

Development of scenario management and requirements tool (SMaRT): towards supporting scenario-based requirements engineering methodology

Azlin Nordin¹, Amirul Ikhwan Omar¹, Megat Usamah Megat Mohamed Amin¹, Norsaremah Salleh¹

¹Department of Computer Science, Kulliyah of Information and Communication Technology
International Islamic University Malaysia
50728 Gombak, Kuala Lumpur, Malaysia

*Corresponding author E-mail : azlinnordin@iiu.edu.my

Abstract

Scenario-based methodology has been applied in analyzing real world experience into representation in software environment. Nonetheless, capturing, documenting and managing scenarios are extremely labor-intensive. Hence, the generic aims of this project are to investigate, explore and analyze requirements management process towards supporting scenario-based requirements approach. Based on our analysis of the existing RE or Requirements Management (RM) tools, most of the tools had less focus or missing some significant RE features to facilitate scenario-based methodology. We analyzed all the relevant features and developed a prototype i.e. Scenario-based Requirements Tool (SMaRT) to demonstrate how scenario-based approach can be implemented to further supports scenario-based RE methodology and covers most of the RE process i.e. elicitation, analysis and negotiation, documentation, validation, and management. SMaRT improves the RE tool features i.e. project and requirements management, document generation and validation function to further supports scenario-based RE methodology and covers most of the RE process i.e. elicitation, analysis, negotiation, documentation, validation, and management.

Keywords: : Requirements Engineering process; Scenario Management and Requirements Tool; Scenario-based requirements methodology.

1. Introduction

A huge demand in software industries results in a very competitive environment between software companies. Customers deemed for a quick but complete product, thus requires software companies to manage their projects efficiently and effectively. In the last three decades, researchers in Computer-Aided Software Engineering (CASE) have tried to find solutions to help system developers by providing integrated system development tools that support SE [1]. Since then, RE tools shows rapid development to provide wider and greater support for different methodologies.

Scenario-based methodology has been advocated as one of the most effective means of communication between users and stakeholders, and the requirements were analyzed in real world experience [2]. Nonetheless, scenarios are extremely labor-intensive to capture and document because the requirement engineers need to walkthrough each possible event sequences; and consider all possible exceptions and errors that may occur at each step [2]. Thus, an alternative that can reduce these efforts will surely improve the effectiveness of this methodology.

Amongst many of the existing RE tools, only a small portion of them support scenario based methodology such as Case Complete[3] and Yakindu [4] (focuses on requirements traceability). In fact, these tools even allow automatic generations of analysis models such as activity diagram from use case specifications. However, the focal points of their work were not to derive requirements from scenarios but rather forwarding the scenarios to

the following tasks such as analysis models or specifications. The detailed discussion on the existing RE tools will be presented in Section 2.

Based on our understanding, none of the existing work focusing on deriving requirements using scenarios, as motivated by scenario-based methodology whereby requirements are elicited from scenarios. SMaRT has seven objectives that are (1) to allow users to manage project requirements systematically; (2) to provide interactive medium for users to derive and manage scenarios; (3) to suggest for users to auto generate requirements from scenarios; (4) to provide functions for users to systematically analyze requirements (i.e. necessity, completeness and feasibility check); (5) to enable requirements negotiation between system developers and stakeholders; (6) to enable users to generate documentation (scenarios, requirement and traceability report); and (7) to allow users to validate requirements online.

This paper is organized as follows. In Section 2, the related work on existing RE tools are analyzed and presented. The methodology adopted in this work is discussed in Section 3. Next, discussion and future work are presented in Section 4. Finally, the conclusion is included in the last section.

2. Related Work

In this work, a comparison between 15 existing RE tools has been made to find out the features for the RE tool support. These tools are selected based on accessibility and availability of resources.

References to the 15 RE tools that are used in the comparison: HP Quality Centre [5], RaQuest [6], Requirement One [7], Rommana [8], Code Beamer [9], Leap SE [10], OneDesk [11], ReqLine [12], Reqtify [13], Case Complete [3], iConcur Axiom [14], Gatherspace [15], Polarion [16], Jama [17] and Yakindu [4].

Amongst the 15 RE tools, only ReqLine [12] and iConcur Axiom [14] tools are open-sources. It is also worth to note that all the tools have an offer of a free trial with a limited usage-period or functionalities. The comparisons between these tools were made based on methodologies and categories they supported. The comparison between 15 RE tools on methodology support is shown in Table 1.

