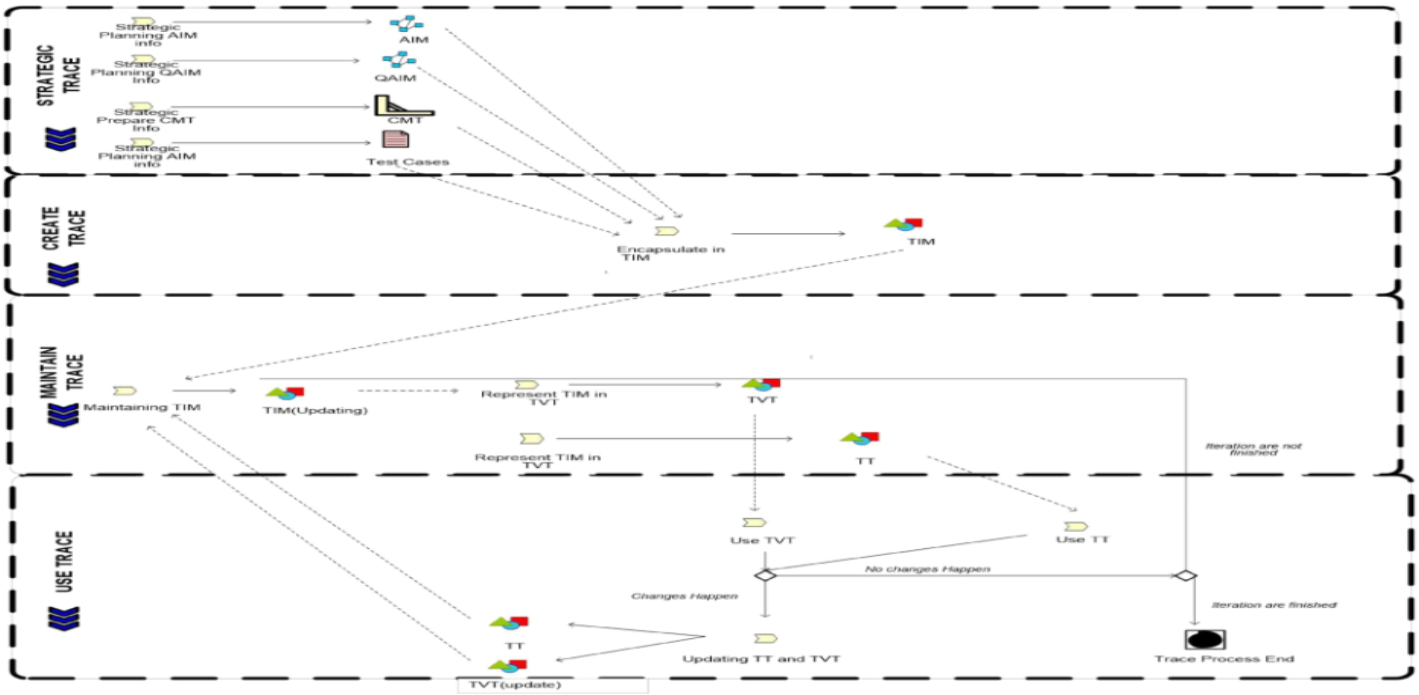

Fig. 1 Final Version of TANC

III. QUALIFICATION OF PANELLISTS

The demographic information for each expert is presented in Table 1 which summarize the knowledge and expertise of the panel. The experts included practitioners and academics that had experience and knowledge in the area of Agile Software Development and/or Requirement Traceability

Table. 1 Experts Distribution

Field	No of Expert	No of Experience (Years)		Roles	No of Projects	
Practitioners	13	Range	0-7	6-Developer 3-Tester 4-Scrum Master/Project Manager	Range	0-5
		Average	2.153		Average	1.46
Academicians	2	3				

Table 1 shows the distribution of experts that involve in the Expert Survey Analysis method. As for the practitioners, they have the range of 0 to 7 number of years' experience in Agile software development and the average of years' experiences of this practitioner are 2.153 years. Meanwhile, for the academicians, one of the academicians has her years' experiences in the domain knowledge of Traceability within 3 years and the other has her years' experiences in the domain knowledge of Agile software within 3 years. Then, the range of total Agile development projects involvements are zero to five projects and the average number of Agile development projects of this practitioners are 1.46 projects.

