

Evaluation of the Software Requirement Tools

Yogita Sharma

Research Scholar

Department of Computer Science

Himachal Pradesh University

Shimla, India

Associate Prof. Aman Kumar Sharma

Department of Computer Science

Himachal Pradesh University

Shimla, India

Abstract— *The foundation of the entire software development life cycle is laid on the requirement. After receiving a request for developing a new software project, understanding the requirements and documenting them properly is the first thing that is done. If the requirements are managed properly then the right solution can be delivered on time and within budget. Because of the limitations of the document based approach used earlier, automated tools are being used. Today, there are so many software requirements tools available that can be used to speed up the process of documenting requirements, communicating them and managing changes to them. This paper aims to provide an evaluation of five such software requirements tools. The evaluation is based on the theoretical analysis and various features provided by them.*

Keywords— Requirement; Requirement Engineering; Software Requirement Tools; Software Requirements Specification; Requirement Management; Traceability.

I. INTRODUCTION

Requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose. No other part of the conceptual work is as difficult as establishing the detailed technical requirements. The success of a project whether measured in functional or financial terms can be directly related to the quality of a requirement. Therefore, Requirements Engineering (RE) plays an important role in the development of the project.

RE is a process of discovering the needs of stakeholders and documenting them for analysis, communication and implementation [14]. Many errors can be detected in the requirements phase. Davis [1] claims that fixing of errors detected in later stages of software development is more expensive than the ones found in the initial stages. If errors are not detected in the requirements phase it leads to warning product development. Wrong requirements can also lead to wastage of valuable resources.

RE can split into two main areas of activities, requirements development and requirements management. Requirements development concerns activities related to elicitation, analysis, documentation and validation of requirements. It deals principally with the content of the requirements. The purpose of requirements development is to produce and analyze customer, product, and product-component requirements. On the other hand, requirements management is the process of understanding and controlling changes to system requirements. It tracks the status of the

requirements and keeps project plans current with the requirements.

RE can face many problems. Some of the reasons can be insufficient user involvement which can delay completion of project. The situation is made worse if modifications to the requirements are made frequently. Developer's time is wasted when he has to fill in the gaps because of the ambiguous and insufficiently detailed requirements.

So, to enable effective management of an integrated RE process, automated tool support is essential. Different software requirements tools provide the capabilities for documenting requirements, managing their change and integrating them in different ways depending on project needs.

The present work gives an evaluation of the requirements tools. It consists of VI sections. Section I being the introduction, section II gives an overview of the RE process. Different tools are described in section III followed by their evaluation in section IV. Analysis and the analysis table are discussed in section V. Finally, section VI presents the conclusion and future scope.

II. REQUIREMENTS ENGINEERING

Requirements Engineering produces one large document, written in a natural language, containing a description of what the system will do without describing how it will do. The quality of a software product is only as good as the process that creates it. RE is the disciplined application of proven principles, methods, tools, and notation to describe a proposed system's intended behavior and its associated constraints.

The process of RE consists of five subcategories. These are: [15]

- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Requirements Management

A. Requirements Elicitation

Requirements elicitation is perhaps the most difficult, critical, error-prone and communication intensive aspect of

software development. Elicitation can succeed only through an effective customer-developer partnership [4]. So, to find out what users really need is the most important goal of RE. Methods that are used for requirement elicitation are interviews, brainstorming sessions and use case approach.

B. Requirements Analysis

This activity is carried out to make the requirements consistent and unambiguous by analyzing, refining and scrutinizing the gathered requirements [5]. The steps involved are drawing context diagram, developing prototypes (if needed), modeling requirements and finalizing the requirements.

C. Requirements Specification

It is a way of representing requirements in a constant format. SRS is a specification of requirements of software product. Depending on the size and complexity of the product, the SRS document may consist of a few pages or may be packed in several volumes [12]. Characteristics of a good SRS document are that it should be complete, consistent, accurate, testable, verifiable, valid, traceable and unambiguous [8].

D. Requirements Validation

The objective of this activity is to certify that the SRS document is an acceptable document of the system to be implemented. It ensures that the requirement exhibits the desirable characteristics of excellent requirement statement and of the excellent requirement specification. It finds errors that exist in the document and improve the process of software development. It is used to improve accuracy and completeness of the software requirement. Audit, walkthrough and prototyping are techniques used for requirement validation process [6].

E. Requirements Management

It concerns activities related to controlling and tracking of changes to agreed requirements, relationships between requirements and dependencies between the requirements specification and other project artifacts. Requirements management can be seen as a supportive process, which helps to manage requirements through the product lifecycle. It is the planning and controlling of the requirements elicitation, specification, analysis and verification activities [3].

III. SOFTWARE REQUIREMENT TOOLS

The document-based approach used earlier to store the requirements has many limitations. It is difficult to keep the document synchronized and current when a change is made to it and when the participants are geographically separated. It is also hard to track the status of the requirements and there is no convenient place to store the requirements that were rejected or deleted from the baseline.