Table 1 indicates that most RE tools support Agile methodology (that is 12 out of 15 tools under investigation). Surprisingly, it was noticed that only 2 tools (i.e. Case Complete and Yakindu) support scenario-based methodology even though based on a research, scenario methodology is becoming a trend among requirement engineers due to its ability to forecast future conditions [18].

Table 1: Comparison of Existing RE Tools based on Its Supported Methodologies.

Tool	Methodologies						
	Agile	Iterative	Waterfall	Use Case Oriented	Goal Oriented	Report Based	Scenario Based
Quality Centre	/	/	/		/	/	
Raquest	/						
Requirement One	/	/	/				
Rommana	/			/		/	
Code Beamer	/		/				
Leap SE	/			/			
OneDesk	/	/	/				
ReqLine	/						
Reqtify	/						
Case Complete				/			/
iConcur Axiom	/			/			
Gatherspace		/				/	
Polarion	/					/	
Jama	/	/	/				
Yakindu				/		/	/

Based on RE process i.e. analysis and negotiation, documentation, validation, and management [23], all the RE tool features were categorized into seven categories that are (1) project management, (2) scenario management, (3) requirements management, (4) analysis, (5) negotiation, (6) documentation generation and (7) validation [19]. The comparison between 15 RE tools based on these seven core processes are shown in Table 2.

Table 2 indicates that most RE tools support project and requirements management (15 tools); document generation (14) while lack of them support validation (7); analysis (6); scenario management (3); and negotiation (1). A more detailed comparison is shown in Table 3. See Appendix.

Table 2: Comparison of the Existing RE Tools based on Categories

Tool	Project	Scenario	Requirement	Analysis	Negotiation	Documentation	Validation
Quality Centre	/		/			/	/
Raquest	/		/			/	
Requirement One	/		/			/	
Rommana	/		/			/	/
Code Beamer	/		/			/	
Leap SE	/		/			/	
OneDesk	/		/	/		/	/
ReqLine	/		/	/			
Reqtify	/		/			/	
Case Complete	/	/	/	/		/	/
iConcur Axiom	/		/	/		/	
Gatherspace	/		/			/	
Polarion	/		/			/	/
Jama	/		/			/	/
Yakindu	/	/	/	/		/	

3. Methodology

The methodology that is used for SMaRT development is the Evolutionary Prototyping as depicted in Figure 1 [20]. The processes of evolutionary prototyping are iterative in nature and involve (1) planning, (2) analysis, (3) design, (4) prototype, (5) implementation, (6) testing, (7) validation and (8) management. By using the evolutionary prototyping, the system being built can continuously be refined and rebuilt throughout development phase.

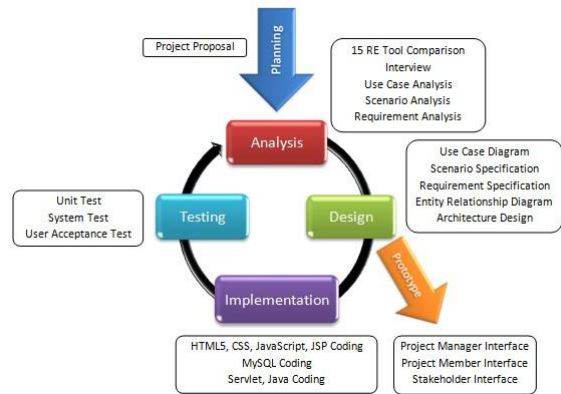


Fig. 1: Evolutionary Prototyping SE methodology [20]

In addition, by using the evolutionary prototype, developers can focus on developing some parts of their program that are less complicated than to develop the system as a whole. This can minimize risks during the development phase. Moreover, this methodology also allows the developers to add new features that were not discovered during earlier phases i.e. requirements and design phases.

Evolutionary prototyping is a general software engineering methodology that is relevant for most of software development project. In the analysis phase, Scenario based Requirements Engineering Methodology (SREM) [24] is adopted as shown in Figure 2.



Fig. 2: Scenario based RE methodology [24]

In SREM, scenarios are used with prototypes as a mechanism for requirements elicitation. A motivation to use scenarios to elicit requirements is that it allows reasoning about problems that may happen when users use the systems. This approach integrates elicitation and validation of scenarios by providing the context for the user to assess a design, which itself is representing a scenario of use case. SREM starts by doing domain analysis [21]. Domain analysis is the process of analyzing software system based on their domain to find their common and variable parts. The main goal of domain analysis is software reuse. Software reuse improves software development life-cycle, increases product quality and reduces maintenance costs.

