# Software Requirements Management

Ali Altalbe
Deanship of e-Learning and Distance Education
King Abdulaziz University
Jeddah, Saudi Arabia

*Abstract*—**Requirements are defined as the desired set of characteristics of a product or a service. In the world of software development, it is estimated that more than half of the failures are attributed towards poor requirements management. This means that although the software functions correctly, it is not what the client requested. Modern software requirements management methodologies are available to reduce the occurrence of such incidents. This paper performs a review on the available literature in the area while tabulating possible methods of managing requirements. It also highlights the benefits of following a proper guideline for the requirements management task. With the introduction of specific software tools for the requirements management task, better software products are now been developed with lesser resources.**

***Keywords*—Software; Software Requirements; Software Development; Software Management**

## I. Introduction

The term 'Requirement' is used to describe the set of desired characteristics and attributes possessed by a particular product or a service. These requirements may take the form of a formal statement of a function that needs to be performed, or it may take the form of an attribute that needs to be an integral part of the said entity. It is obvious that understanding the customer requirement is the number one priority in the designing phase of a system or an application. It is estimated that about 50% - 60% of software malfunctions are caused as a consequence of bad software management. This means that the problem is not associated with programming and the development team is by no means at blame, rather, the problem is purely due to the final product not meeting the customer's requirement. Inadvertently, it is possible for prerequisites to bear blunders due to vague or inadequate descriptions, however, by following modern software requirements management techniques, it is possible to successfully face these pitfalls [1].

Analyzing requirements is inherently a time consuming task which involves a considerable amount of observation. it involves the determination of requirements for an innovative or a changed organization while being mindful about probable incompatibilities. Basically, this process can be segmented as the congregation, incarceration and the specification of the particular project requirements specification.

The rest of this paper is divided into several segments. The literature review presents an overview of the main principle based on current literature available. The next chapter discusses the practices in the field of requirements management. The fourth chapter introduces the requirements traceability matrix, which is essentially one of the best methods available for the task at hand. It also discusses the advantages posed by the use of various tools for the task. The final chapter concludes the paper while briefly noting the potential work for the future.

## II. Literature Review

### A. Overview of Software Requirements Management Principles and Practices

As mentioned before, requirements management is an integral part of the standard project management life cycle. The main purpose of requirement management is to maintain a good relationship between the client and the developer of the project. The main objective of requirement management is therefore to guarantee that the organization, documentation and the verification of a project meets the requirements of the client. Furthermore, the involvement of this process within the life cycle will further enhance the software development by incorporating various management principles that aids to capture, evaluate, articulate, persuade, correspond, supervise and handle the needs for the required outfitted competence. These requirements can be broadly categorized as technical and non-technical [2], [3].

### B. Key Principles

*1) Early engagement of stakeholders in process:* Management techniques must be capable of identifying stakeholders while guaranteeing a wide focus during the requirement analysis

*2) Requirements analysis in the working context:* Management techniques must foresee and analyze the initial requirements into product and process requirements or functional and non-functional requirements and carefully plan the situations at which each service must be delivered

*3) Discriminating evidently between prepared solution and requirement:* In order to facilitate any disparity that needs to be determined and supervised, the operational requirement of the particular need has to be well defined. Both descriptions necessitate their own varying classifications, drivers and metrics that develop at different speeds

*4) Distinguishing between the roles of customer, supplier and user:* These different roles necessitate different mind-sets, ethics, proficiencies, responsibilities, intentions and working measures. The distinguishment between these roles therefore enables expertise to be enhanced and the provision of a strong review of all related activities

*5) Early identification and addressing of the interoperability :* Few services can be installed secludedly. Due to the this nature, most of them are required to amalgamate, interface and operate with one another. Management techniques need to recognize and tackle with such interoperability issues from the beginning. This ensures the seamless integration of products that will eventually provide an elucidated system of systems

*6) Review achievability:* Requirements management should sustain the constant estimation of the planned solution in order to make sure that it is attainable and the predefined criteria are met with

*7) Trade-off between presentation, capital investment and time:* To reduce the acquisition risk, either presentation, capital or time has to be negotiated, which in turn might have an impact on the return on investment

*8) Aid in managing jeopardy and ambiguity:* By implementing a cyclic approach to the development process, it aids in providing lucidity and can tremendously reduce ambiguities which may prevail in final products. Through the use of requirements engineering, if any ambiguities do arise, necessary provisions can be made to focus the attention of stakeholders on such matters immediately. These ambiguities are often the result of impractical user requirements and the handling of such is known as risk management

*9) Familiarizing or scale to suit:* Requirements management should be pliable to all attainments of all final goods and services regardless of the intricacy

*10)Help in identifying, analyzing and final s election of choices:* In practical scenarios, it is often the case that the primarily identified 'best' approach later turns out to be meagre in comparison with others. By doing a thorough requirements analysis, it is possible to choose an approach which provides the finest returns for capital, time and investment

## III. Practices in Requirements Management

The administration of software requirements is tricky even under the finest of conditions. Therefore, it is often necessary to reduce the complexity and improve the possibilities of achievement prior to the application of such methods. Top practices followed in the industry can be categorized as follows [4].

