Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REQUIREMENT TRACEABILITY MATRIX THROUGH DOCUMENTATION FOR SCRUM METHODOLOGY

GUNAVATHI DURAISAMY AND RODZIAH ATAN

Department of Information System, Universiti Putra Malaysia, Serdang, Selangor, Malaysia

ABSTRACT

Requirement traceability matrix is a table that captures the complete user and system requirement for the system. It helps to trace from requirement till testing in order to verify that the requirement is fulfilled. In SCRUM development methodology, requirement traceability matrix is used to capture the linkage of user stories between product backlog and sprint backlog. The linkages between the requirements are retrieved through these two documents. However, unstructured format of both documents do not help in getting the requirement traceability. Thus, requirement traceability has become an issue for SCRUM practitioners especially for system development and maintenance. Therefore, this study will introduce structured format of available artifacts and develop a tracing tool to automatically generate the requirement traceability matrix by keyword searching. Both the documents used in this study have to be prepared by using the structured format and the proposed traceability tool is able to generate the requirement traceability matrix automatically by keyword searching functionality. The result shows that the introduced structured format is very useful and it has increase the efficiency of retrieving the matrix far better than previous process. There is a significant time saved up to 95% for generating the requirement traceability matrix using the proposed method and tool. As a conclusion, requirement traceability can be achieved in SCRUM methodology through the proposed structured documentation and the tool developed.

Keywords: *Requirement Traceability Matrix, SCRUM Development Methodology, Structured Documentation, Forward Traceability, Backward Traceability, Correctness, Time Saved*

1. INTRODUCTION

The complete user and system requirements for the system are captures in a table called requirement traceability matrix. This table helps to trace the requirements from user requirement to the test cases in order to verify that the requirements are fulfilled. Requirement traceability matrix is used to capture the linkage of user stories between product backlog and sprint backlog in SCRUM development methodology. Lightweight documentation practice of SCRUM leads to having unstructured format of documentation for both product and sprint backlog. Thus, there are no proper linkages between these two documents and requirement traceability has been an ongoing issue for SCRUM practitioners.

Traceability is defined as "an ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e., from its origin, through its development and specification, to its subsequent deployment and use" (Gotel and Finkelstein, 1994). Whenever large and/or complex software systems are to be maintained, traceability support is required and it may become critical to project success (Watkins and Neal, 1994). Brad Appleton, B (2005) identified several reasons for tracing in the Agile project. Change impact analysis and product conformance are the two top important reasons for traceability in Agile. However, Agile developers view traceability as a heavyweight and burdensome activity (Appleton, B. 2005; Clelang-Huang 2006).

SCRUM is one of the famous Agile methodology which produces a product during small cycles called iterations and its functionality of product increases during each iterations by adding new properties. As the requirement changes frequently at the end of each iteration, these changes have to be considered to be included in the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

next iteration. There are only two documents (product backlog and sprint backlog) which capture the requirement in the user stories form. The linkages between the requirements are through these two documents which are used for the change request impact analysis.

Lightweight development processes, frequent deliverables and minimal documentation (Ambler, 2004) are the characteristics of SCRUM where the prepared documentation are mostly in informal structure. There are no proper identifications of the requirements in such unstructured documents. Unique identifications are important to create a link in between requirements and artifacts. Traceability is described by the "link that connect related artifacts" (Ramesh and Jarke, 2001). Since the sprint backlog depends on product backlog, there should be a linkage between these two documents. By using these links, we can trace the relations that exist among the documented requirements (Lago et al., 2008) and the requirement traceability matrix is generated from there.

In this paper, we introduce a tracing method by proposing a structured format to prepare product backlog and sprint backlog. We have further developed an automated tracing tool to generate the requirement traceability matrix by keyword searching functionality.

The remaining of the paper is structured as follows: In section 2, we present the literature review on related word of this research. In section 3, we present the research methodology used in this work and the experimental setup for the case studies. Section 4 presents the findings and discussion of the results. Finally, in Section 5, we conclude the findings and direction for future work.

2. LITERATURE REVIEW

Cleland-Huang (2006) stated that requirement traceability helps developers to control and manage the development and evaluation of the software system. Gotel and Finkelstein (1994) defined traceability as the "ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e., from its origin, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases".