In the domain analysis stage, interview session was commanded to gain more understanding about the scope and features to be developed. A qualitative methodological approach was used in this research in order to gain understanding of the RE domain. Data was gathered through a structured interview with an RE expert to determine the necessary functions in which an effective RE tool should have.

Data gathered from the interview were analyzed and represented in a Use Case Diagram (UCD). The UCD is validated for correctness with respect to its syntax and semantics. As a result, a total of 59 use cases were categorized into 13 features as listed in Table 4.

Table 4: No of Use Cases and Requirements in Each Module.

Feature	No. of Use Cases	No. of Requirements	Feature	No. of Use Cases	No. of Requirements
Registration	3	4	Analysis	2	2
Login & Logout	4	4	Documentation	5	5
Project	6	5	Dictionary	6	6
Scenario	7	8	Search	2	2
Requirements	6	6	History	1	1
Validation	5	5	Account Setting	9	9
Negotiation	3	3			

After that, use case diagram was used as a basis to create scenarios. Next, by walking through each possible event sequences in the use case, scenarios were generated. Next stage, scenarios were used as a validation tool to create requirements that are appropriate for users. Scenarios were generated by walking through each possible event sequences in the use cases considering all possible exceptions and errors that may occur at each step. Eventually, 13 scenarios were identified and documented. The scenario specifications are provided in the project documentation. The output of SREM is a set of use cases, scenarios and requirements specifications.

The architecture design for SMaRT is based on three-tier Architecture [21]. The three-tier architecture for Web development as shown in Figure 3 is divided into (1) Client applications, (2) application server and (3) data source. The top-most level is the client application. In this layer, it displays static content, and some cached dynamic content. In simple term, it is a layer, which users can access directly from their browsers or an operating systems GUI.

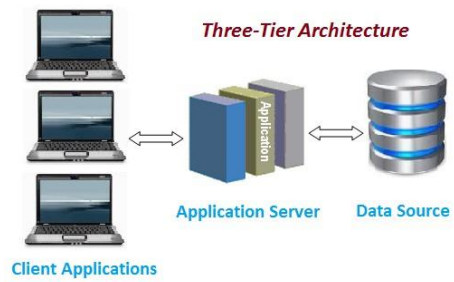


Fig 3: Three-Tier Architecture Design [21]

Next, the application server or also known as logical tier controls the applications functionalities by performing detailed processing based on users input from the front end. Examples of middle dynamic content processing are Ruby on Rails, Java EE, PHP and Python. In SMaRT, Java EE was implemented as middle dynamic content processing. The final part is the back-end, which is also known as the database of the system. This tier is where the systems store all its data and the database management system software manages and provides access to the data. In SMaRT, MySQL were used as DBMS to manage users' data.

Entity Relationship Diagram (ERD) shows the total numbers of table associated with the system and the relationships between them [22]. The relationships can be in many-to-many, one-to-many and many-to-one. In SMaRT, 18 tables had been identified.

The user interface design was developed by using Moqups online prototyping tool as shown in Figure 4 [23]. The figure shows an example of prototype design for *Requirement* page, where the top of the page shows a navigation bar consists of main functions of the SMaRT, i.e. project, scenario, requirement, analysis and documentation. In addition, a tab is located below the navigation bar, in this case, the *Requirement* page consists of two tab, which are *Main* and *Add* tab. The *Main* tab shows the requirements that have been created and the *Add* tab allows users to add new requirements.

The screenshot shows a web interface for the 'Requirement' page. At the top is a navigation bar with tabs for 'SMaRT', 'Project', 'Scenario', 'Requirement', 'Analysis', 'Documentation', and 'Others'. Below the navigation bar is a sub-navigation bar with 'Main' and 'Add' tabs. The main content area displays a table of requirements with columns: Req ID, Name, Type, Author, Priority, and Status. The table contains 14 rows of data, with the last row (R-014) highlighted in blue.

Fig. 4: Prototype of Requirements page

SMaRT reduces the redundancy process and allows efficient use of time and effort, where users i.e. requirements engineers could generate requirements from the added scenarios, which makes it specifically different from all other existing RE tools under the authors investigation. SMaRT is an RE tool, which provides interactive web-based system that supports requirement engineers to input, manage, view, analyze, and work with scenarios and auto generate them into requirements instead of rewriting the requirements from scratch.

