

Tools for Requirements Management in GSD: A Survey

Muhammad Mukhtar, Zishan Hussain Chuhan, Zulfiqar Ahmad

Abstract--- The software requirement specification (SRS) is a volatile document. Even well documented SRS evolves and grows throughout Software Development Life Cycle. Requirements management plays an important role to manage evolution and growth in SRS. Managing requirements in manual ways becomes very difficult especially in global software development due to some additional factors (time zone difference, cultural issues, geographical boundaries, etc.). To overcome this difficulty software industry moves to automate the requirements management. In this research activity, our focus is to survey about tools to automate the requirements management in global software development (GSD). We consider existing tools in this survey those are not developed for GSD and evaluate them on defined parameters in the context of GSD. In short our goal is to validate the existing tools for GSD.

Index Terms - Requirements Management, SRS, Tools, Automation, GSD, Integration, Access Control

1 Introduction

Requirements management (RM) is a discipline, in which change and versions of requirement is controlled, update plan according to the current requirements, managing traceability of requirements (impact analysis), and status of the requirements is tracked [1]. SRS is a volatile and dynamic document because requirements are frequently re-allocated between different builds [5]. To manage the changes in SRS, RM plays an important role. Totally manual requirements management can be proved costly, inefficient and time consuming, especially in global software development. Because in GSD there are some additional factors also includes and have huge impact on requirements management activity [6]. Additional factors in global software development that have an impact on requirements management are time zone difference, cultural issues, and geographical boundaries [10]. Difficulties in requirements management is increased in GSD because stakeholders are distributed in this type of development [8].

- *Muhammad Mukhtar* is currently working as Assistant Professor The University of Lahore, Pakistan , PH-00923215653221, Email: Muhammad.mukhtar@cs.uol.edu.pk
- *Zishan Hussain Chuhan* is currently pursuing PhD from University of Lahore ,Pakistan . PH-00923364397507, Email: zishan.hussain@cs.uol.edu.pk
- *Zulfiqar Ahmed* is currently working as Lecturer CIIT Islamabad ,Pakistan, , PH-00923345191165 ,Email: Zulfiqar@ gmail.com

We arrange the problematic factors in GSD across their causing factors as discussed in section 2.1. To overcome the difficulty to manage growth and evolution of requirements software industry moves to the automation of requirements management [1] and

develop many tools These tools proved to be very helpful to keep specifications consistent, up-to-date, ensure requirements traceability and accessible [3, 9]. In fact, the existing tools (DOORS, Analyst Pro, PACE, etc) were not developed for global software development. Existing requirements management tools require a high degree of knowledge to understand and use it [5]. We have extract criterion to evaluate existing tools in the context of GSD by considering additional problematic factors. To defining this criterion [2, 3, 9] proved very helpful. On the base of evaluation, result is concluded that which of the tool can be proved appropriate for global software development and which are not. Also some suggestions will be provided to enhance the existing tools to make them helpful in the context of global software development. Remaining paper will be divided as, Section 2 describes requirements management, section 3 describes requirements management tools, section 4 present syntheses for tools and section 5 concludes the paper and provide the intention on future work.

2. Requirements Management:

Requirements management is the process of plan, execute, monitor and control the requirements. RM plays an important role in engineering discipline, especially in distributed projects. The main activities of RM are manage versions and changes, store requirements attributes link requirement to other system elements, track status, view requirements subsets, control access, and communicate with stakeholders [1]. Monitoring and controlling the requirements is very crucial activity in software engineering. Main goal of the requirements management is to meet the stakeholders' conditions and requirements, so that final result will be according to desire [9]. Uncontrolled requirements lead to many software

project failures. Shortly if software engineer will not control and monitor the requirements then project may be move to failure [1].

2.1 Requirements Management in GSD:

Management of requirements becomes more problematic in global software development due to some additional factors. Also importance of requirements management increased in GSD environment due to distribution of project tasks on global locations. By the help of literature about challenges and issues of GSD, we arrange the problems of GSD across their causes as follow.