SCRUM is one of the preferred Agile methodologies. There are four Agile manifestos published by a group of software practitioners and consultant in 2001 (Beck et al, 2001; Cockburn, 2002). One of the manifestos emphasize was on 'Working Software over Comprehensive Documentation". Agile methods do not rule out the use of documentation though it is fair to say that the role of externalized knowledge is much reduced (Turk et al, 2005 P70). For many Agilest, the broader goal of forming a sound understanding of user requirement is of paramount importance. Therefore, to achieve this study they will use various combinations of face-to-face interactions and the documentation delivers it in the most efficient way (Fowlers and Highsmith, 2001). Since, it has been emphasized on minimal documentation, SCRUM is called as a lightweight documentation methodology. The requirements in user stories form are documented in two main documents of the SCRUM which are product backlog and sprint backlog.

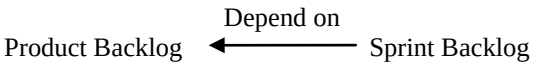
Appleton, B (2005) listed several reasons for tracing in an Agile project. Cleland-Huang et al. (2007), stated that Agile traceability goals are similar to those found in non-Agile projects, but differ in the way on which they are achieved. According to Appleton (2005), many tracing goals can be satisfied through trust and communications in a project if looked beyond the traditional traceability technique's limitations.

Lago et al. (2008), commented that traceability is expensive to maintain, which is similar to the general problems of software documentation and if traceability links are kept manually, they tend not to be updated or just forgotten as soon as some development deadline approaches. Therefore, as cited by Cleland-Huang (2005), automation can definitely contribute in maintaining the traceability.

Lago et al. (2008), also mentioned that traceability is described by the link that connect

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

related artifact. The requirements are linked through product backlog and sprint backlog in SCRUM. As explained by Ramesh and Jarke (2001) and Lago et al. (2008), we noted that the relation among the documented artifacts can be traced using these links. Sprint backlog depends on product backlog as depicted in the model below:



From the literature reviews conducted, we could perceive that even though SCRUM practice lightweight documentation, it is imperative to have requirement traceability. From the previous studies, we found that the traceability link can be retrieved from the prepared documentation. We also learned that the documentation should have a unique identifier to link the requirements from both the documents. By using an automated tool, we are able to access the link easily and generate the requirement traceability matrix by keyword searching.

3. METHODOLOGY

The research design used in this study is 'One Group Pretest-Posttest' design. A single group was involved in the pre-experimental evaluation. Then the experimental treatment was administered. Finally, the same group was evaluated after the experiment. This design is represented as follows:

Group	Time →		
Group 1	Obs	Tx	Obs

Obs = Observation Tx = Treatment

3.1 Source Of Data

The group of people involved in this study are experienced employees of a software warehouse. This organization provides services on information technology application development, system/application maintenance and infrastructure management. There are more than 40 employees working in this organization from various levels. The rationale of selecting this organization in this study is because it is certified in CMMI level 3 and practice SCRUM methodology in their software development process.

For the experimentental design, we selected 5 people (1 project manager, 2 system developers and 2 testers). They are the people who are involved in collecting, managing requirement and preparing the documents such as product backlog and sprint backlog. The rationale behind selecting them for the pretest and posttest experiments is because the selected group of people are the one updating the documents frequently and using them in their daily project development task. Besides this, for the purpose of this study, this group of people are also involved in the interview session conducted to gather information on problem faced, user acceptance testing on the developed tool and the users feedback after the tool implementation. Two main documents were used in this study are product backlog and sprint backlog.

3.2 Metrics And Attributes

Metric: Traceability

Definition: Forward and backward traceability by using the keyword from the documents

Attribute:

1. Number of records retrieved for forward traceability using keyword for each column
2. Time taken to generate matrix for forward traceability using keyword for each column
3. Number of records retrieved for backward traceability using keyword for each column
4. Time taken to generate matrix for backward traceability using keyword for each column

3.3 Experimental Setup

Traceability metrics measures the process of generating requirement traceability matrix by manual process and by using the automated tool. Number of documents used, number of columns

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from the documents used to generate the matrix and keywords used are the attributes for the measurement. Number of records retrieved for both process using the same attributes have been recorded for both backward and forward traceability.

3.4 Case Studies

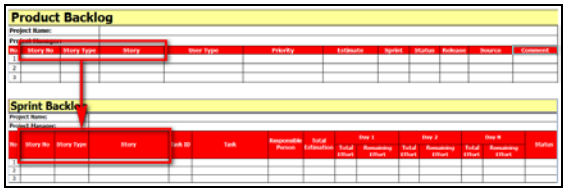
In this section, we presents a detailed description of the case study used in the research. It describes the case study design, execution and the data collected. There are two case studies in this research. VCR is a web based system used to generate report for custom related record. SAMS is a Sport Administrative Management System which use client – server architecture.