The tool also supports greater functionality compared to previous RE tool by covering most of the RE process i.e. elicitation, analysis and negotiation, documentation, validation, and management [19].

After designing the prototype, the requirements were synchronized with prototype before the developers implement the SMaRT system. Implementation of SMaRT is using the latest version of HyperText Markup Language i.e. the HTML5. Moreover, for User interface, Bootstrap was chosen as it minimizes time in designing CSS and JavaScript. In addition, Java EE was chosen as middle tier to process the information between user interface and database which is MySQL.

Testing was conducted to find any defects in the SMART system. Based on the SMART test plan, 3 types of testing were deployed that are: (1) System Test, (2) Unit Test and (3) User Acceptance Test. Initially, Unit Test was conducted by the developers using JUnit. Then, System Test was conducted by the software tester to check that all modules are working properly and list all the defects. Finally, in the User Acceptance Test, real users conducted the test and reported all defects, which appeared during the testing period. Based on the testing result, the developers fixed the defects and enhanced them accordingly.

4. Discussion and Future Work

This paper has analyzed and presented a study on existing RE tools and came out with several conclusions. The importance of RE tools in assisting requirement engineers has been highlighted in this paper. The comparison between 15 RE tools concludes the needs for an improvement in RE tool to support scenario-based methodology and cover the RE process i.e. elicitation, analysis and negotiation, documentation, validation, and management.

SMaRT supports users to manage RE process using scenario-based methodology. SMaRT reduces the time and workload consumed during the process of capturing and documenting scenarios and assist user to transfer scenarios into requirements using the scenario-to-requirements auto generation function. SMaRT also provide functions to analyze, negotiate, validate and document requirements.

The RE tools market is fast changing. Its huge demand encourages newcomers to introduce new and interesting capabilities to support RE processes. Scenario methodology might only be suitable in certain environment, which makes the competition for the most effective RE tools becomes tougher. Huge investment is required in order to deliver complex and sophisticated features.

There are some constraints and limitations of SMaRT system. Even though the difficulties of rewriting a requirement from scratch has already been solved by the generation of requirement from scenario, but the issues on how to reduce the effort taken in writing the scenario remains unsolved.

5. Conclusion

In the future, we plan to expand the RE tool functionalities to also support other variants of RE methodologies, which are highly being recognized by most companies such as Agile methodology and viewpoint approach. We also plan to widen the spectrum of the RE tools in making the comparison to strengthen the ideas being demonstrated in this paper.

Acknowledgement

This work is sponsored by Research Initiative Grant (RIGS-16-344-0508). We would also like to thank the KICT faculty and IUM for the opportunity to conduct this research.