2.1.1 Problems in GSD:

By [2, 3, 4, 6, 8, 9, 10, 11] we have come to know that following can be the major problems raised during software development but in GSD their impact increased due to additional causes.

- Communication
- Coordination
- Cooperation
- Control
- Knowledge management

2.1.2 Causes for problems in GSD:

Problems described in section 2.1.1 can be raised due to following causes [8, 9, 10, 11].

- Cultural Issues
- Technical Issues
- Geographical Issues
- Time-Zone Difference
- Lack of Trust
- Language

2.1.3 Problem vs. Causes:

In this section we describe that which causing factors have an impact on particular problem.

2.1.3.1 Communication:

During software development in global environment communication can be problematic due to language, lack of trust, time-zone difference, geographical distance, technical issues, and cultural diversity.

2.3.1.2 Coordination:

All causing problems participate to raise the coordination problem in GSD.

2.3.1.3 Cooperation:

Cooperation is lacking due to language, geographical distance, and lack of trust.

2.3.1.4 Control:

Controlling problem is raised in global software development due to geographical distance, cultural diversity, time zones, and language.

2.3.1.5 Knowledge Management:

Planning and sharing of knowledge can become a problem due to language, time zones, cultural diversity, lack of trust, and geographical distance.

3. Requirements Management Tools:

"A tool is not a process in itself, but it supports and enables an established process."[1]

Managing requirements in manual way will be problematic especially in collaborative software development. To overcome this difficulty software industry moves for automation. RM tools provide help and ease the software development process, especially in collaborative software development where development teams are working at different places [9].

In this section existing requirements management tools will be discussed one by one. We did not include those tools that are already evaluated by [1, 9].

3.1 EGRET:

V. Sinha et al. [7] develop requirements management tool; Eclipse based Global Requirements Tool (EGRET). It provides support for managing requirements in collaborative development. Main views of EGRET are requirements explorer, communication record, traceability, project stakeholders, project phases and eclipse navigator view. EGRET is developed for managing requirements in collaboration. It accepts all the challenges rose in global software development and provide a better solution. It provides synchronous and asynchronous communication to facilitate the informal conversation. Jazz tool is used to support synchronous communication. It also promotes the awareness among the stakeholders. Manage changes through formal collaboration. Change management is done in three steps, submitting change request; processing change request and acting on change notification. It also provides the facility of knowledge management to help users navigate the data. Overview of EGRET is given in Fig 1.

"The EGRET tool can support global software development teams in collaborating on requirements management". [8]

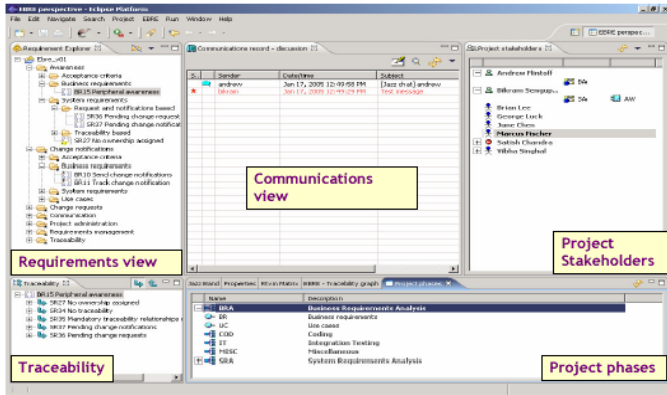


Fig 1. Overview of EGRET [7]

3.2 PARSNIP:

M. A. Babar and D. Zowghi [4] develop requirements management PARSEd Natural language Input Processor by integrating various tools. It consists of a number of independent or standalone tools. A high level view of PARSNIP has been shown in Fig 2. CARET (Computer-Assisted Requirements Evolution Toolset) used to check the inconsistencies about requirements. Tool X supports specification and procurement process on requirements. It consists of single repository to store requirements. CARL (Computer-Assisted Reasoning in natural Language) was a prototype to apply the CARET framework.