The software requirements tools provide a mechanism to solve this problem. These tools provide strong capabilities for defining attributes for each requirement type. Learning aids such as tutorials or sample projects are also provided to help users to use them conveniently.

Some of the requirement tools considered in the research are Rationale RequisitePro, Objectiver, CaseComplete, RMTrak and Optimal Trace. These tools are complimentary to each other in the manner in which they manage the requirements and generate the requirements documents.

A. Rational RequisitePro

RequisitePro is a requirements management tool developed by IBM. Microsoft Word is the primary interface used for managing requirements. It gives software development project teams the ability to manage all project requirements comprehensively and facilitating team collaboration and communication. It is based on the use case model. Rational RequisitePro increases the quality, reduces the risks and lead to right solutions on time and within budget [7]. It improves traceability and provides requirements tracking and management capability. Oracle, Microsoft SQL Server and MS Access are the three databases supported by it. RequisitePro integrates with the industry leading tools to optimize the flow of requirements data throughout the project.

One of the powerful feature of Rational RequisitePro is the change tracking. It provides the industry standard templates. It also helps in managing the multiple versions of the projects by archiving the projects. [11]

B. Objectiver

Objectiver is a RE tools to identify all the project requirements using a structured modeling approach. It has been developed by Cediti and designed by RE practitioners. The tool relies on Kaos that is a goal driven methodology. Because of the systematic identification of the links between the goals and objectives being pursued, the roles and responsibility and the process and their priorities, it gives a global and detailed view of the system or the project. A consistent and complete glossary of all problems related items is also provided by this tool.

Automatic document generation according to the standards is one of the main features of the tool. It also ensures traceability. It is packaged with the web generation tool that permits to share the processes and the goals with the employees. Graphical editor is provided to represent the concepts and their relationships. [9]

C. CaseComplete

CaseComplete is a software development tool developed by Serlio Software. It provides the ability to edit the textual portion of use cases and requirements in a guided environment and provides the ability to create various types of diagrams including use case diagrams. CaseComplete comes with built-in reports. The reports are generated in MS Word and UML (Unified Modeling Language) models directly from use case models. The major strength of the tool is its ease of use. Within minutes one can install it, write a use case and generate a clear requirements document. It guides its users through each step and collaborates with team to produce a complete and accurate set of requirements. As the use case serve as an excellent foundation for functional testing, CaseComplete takes this one step further by allowing entering test procedures, suggested inputs and expected output results for each step and extension. It also allows tracking any changes, additions and deletions to the project. It

provides an index of glossary items to help understand the various technologies used in the system. [2]

D. RMTrak

both the naive and the advanced users. The tool is designed with two primary features. First, it allows its users to create their documents as they usually do while adjusting to their needs. Secondly, it keeps the important information in the document themselves. RMTrak allows managing the development projects requirements.

Different kinds of reports and views depending on the needs can also be generated using the tool. To mark the most important information within a document, tags are included. It eliminates problems of database becoming corrupt or about losing RMTrak database files. It provides the ability to monitor any changes made to the requirements and also works with the trusted tools. It keeps track of the requirements either approved or disapproved the required tests. It has a tag filter that allows more than one project to share documents among themselves. [13]

D. Optimal Trace

Optimal trace is a collaborative team oriented requirements capture and management tool intended for the structured capture of requirements, thereby allowing the scope of a system to be established. It has rich GUI. It allows integrating with the external design tools. It also provides document profiles facility containing the requirements document templates and allows one to create profile and templates too. Multiple projects can be loaded at any one time. Traceability and project version history are among the important features provided by the tool.

It provides a mechanism for capturing both functional and non-functional requirements for a system using the structured and simple requirements. It generates the requirements documents in MS Word. Users can view the errors and trace their location to the corresponding table. Optimal Trace Professional provides separate plug-in integration into the leading UML design tools, Compuware's Optimal J, IBM/Rational Rose, and Borland. Optimal Trace's automatic document generation make communication easy. Project baselines can be created and be notified of changes to specific project element. [10]

IV. EVALUATION OF THE TOOLS

A comparative evaluation of various tools has been performed. The evaluation is based on various features supported by the tools that are considered important for the management of the requirements. Some of the features are described briefly below.

A. Documentation Support

A tool must produce a well structured and easily understood requirements document conforming to the standards. These requirements do help in the entire Software Development Life Cycle (SDLC). Tool should also let the requirements created outside of a project document to be imported or copied into the document.

B. Traceability

RMTrak is a tagging, tracing, allocation and verification system, also known as Requirements Management tool and is developed by RBC Product Development. It can be used by each requirement and track every change made to the requirement.

It is concerned with documenting the life of a requirement and providing bidirectional traceability between various associated requirements. It enables users to find the origin of

C. Templates

It provides a standard format for describing tasks that user needs to perform. These are structured and consistent way to organize the product's functional and non-functional requirements. Tool should also provide its user the ability to create own templates easily and without any difficulty if needed.

D. Glossary

It is used to define the different terminology used in the project. It proves a great asset to the development team. It reduces misunderstanding.

E. Tool integration

A tool should support integration with other industry-leading tools and other lifecycle tools. It will optimize the flow of requirements data throughout the project.