References

- [1] L. Jiang and A. Eberlein, "A Tool For Requirements Engineering Process Development," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, 2007, vol. 2, pp. 319–325.
- [2] A. G. Sutcliffe, N. A. M. Maiden, S. Minocha, and D. Manuel, "Supporting scenario-based requirements engineering," *IEEE Trans. Softw. Eng.*, vol. 24, no. 12, pp. 1072–1088, Dec. 1998.
- [3] "Use Cases and Requirements Management." [Online]. Available: <http://casecomplete.com/>. [Accessed: 21-Aug-2017].
- [4] "YAKINDU Traceability – itemis' professional traceability tool." [Online]. Available: <https://www.itemis.com/en/yakindu/traceability/>. [Accessed: 21-Aug-2017].
- [5] "Requirements Management, Requirements Definition & Collaboration | Hewlett Packard Enterprise." [Online]. Available: <https://saas.hpe.com/en-us/software/requirements-management>. [Accessed: 21-Aug-2017].
- [6] "Requirements Management Tool RaQuest." [Online]. Available: <http://www.raquest.com/>. [Accessed: 21-Aug-2017].
- [7] "RequirementONE." [Online]. Available: <http://www.requirementone.com/>. [Accessed: 21-Aug-2017].
- [8] "Application Lifecycle Management tools | ALM Tools." [Online]. Available: <http://www.rommanasoftware.com/>. [Accessed: 21-Aug-2017].
- [9] "Agile Requirements Management Software | codeBeamer ALM." [Online]. Available: <https://intland.com/requirements-management/>. [Accessed: 21-Aug-2017].
- [10] "Leap SE Web - Agile Requirements Made Easy." [Online]. Available: <http://www.leapse.com/>. [Accessed: 21-Aug-2017].
- [11] "Software for product management, project management, & help desk." [Online]. Available: <https://www.onedesk.com/>. [Accessed: 21-Aug-2017].
- [12] "ReqLine Download (ReqLine.exe)." [Online]. Available: <http://downloads.informer.com/reqline/>. [Accessed: 21-Aug-2017].
- [13] "Reqtify - Claytex." [Online]. Available: <http://www.claytex.com/products/reqtify/>. [Accessed: 21-Aug-2017].
- [14] "ProjectManagement.com - Axiom." [Online]. Available: <https://www.projectmanagement.com/tools/261864/Axiom>. [Accessed: 21-Aug-2017].
- [15] "Requirements Management Software - Tool and Software for Product/Project Management." [Online]. Available: <http://www.gatherspace.com/>. [Accessed: 21-Aug-2017].
- [16] "Polarion® REQUIREMENTS™ - Requirements Management - Dedicated Systems." [Online]. Available: <https://dedicatedsystems.com.au/products/polarion-requirements-management/>. [Accessed: 21-Aug-2017].
- [17] "Requirements Management | Requirements Software | Jama Software." [Online]. Available: <http://www.jamasoftware.com/jama-%20requirements-management/>. [Accessed: 21-Aug-2017].
- [18] J. F. Coates, "Scenario planning," *Technol. Forecast. Soc. Change*, vol. 113, pp. 99–102, 2016.
- [19] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, 1 edition. Chichester; New York: Wiley, 1998.
- [20] J. Crinnion, *Evolutionary Systems Development: A Practical Guide to the Use of Prototyping Within a Structured Systems Methodology*. Perseus Publishing, 1992.
- [21] W. Eckerson, "Three tier client/server architecture: Achieving scalability, performance and efficiency in client server applications," *Open Inf. Syst.*, vol. 10, no. 1, 1995.
- [22] P. P.-S. Chen, "The Entity-relationship Model—Toward a Unified View of Data," *ACM Trans Database Syst*, vol. 1, no. 1, pp. 9–36, Mar. 1976.
- [23] "Online Mockup, Wireframe & UI Prototyping Tool · Moqups." [Online]. Available: <https://moqups.com/>. [Accessed: 21-Aug-2017].

APPENDIX:

Table 3.
Detailed Comparison Between 15 RE Tools

Tool	Analysis			Modelling		Verification and Validation				Management					
	Requirements Discussion	Spell Checking	Ambiguity Check	Activity Diagram	Entity Relationship Diagram	Online Review	Test Case	Test Scenario	Execute Test	C, E, D, V Requirement	Requirement Details	Versioning	Change Management	Change Notification	Change History
Quality Centre		/					/	/	/	/	/	/	/		/
Raquest		/					/			/	/	/			/
Requirement One		/					/			/	/	/			/
Rommana		/				/	/	/	/	/	/	/	/		/
Code Beamer		/					/	/	/	/	/	/			
Leap SE		/			/					/	/	/			/
OneDesk	/	/				/				/	/				/
ReqLine		/	/							/					
Reqtify		/								/	/	/	/		
Case Complete		/		/		/	/			/	/		/		/
iConcur Axiom	/	/	/				/			/	/			/	
Gatherspace		/					/			/	/		/		/
Polarion		/				/	/		/	/	/				/
Yakindu															
Jama		/				/	/			/	/				/

Table 3:
Detailed Comparison Between 15 RE Tools-cont.

Tool	Traceability			Collaboration		Documentation							Other Tool Capabilities			
	Traceability Matrix	Impact Analysis	Risk Analysis	Project Sharing	Manage Access	Excel	Word	PDF	XML	HTML	CSV	Custom	Search and Filter	Project Summary	Import and Export Project	Glossary and Term Definition
Quality Centre	/	/	/	/	/							/	/		/	
Raquest	/		/		/	/	/			/	/			/	/	
Requirement One	/					/	/						/		/	
Rommana	/	/	/	/		/	/					/			/	
Code Beamer	/						/					/			/	
Leap SE	/	/										/			/	
OneDesk	/			/	/	/		/	/	/	/		/	/	/	/
ReqLine			/	/	/											
Reqtify	/	/	/				/	/	/			/		/	/	
Case Complete	/			/	/	/	/			/			/		/	/
iConcur Axiom	/			/	/							/	/	/	/	/
Gatherspace	/			/								/				/
Polarion	/		/									/	/		/	
Yakindu																
Jama	/	/		/								/	/		/	