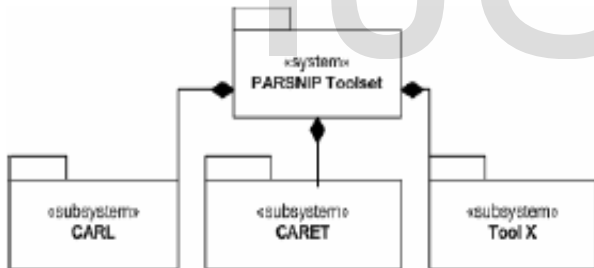


Fig 2. High-Level view of PARSNIP [4].

3.3 RM-Tool:

M. Lang and J. Duggan present prototype design for collaborative requirements management named RM-Tool [2]. In [2], at first author defines assessment of requirements management tool, then mentioning the weak-nesses of current requirements management and finally present the prototype design to covering the mentioned weaknesses. Main focus of this tool is to satisfy those problematic factors that are ill satisfied by current tools instead of all coverage. Main features of RM-Tool are; maintain shared data dictionary and requirements description, support for

standard modeling notations, store and manipulate rich text, supports time and location independent communication, backwards and forwards tracking, project management, reporting capabilities and generate hard copy. RM-tool overcome the problems faced in global software development. Like it facilitates the improved communication and understanding by providing a balance between technical and non-technical specification techniques. Also facilitate the enhanced collaboration between developers and end-users as well as control the impact of changes to requirements. High level use case diagram for RM-Tool has been shown in Fig 3.

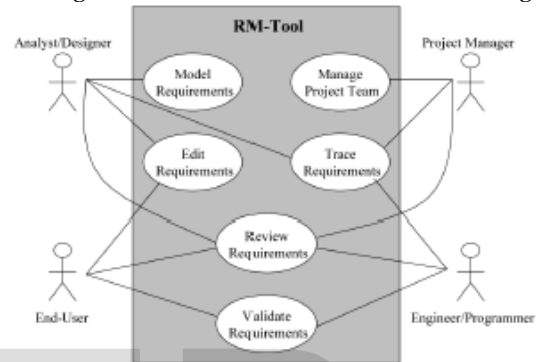


Fig 3. High-Level use case diagram for RM Tool [2].

3.4 Analyst Pro:

Goda Software presents solution for requirements management in the shape of Analyst Pro [12]. It facilitates requirements specification, tracking and visual traceability analysis. It is a scalable solution, which can provide a collaborative environment that allows sharing of common pool of project information among stakeholders. The requirements can be tracked through design and testing. It classifies requirements; generate specification documents in automated way, and importing and exporting requirements facility. Also provide facility of automated change and knowledge management. Facilitate to reusability as well as traceability of requirements. A simple view has been given in Fig 4.

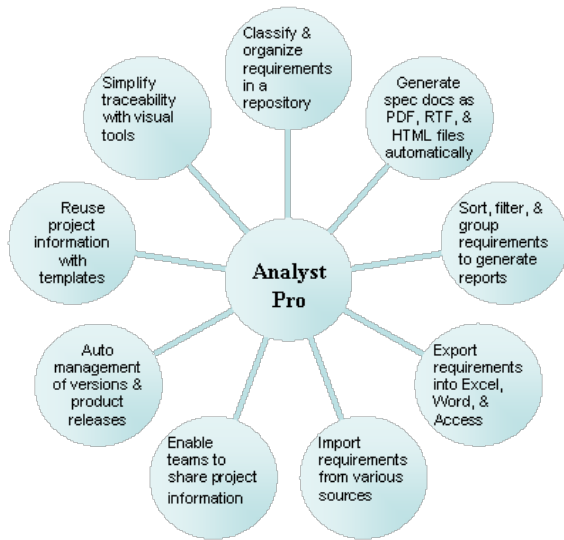


Fig 4. Overview of Analyst Pro [12]

3.5 CORE 5.1:

Vitech Corporation develops a tool for requirements management CORE 5.1 [13]. Main features of CORE 5.1 are; reduce schedule risk, improve communication, enable collaboration, define and verify requirements. Also it ensures completeness and consistency as well as provides facility to plan the test at early stage. It also ensures up-to-date documentation and improves planning, visibility and control. Shortly we can say that it can be helpful in global software development by overcoming the issues in GSD. It has an additional factor of validation and verification. In other words we can say that provides a valuable communication between customer and developer.