F. Graphical Representation

It not only provides a nice picture of data, but it is also quick. Randomness in data can be easily identified. It helps in comparative analysis of data and avoids wastage of money.

G. Checklist

A checklist provides a way to verify the criteria of the requirements and facilitates to deliver quality and tested requirements. It helps to ensure consistency and completeness in carrying out a task.

V. ANALYSIS

The tools namely Rationale RequisitePro, Objectiver, CaseComplete, RMTrak and Optimal Trace were validated for all the features discussed in Section IV. The evaluation is done using their trial versions. The tools were analyzed theoretically on the literature, manuals and were further validated by execution of the tools.

Table 1 provides the information about the features provided by the tools. The right tick (✓) show that the tools have the corresponding functionality and (X) show the absence of the one. The features supported by every tool include templates, traceability, documentation support and graphical representation. Checklist is the one criterion fulfilled only by Objectiver. RMTrak, Rationale RequisitePro and Optimal Trace tools do not provide glossary and checklist feature. Similarly, CaseComplete does not support tool integration and checklist. The advantage of CaseComplete above all other tools is its ease of use. It can be installed within minutes and it guides through each step. From the table it is interpreted that among the tools analyzed

in the research, Objectiver is the only tool that support all the functionalities that have been considered for the research

while other tools lack one or two features. Hence, Objectiver is the best tool.

TABLE 1: ANALYSIS OF SOFTWARE REQUIREMENTS TOOLS

Tools/Attributes	Glossary	Templates	Traceability	Tool Integration	Documentation Support	Graphical Representation	Checklist
Rational RequisitePro	X	√	√	√	√	√	X
Objectiver	√	√	√	√	√	√	√
CaseComplete	√	√	√	X	√	√	X
RMTrak	X	√	√	√	√	√	X
Optimal Trace	X	√	√	√	√	√	X

Note: X: means feature is available.

√: means feature is not available.

VI. CONCLUSION AND FUTURE SCOPE

The study evaluated five software requirements tools namely Rationale RequisitePro, Objectiver, CaseComplete, RMTrak and Optimal Trace. Requirements engineering is an important process for the software development and use of automated tools make this process easy. Traceability, documentation support, glossary, templates, tool integration, graphical representation and checklist are the features considered important for the proper management of the requirements have been used as the evaluation criteria in this study. These features make the process of requirements management efficient and faster for the administration and the team members. The automated software requirements tools solve the problems of the

document based approach. They provide proper documentation that can be used in other phases of the SDLC and a lot of flexibility in managing changes to the requirements.

The evaluation of the tools is based on their theoretical analysis and is performed using the trial versions of the tools and their respective manuals. Only five tools have been used for the study but there are numbers of software requirements tools available that provide functionalities other than the ones mentioned here. So the research can be extended by adding more tools and more functionality for evaluation.

REFERENCES

1. A.M. Davis, Software Requirements- Objects, Functions, and States, Prentice Hall, 1993.
2. CaseComplete, [www. casecomplete.com/](http://www.casecomplete.com/) (visited on 17-12-2013 at 1200 hrs).
3. J.D. Arthur, R.R. Sud, "Requirements Management Tools A Qualitative Assessment", Department of Computer Science, Virginia Tech Blacksburg, VA 24060 USA.
4. K.E. Weigers, Software Engineering, Microsoft Press, 1993.
5. K.K. Aggarwal, Y Singh, Software Engineering, New Age Techno Press, 2008.
6. M.U. Bokhari, S.T. Siddiqui, "A Comparative Study of Software Requirements Tools For Secure Software Development", BIJIT-2010 Vol.2 No.2.
7. M.U. Bokhari, S.T. Siddiqui, "Metrics for Requirements Engineering and Automated Requirements Tools", Proceedings of the 5th National Conference; INDIACom-2011, Computing For Nation Development, March 10 – 11, 2011.
8. N.S Gill, Software Engineering, Khanna Book Publishing Co. (P) Ltd, 2002.
9. Objectiver, www.objectiver.com/(visited on 24-12-2013 at 1100 hrs).
10. Optimal Trace, [supportline.microfocus.com/Documentation/books/ Optimal Trace/Doc/CI51/PDF/Optimal-Trace-Professional-Help.pdf](http://supportline.microfocus.com/Documentation/books/Optimal%20Trace/Doc/CI51/PDF/Optimal-Trace-Professional-Help.pdf) (visited on 01-01-2013 at 1400 hrs).
11. Rationale RequisitePro, www-03.ibm.com/software/products/en/reqpro/ (visited on 08-01-2013 at 1100 hrs).
12. R Fairley, Software Engineering Concepts, Tata McGraw Hill, 1997
13. RMTrak, www.rmtrak.com/home.aspx (visited on 20-01-2013 at 1230 hrs).
14. S Easterbrook, B Nuseibeh, "Requirements Engineering: A Roadmap" International Conference Software Engineering, 2000.
15. S Thangasamy, Essentials of Software Engineering, WILEY INDIA, 2012.

IJERT