IV. STATISTICAL ANALYSIS ON TANC'S SUCCESS CRITERIA

The panellists were asked to rate the fulfilment of criteria for TANC, from 'not at all agree' to 'completely agree' for each requirement. This was done using a 5-point Likert scale ranging from 1 ('not at all agree') to 5 ('completely agree'). In order to measure the level of consensus among the experts for the listed criteria, Kendall's coefficient of concordance (W) was used to measure in the survey as the TANC is ready to be ranked. The values are as presented in Table 2. Kendall's coefficients of concordance (W) reached 0.368 and the p-value reached 0.000 (0.000988) for third round (Table 16 is extracted from Microsoft Excel software). Thus, the results of the third-round study were found statistically significant (p-value < 0.05) and consistent.

Table. 2 Kendall's coefficients of concordance and p-value for TANC criteria

k (the number of questions)	7
m (the number of expert)	15
W (Kendall's coefficient of concordance)	0.368968
Chi Sq (degrees of freedom)	36.15889
df (Chi Sq distribution)	14
p-value	0.000988

Table. 3 Level of consensus among the experts for the listed criteria

	C	MP	R	A	G	P	E
N Valid	15	15	15	15	15	15	15
Mean	4.33	3.8	3.6	3.67	3.93	3.8	4.13

Based on Table 3, a total in 15 experts, consist of 13 practitioners and 2 academicians rated TANC based on 7 types of criteria. The first criteria is Change impact (C) and the mean value is 4.33, the second criteria is Multipath (MP) and the mean value is 3.80, the third criteria is Redundancy (R) and the mean value is 3.60, the fourth criteria is Adaptability (A) and the mean value is 3.67, the fifth criteria is Granularity (G) and the mean value is 3.93, the sixth criteria is Propagation (P) and the mean value is 3.80, and the last criteria is Evolvability (E) and the mean value is 4.13. The overall mean value of all the criteria is above 3, which is acceptable. Then, the data collected from each round was analyse using statistical methods. The mean value for each criterion in the model was calculated to measure the satisfaction rate. This result indicates that majority of the SAR satisfied the TANC model.

V. CONCLUSION& FUTURE WORK

The result of the Expert Survey Analysis provided the reliability that the refined approach was acceptable for the expert. Almost positive feedback received by the experts. It shows that TANC is considered acceptable approach based on the focus issues. However, TANC still needs to be enhanced in order to improve the change impact trace technique.

ACKNOWLEDGMENT

This project is supported by Tier 1 Grant Scheme, Universiti Tun Hussein Onn Malaysia (UTHM), and Research Management Centre (RMC), under the Vot Project Number: H130