3.6 CONTOUR:

Jama Software presents solutions for collaborative requirements management in the form of CONTOUR [14]. It has four main components capture, control, communicate and collaborate. Capture: it is easy to start capturing requirements immediately; each team can configure CONTOUR to meet their individual needs. It provides a rich editor that enables users to capture requirements directly to the central repository. Control: it provides control over change within projects. It elicits any change to individual requirements and generates a version history for corresponding documents. Manage the baseline according to change as well as perform the impact analysis over change. Communicate: CONTOUR ensures an effective communication between the stakeholders. All stakeholders, regardless of location gain immediate access to the latest information and receive notification

when requirements go to change. Collaborate: Contour is designed to provide the entire team with access to requirements and other artifacts. Each requirement and artifact in Contour has a discussion thread where the team can add comments and collaborate on individual items.

3.7 Projectricity:

Projectricity develops a requirements management tool named Projectricity [15]. It is a web-based project management platform that enables project team members to efficiently communicate and work collaboratively no matter where they are located. It manages the requirements at all levels of project. It has the ability to manage project and task information. Also it is able to manage requirements, test plans, change requests, traceability, and problems in requirements, risks and documentation. Track the status of requirements to keep the project with current requirements. Provide bi-directional linking between requirements to track the status and impact analysis in the context of traceability. Keep assets to reuse them as well as provide sequential workflow for the development.

3.8 Feature Plan :

Feature Plan introduces a requirements management tool named as Feature Plan [16]. It has the capability to handle many problems like unstructured processes, distributed and no linking requirements. It has the ability to improve the management process, centralize the data, define roadmaps for stakeholders, and automate documentation. An overview has been given in Fig 5.



Fig 5. Overview of FeaturePlan 2.6 [16].

3.9 Cradle 5.2:

3SL Inc. introduces Cradle 5.2 for software development as a tool [17]. It can be used for requirements management purposes. It has ten modules, five components. It has capability to capture and manage the requirements. It has a module REQ that provide the complete solution for requirements management.

4 Synthesis:

In this section criterion is defined for the evaluation of above discussed tools by the help of [2, 3, 9, 18] and section 2.1. We define criterion on the base of major problems. In this section we provide evidence for criterion. Also provide reasons for selecting a particular criterion parameter. Different parameters are enlisted below with definition and reason of selection.

4.1 Way of Communication:

Communication allows people to exchange thoughts by one of several methods. There are auditory means, such as speaking, and physical means, such as sign language, face to face. Communication is most problematic in global software development. A tool should facilitate communication in all ways like written, face-to-face, etc. We make a communication tree in Fig. 6 that elaborates all possible way of communications. Fig. 6 consists of some blocks which are considered for clarity in figure. Block A consist of different way of communications. There is a node A & B in tree which represent that here blocks are replaced in fact. Way of communication parameter is selected to solve the communication, cooperation, coordination, and control. If there is no way of communication then there will be no-coordination, no control, and no cooperation exist between different locations.

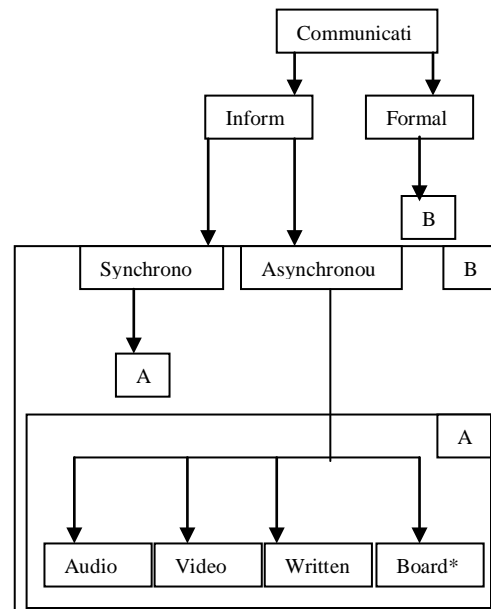


Fig. 6 Communication Tree

4.2 Multi-Language:

Multi-Language means either tool support multiple languages. In other words we can say that is there any translator in tool that translates from one language to the other one. We select multi-language as a criterion to overcome the linguistic problem. With out multi-language translator communication, coordination and cooperation cannot be effective. In GSD its importance is increased.