The first case study for this research is a project named VCR. This is a web-based application that is used to generate the custom's report for a semiconductor company. It is a newly initiated project and follows SCRUM methodology for the development process. The team members involved in this project are 1 project manager, 2 system developers and 2 testers. The project timeline was 3 months.

The second case study for this research is a project named Sport Administrative Management System (SAMS). It is a client-server based application which is used by the sport complex to manage the booking and payment for the courts. It is a new project handle by the same team and using SCRUM as their development methodology.

3.5 System Design

System design started with designing the structured format for the document. Two documents were analyzed and used in this study which are product backlog and sprint backlog. A structured format was proposed and users were to follow the format to prepare the documents and save the documents with the proposed naming convention as well. The figure shows the structured format and the linkage between both documents.

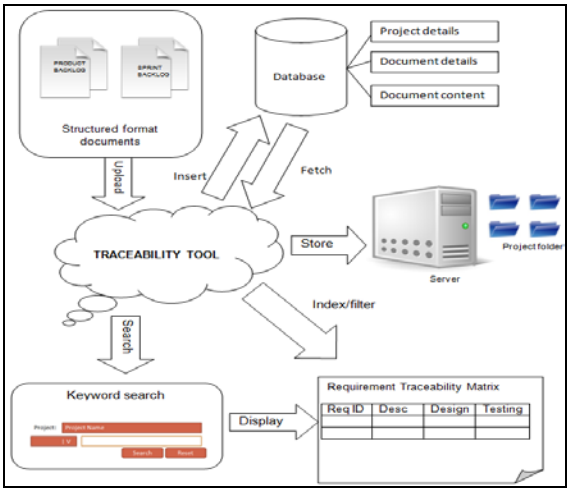


The document then will be uploaded to the traceability tool according to the project it belong and stored in designated location path according to the project in the server.

The project information, the document information and the document content will be inserted into the database according to the column and table designed upon uploading the document in the traceability tool. The database contains table for each item and each project will be assigned with a unique id for tracing purposes. The document will be inserted to the table together with the assigned project id. The archived document will be moved to archived folder and table when user deletes the documents from the tool.

There will be a module for the user to search by keyword and this will generate requirement traceability matrix automatically. Users can select the field as per defined in the document. The field will be listed in a drop down listing for selection. Users will be keying in the keyword in given field. The tool then will search for the match of the keyword entered from the field selected in database.

After finding the match and other relevant details, the tool will generate the requirement traceability matrix and pass it to the matrix. The requirement traceability matrix will be generated automatically and show the details fetched from the database on order after indexed and sorted.



4. RESULTS AND DISCUSSIONS

Below table and figure show the results for Traceability and Attribute 1: Number of record retrieved for forward traceability using keyword for

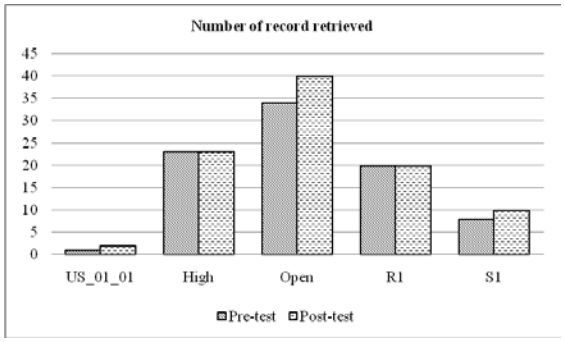
Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

each column. The correctness (%) was calculated based on the formula below:

$$\text{Correctness (\%)} = \left(\frac{\text{Number of record retrieve}}{\text{Number of total correct record}} \right) * 100$$

No	Column	Keyword	Total correct record	Pre-test		Post-test	
				No of record retrieve	Correctness (%)	No of record retrieve	Correctness (%)
1.	Story No	US_01_01	2	1	50	2	100
2.	Priority	High	23	23	100	23	100
3.	Status	Open	40	34	85	40	100
4.	Release	R1	20	20	100	20	100
5.	Sprint	S1	10	8	80	10	100