REFERENCES

1. Abad, Z. S. H., Sadi, M. H., &Ramsin, R. (2010). Towards Tool Support for Situational Engineering of Agile Methodologies. 2010 Asia Pacific Software Engineering Conference, 326–335.
2. Abbors, F., Lilius, J., Joukahaisenkatu, A., Abbors, E. F., &Truscan, D. (2009). Tracing Requirements In A Model-Based Testing Approach, 123–128.
3. Ajila, S. A., Kaba, A. B., &Universite, U. B. De. (n.d.). Traceability Mechanisms, 157–162.
4. Akbari, F., &Sharafi, S. M. (2012). A review to the usage of concepts of software architecture in agile methods. 2012 International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 389–392.
5. Alaa, G. (2014). A multi-faceted Roadmap of Requirements Traceability Types Adoption in SCRUM : An Empirical Study, 1–9.
6. Breivold, H. P., Sundmark, D., Wallin, P., & Larsson, S. (2010). What Does Research Say about Agile and Architecture? 2010 Fifth International Conference on Software Engineering Advances, 32–37.]
7. Chen, X., Hosking, J., Grundy, J., & Amor, R. (2013). Development of Robust Traceability Benchmarks. 2013 22nd Australian Software Engineering Conference, 145–154.
8. Cleland-huang, J., Rahimi, M., &Mäder, P. (2014). Achieving Lightweight Trustworthy Traceability, 849–852.
9. Cleland-huang, J., Settimi, R., Benkhadra, O., Berezhanskaya, E., & Christina, S. (2005). Goal-Centric Traceability for Managing Non-Functional Requirements, 362–371.
10. Dybå, T., &Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and software technology, 50(9), 833-859.
11. Fitzgerald, B., Stol, K., Sullivan, R. O., & Brien, D. O. (2013). Scaling Agile Methods to Regulated Environments : An Industry Case Study, 863–872.
12. Ge, X., Paige, R. F., & McDermid, J. a. (2010). An Iterative Approach for Development of Safety-Critical Software and Safety Arguments. 2010 Agile Conference, 35–43. doi:10.1109/AGILE.2010.10
13. Jeon, S., Han, M., Lee, E., & Lee, K. (2011). Quality Attribute driven Agile Development, 211–218.
14. 14. Jia-ming, Z. (2010). The software development process methodology of resource-based access control. 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 4, 111–117.
15. 15. Jiang, L., &Eberlein, A. (2008). Towards a framework for understanding the relationships between classical software engineering and agile methodologies. Proceedings of the 2008 International Workshop on Scrutinizing Agile Practices or Shoot-out at the Agile Corral - APOS '08, 9–14.
16. Kanwal, F., Junaid, K., &Fahiem, M. A. (2010). A Hybrid Software Architecture Evaluation Method for FDD - An Agile Process Model. 2010 International Conference on Computational Intelligence and Software Engineering, 1–5.
17. Kassab, M., &Ormandjieva, O. (2006). Towards an Aspect-Oriented Software Development Model with Tractability Mechanism, 1–8.
18. Kourie, D. G., & Watson, B. W. (2003). Standards and Agile Software Development, 1–11.
19. Lee, W., Park, S., Lee, K., Lee, C., Lee, B., Jung, W., Wu, C. (2005). Agile development of Web application by supporting process execution and extended UML model. 12th Asia-Pacific Software Engineering Conference (APSEC'05), 8 pp.
20. Tarhan, A., & Yilmaz, S. G. (2014). Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process. Information and Software Technology, 56(5), 477-494.
21. Lehto, I., Tkk, P. O. B. F.-, &Rautiainen, K. (2009). Software Development Governance Challenges of a Middle-Sized Company in Agile Transition, 36–39.
22. Mahmoud, A., &Niu, N. (2013). Supporting Requirements Traceability through Refactoring, 32–41.
23. Moe, N. B., Aurum, A., &Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. Information and Software Technology, 54(8), 853-865
24. Paige, R. R. R. F. (2010). Iterative criteria-based approach to engineering the requirements of software development methodologies, (April 2009).
25. Passos, L., Apel, S., &Kästner, C. (2013). Feature-Oriented Software Evolution Categories and Subject Descriptors.
26. Rao, A. A., & Gopichand, M. (2011). Four Layered Approach to Non-Functional Requirements Analysis, 8(6), 371–379.
27. Reed, K., Damiani, E., Gianini, G., & Colombo, A. (2004). Agile management of uncertain requirements via generalizations. Proceedings of the 2004 Workshop on Quantitative Techniques for Software Agile Process - QUTE-SWAP '04, 40–45.
28. Taromirad, M., &Ramsin, R. (2008). An Appraisal of Existing Evaluation Frameworks for Agile Methodologies. 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008), 418–427.
29. Tekinerdogan, B., Hofmann, C., &Aksit, M. (2007). Modeling Traceability of Concerns in Architectural Views.
30. Omoronyia, I., Sindre, G., &Stålhane, T. (2011). Exploring a Bayesian and linear approach to requirements traceability. Information and Software Technology, 53(8), 851-871.
31. Vanderleest, S. H., College, C., & Rapids, G. (2009). ESCAPE THE WATERFALL: AGILE FOR AEROSPACE, 1–16.
32. Arlow, J., Neustadt, I.: UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design, 2nd edn. Addison-Wesley, Boston, MA (2005).
33. Gotel, O. C., and Finkelstein, A. C. (1994, April). An analysis of the requirements traceability problem. In Requirements Engineering, 1994., Proceedings of the First International Conference on (pp. 94-101). IEEE
34. Duffield, C. (1993). The Delphi technique: a comparison of results obtained using two expert panels International Journal of Nursing Studies 30(3): 227-237.
35. Powell, C. (2003). The Delphi technique: myths and realities. Journal of Advanced Nursing 41(4): 376-382.
36. Skulmoski, G. J. and Hartman F. T. (2007). The Delphi Method for Graduate Research. Journal of Information Technology Education 6(1): 1-21.
37. Arbain, A.F., Ghani, I., Jawawi, D.N.A., Kadir, W.M.N.W. (2015). Logical Model for Synchronizing TANC with XP. KSII The 7 th International Conference on Internet (ICONI) 2015.
38. Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. Information and software technology, 50(4), 280-295.
39. Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying SCRUM principles to software product management. Information and software technology, 53(1), 58-70.
40. Laanti, M., Salo, O., &Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology, 53(3), 276-290.