4.3 Standard Notations:

Standard notations means either tool support standard system modeling and specification techniques, which proved to be helpful for all stakeholders located at different locations. It is selected as a parameter to overcome ambiguity in communication to cover knowledge management and cultural diversity.

4.4 Notification and Suggestions:

Notification and suggestions are selected as parameter to awareness at long distance to cover the geographical boundaries. It is selected as a parameter to facilitate effective communication and provide help in knowledge management in global software development.

4.5 Format:

Format means how tool is storing information. This parameter is further divided into two sub-parameters (natural language and formal notations). Format of storing requirements is selected for effective communication and better knowledge management.

4.6 Integration:

Integration means the ability of inter-tool communication. A tool should be able to work with other tools. Integration is selected as a parameter to cover the geographical boundaries. It provides help to facilitate effective communication between different parties in global software development.

4.7 Web Access:

Web Access means either tool has a web interface that makes it unnecessary to install a client application for occasional users.

It is selected as a parameter to facilitate communication and coordination with different stakeholders for occasional user in global software development.

4.8 Simultaneous Use:

Simultaneous use means tool allows interacting with multiple users simultaneously. It is selected as parameter to support synchronous communication and effective coordination.

4.9 Roles of Stakeholders:

Roles of Stakeholders means either tool specify and identify the particular role across particular stakeholder. Role identification is selected as a parameter to overcome the problem of control.

4.10 Access Control:

Access control means either tool provides access control to each participant on appropriate data. Accessing control is selected as criterion parameter to cover lack of trust issue.

4.11 Track Status:

Track status means keeping the current status of each requirements to keep the project up-to-date. It is selected as a criterion parameter to improve the trust and make up-to-date the stakeholders for effective communication.

Criteria		Tools →									
		EGRET	PARSNIP	RM-Tool*	Analyst Pro	CORE 5.1	CONTOUR	Projectrity	FeaturePlan 2.6		
Format	Natural Language	Yes	Partial	Yes	Yes	Yes	No	Yes	No		
	Formal Way	Yes	?	Yes							
Track Status		Yes	Partial	?	Yes	Yes	Yes	Yes	Yes		
Usability, Simplicity		Yes	No	Yes	Yes	Partial	No	?	?		
Multi-platform		Yes	No	Yes	Yes	Yes	Yes	No	No		
Integration		Yes	Yes	Yes	Yes	Yes	Partial	No	No		
Web Access		Yes	No	Yes	?	No	Yes	No	No		
Access Control		Yes	No	Yes	Yes	Yes	Yes	No	Yes		
Information Sharing		Yes	No	Yes	Yes	Partial	Yes	Yes	No		
Simultaneous Use		Yes	No	Yes	Yes	No	Yes	No	No		
Roles of Stakeholders		Yes	No	Yes	Yes	Yes	Yes	Yes	Yes		
Standard notations		Yes	No	Yes	Partial	Yes	Yes	Yes	No		
Flexibility		Yes	No	No	Yes	Yes	No	No	Yes		
Multi-Language		No	No	No	No	No	No	No	No		
Notification/Suggestions		Yes	No	No	Yes	Yes	Yes	Yes	Yes		
Way of Communication	Synchronous	Formal	Written	Yes	No	No	?	Yes	Yes	No	Yes
			Video	No	No	No	No	No	No	No	No
			Board	No	No	No	No	No	No	No	No
			Voice	No	No	No	No	Partial	No	No	No
	Informal	Written	?	No	No	?	Yes	Yes	No	Yes	
		Video	No	No	No	No	No	No	No	No	
		Board	No	No	No	No	No	No	No	No	
		Voice	No	No	No	No	Partial	No	No	No	
	Asy	Forr	Written	Yes	No	No	?	Yes	Yes	No	Yes
			Video	No	No	No	No	No	No	No	No

	Informal	Board	No	No	No	No	No	No	No	No
		Voice	No	No	No	No	Partial	No	No	No
		Written	?	No	No	?	Yes	Yes	No	Yes
		Video	No	No	No	No	No	No	No	No
		Board	No	No	No	No	No	No	No	No
		Voice	No	No	No	No	Partial	No	No	No

Table 1. Synthesis Matrix.