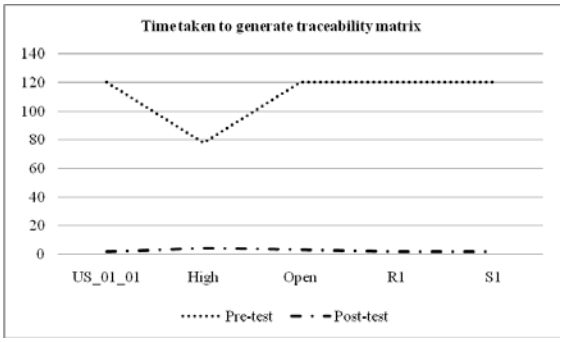


Traceability metric was measured by calculating the number of record retrieved for forward traceability (from product backlog to sprint backlog) using keyword for each column. The same keyword was used for both pre-test and post-test and number of record retrieved was recorded. Based on the results gathered, there were keywords showing positive differences of records retrieved from pre and post test. This indicated that the proposed traceability tool retrieved more number of records than the pre test. From here, it was observed that by using the traceability tool the correctness of records retrieved is better than the manual process.

Below table and figure shows the results for Traceability and Attribute 2: Time taken to generate matrix for forward traceability using keyword for each column. The time saved (%) was calculated based on the formula below:

$$\text{Time saved (\%)} = \left(\frac{\text{Time taken for pre-test} - \text{Time taken for post-test}}{\text{Time taken for pre-test}} \right) * 100$$

No	Column	Keyword	Time Taken(sec)		Time saved (sec)	Percentage of time saved (%)
			Pre-test	Post-test		
1.	Story No	US_01_01	120	2	118	98.33
2.	Priority	High	78	4.3	73.7	94.49
3.	Status	Open	120	3.2	116.8	97.33
4.	Release	R1	120	1.6	118.4	98.67
5.	Sprint	S1	120	1.6	118.4	98.67



The same column and keywords were used to record the time taken to generate the traceability matrix for both pre and post test. From the results, it was very obvious that the traceability tool generated the traceability matrix within seconds while manual process took minutes. The difference rates were more than 95% faster for each of the keyword. The forward traceability matrix was generated with an average of 4 seconds by using the traceability tool.

5. CONCLUSION

The main goal of this study was to adapt traceability in SCRUM methodology from its available documentations. This study has introduced a structured format for documenting the requirement in product backlog and sprint backlog. Traceability tool which enable user to trace the requirement by keyword searching function and generate the requirement traceability matrix automatically has also been introduced by this study. The study proves that SCRUM documents can be used to trace requirement and with the solution provided it has proved that it has increase the effectiveness and it is very useful to the users.

There are areas that can be improved and extended as future work. The future works suggested for this study are as below:

- To extend the scope to other methodology in Agile
- To extend the scope to have system testing and product release information in the traceability matrix

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1]. Ambler, S. (2004) The Object Primer: Agile Model-Driven Development with UML 2.0

[2]. Appleton, B. (2005) ACME Blog: Traceability and TRUST-ability. Accessed June 2011 from http://bradapp.blogspot.com/2005/03/traceability-and-trust-ability.html

[3]. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grennung, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development. http://AgileManifasto.org March 2002

[4]. Cockburn, A. (2002). Agile Software Development. Boston, Addison- Wesley

[5]. Cleland-Huang, J. (2011) Traceability in Agile Projects. Software and Systems Traceability. Springer

[6]. Cleland-Huang, J. (2006) Just enough requirements traceability. COMPSAC **1**, 41-42

[7]. Cleland-Huang, J., Berenbach, B., Clark, S., Settimi, R. & Romanova, E. (2007) Best practices for automated traceability. IEEE Comp. **40**(6), 27–35 ISSN:0018-9162

[8]. Fowler, M., & Highsmith, J. (2001, July 16). The agile manifesto. Retrieved November 21, 2006 from http://www.ddj.com/dept/architect/184414755.

[9]. Gotel, O. & Finkelstein, A. (1994) An analysis of the requirements traceability problem. In: Proceedings of the International Conference on Requirements Engineering (RE), pp. 94–102. IEEE Computer Society, Springs, Colorado

[10]. Lago, P. , Muccini, H., Vliet, & H.V,. (2008) A scoped approach to traceability management, pp. 168-182. The Journal of System and Software 82.

[11]. Ramesh, B. & Jarke, M. (2001) Toward reference models for requirements traceability. IEEE Trans. Softw. Eng. 27(1), 58–93

[12]. Turk, D., France, R., & Rumpe, B. (2005). Assumptions underlying agile softwaredevelopment processes. Journal of Database Management 16(4), 62-87 [Electronic version]. Retrieved September 5, 2006 from http://proquest.umi.com.

[13]. Watkins, R., Neal, M., (1994). Why and how of requirement tracing. IEEE Software 11 (4), 104-106.