4.12 Flexibility:

Flexibility means tool should flexible to support different types of documented SRS. This parameter is selected to support geographical issues in global software development.

4.13 Information Sharing:

Information sharing means tool should provide facility to manage the shared information for stakeholders. It is selected as a criterion parameter to support knowledge management as well as improve coordination and cooperation.

4.14 Usability, Simplicity:

It means either tool has facility that a non-technical user can easily interact with it for posing queries.

4.15 Multi-Platform:

Multi-Platform means either tool can be operated on multiple platforms or it is specific to the particular platform. It is selected as a parameter to evaluate that either tool can capture geographical boundaries (distance). Synthesis Matrix is given in Table 1.

Values Used in Table 1.

Yes: means particular tool meets the corresponding criteria.

No: means particular tool did not support corresponding feature.

Partial means tool meets criterion partially.

?: means no clarity about particular feature across the tool

5 Conclusions and Future Work

In this survey paper we have included some tools to evaluating in GSD context. From table 1 synthesis matrix, no tool completely confirms the defined criterion. RM-Tool confirms some but not all requirements about GSD, but complete general requirements are not mentioned in table because it only discusses the GSD requirements for requirements management tool. Analyst Pro can also be used for collaborative development with some enhancement.

In the future we will suggest valuable suggestions to meet the GSD criterion.

6 REFERENCES

- [1] K.E. Weigers. "Automating Requirements Management". Software Development, July 1999.
- [2] M. Lag and J Duggan, "A tool to support Collaborative Software Requirements Management". Requirements Engineering vol 6 No. 3. London limited 2001.
- [3] M. Hoffmann, N. Kuh, M. Wber, and M. Bittner, "Requirements for Requirements Management tools", Proceedings of the 12th International Requirements Engineering Conference 2004.
- [4] M.A. Babar and D. Zowghi, "Developing a Requirements Management Toolset: Lesson Learned", Proceedings of the 2004 Australian Software Engineering Conference.
- [5] T. Hammer and L. Huffman, "Automated Requirements Management- Beware HOW You Use Tools An Experience Report," *Proceeding of 3rd International Conference on Requirements Engineering*, 1998.
- [6] M. Lag and J Duggan, "Toward Enhanced Automated Support for Collaborative Software Requirements Management", *International conference on information systems development*, 2001.
- [7] V. Sinha, B. Sengupta and S. Chandra "EGRET: A Collaborative Tool for distributed requirements management", TR, report RI06001, IBM Research 2005.
- [8] V. Sinha, B. Sengupta and S. Chandra. "Enabling Collaboration in Distributed Requirements Management". *IEEE Software*, Vol. 23, Issue 5. Oct. 2006. p. 52-61.
- [9] S. Heinonen, "Requirements Management Tool Support for Software Engineering in Collaboration". University of Oulu, Department of Electrical and Information Engineering. Master's Thesis, virtual.Vtt.fi. 2006.
- [10] D. Damian and D. Zowghi, "Requirements Engineering challenges in multi-site software development organizations", *Requirements Engineering Journal* 2003.
- [11] D. Damian. "Requirements engineering in distributed projects". *IEEE International Conference on Global Software Engineering (ICGSE'06)*, 2006.
- [12] <http://www.analysttool.com> (28-5-2014)
- [13] <http://www.vitechcorp.com/solutions> (26-03-2014)
- [14] <http://www.jamasoftware.com> (23-05-2014)
- [15] <http://www.projectricity.com/> (14-05-2014)
- [16] <http://www.featureplan.com/> (09-06-2014)
- [17] <http://www.threesl.com/> (19-05-2014)
- [18] <http://www.paper-review.com/tools/rms/read.php> (25-05-2014)