- Project Planning
- Work Estimation
- Progress Tracking
- Learning for the future

### *A. Defining Requirements for Requirements Management Practice*

Three very important considerations have to be kept in mind to successfully implement such a practice. These three factors are people, process and technology. The goal is to effectively achieve efficiency by determining which tool or technique should be used for each occasion [5].

*1) From a Process perspective:* Resource Management strategies assign tasks for both the resource management process itself as well as for the closely related change management process

*2) From a People perspective:* The identification and definition of system development roles as a source of requirement falls into this category. These categories might deliver roles such as designing, coding or testing and the allocated person is responsible for eliciting and analyzing the requirement.

*3) From a Technology perspective:* Identification of the tool of choice is required through this category. The tool can be diversified, ranging from simple pen and paper to software packages such as Microsoft Word or Excel to even sophisticated management repository packages.

### *B. Managing Change Requests*

Change management is the task during which the changes in the requirements are executed in a well managed mode presented through a pre-described structure with sensible changes. Software Configuration Management (SCM) or the change control process is considered as the perfect solution for tackling changes in the software development life cycle. In its essence, SCM is a job involving the tracing and the management of modifications of the software itself. Due to the vast benefits this process can deliver, it is considered to be vital in the industry. The change management process defines the requirement for change tracing and teh capability to validate that the final deliverable software has all of the desired augmentations that are required to be contained within the next release. The methods listed below are described for each and every software project in order to ensure that a proper change management procedure is employed.

- Identification of change in requirement is the procedure of recognizing the functionality that describes everything of a configation item. A configuration element is an item for consumption that has a consuming user function
- Change status explanation is the facility to record and account for on the configuration items at any given time instance
- Change control is a collection of processes and agreed upon stages that are necessary to change a configuration item's features and reestablishing the baseline
- Configuration assessments are busted into physical and functional configuration assessments. They arise either at the time of effecting a modification to the project or at the time of delivery

### *C. Potential Issues of Managing Change*

The changes made in software usually results in affecting many users as well the stored data. If these changes are not well managed, the end result will be disastrous [6], [7].

*1) Data Storage:* Database changes are introduced that will require application modifications. Lack of a defined change in the management process often makes it difficult for the development team to troubleshoot or resolve an issue clearly

*2) Data Movement:* If not managed properly, this may lead to a issues of compromised information continuity. The requested change may often require changing a data feed into a data warehouse. If saving is not properly carried out, data recovery is extremely difficult

*3) Security:* The new changes made to the code might induce security breaches. If the changes are not properly managed and untested, it can lead to potential data catastrophes

*4) Metadata Problem:* A new software addition without updating the metadata repository will lead to incomplete analyses of future projects as well. If new processes are added without updating the metadata, similarly, it might lead to providing an incomplete picture of the operation of the system

*5) Data Quality Problem:* If the changes made are not compatible or comparable with the existing quality, it will lead to failures of the process in the future

*6) Documentation:* Changes made must also be updated in the documents with utmost care and if failed, will often result in leaving the production team in misery

*7) Configuration Management:* Changes may be introduced without updating the configuration management database, thus creating problems with the production and production support

### *D. Change Control Approach*

Changes made to the baseline of the system are subject to the approval of the configuration control board. Following are the steps that are followed to handle a request to change the baseline [7].

1) Submit the change request along with the information about financial cost and resources required, etc. These changes are then submitted to the change control board
2) Access the change request after the change of evaluation
3) Depending on the outcome of step 2, outpour request is either accepted or rejected. If the submitted information is incomplete, it can be deferred
4) If the change request is accepted, necessary planning is done to implement the change by assigning work to the developer team
5) Once the changes are validated and examined by the quality control team, all configuration items are then updated throughout the entire library and the baseline of the system

## IV. Requirements Traceability Matrix

The creation and implementation of requirements traceability techniques are done for the completeness, consistency, and traceability of the system requirements. It can be defined as the capability to define and trace the life cycle of a desired requirement, in both the backward and forward phases. Two methods are used for traceability cross referencing. One method can have cross referencing phrases such as 'see section A'. This method implements techniques such as numbering or tagging of requirements and changes in specialized tables or matrices. Each new addition is cross referenced at the older location in order to point towards these changes. The other method involves the restructuring of the documentation in terms of an underlying network of a graph that is used to keep track of the changes in the requirements. A traceability matrix is obtained by correlating requirements with the products in the software life cycle that satisfies them. Tests are correlated with the requirements of the designed product and the desired. These traceability matrices can be generated using multiple tools incorporating management software packages, spreadsheets, database packages or with hyper-linked tables on a processor. Configuration management plans are used on products since traceability is a key component of managing changes. Traceability ensures completeness such that all higher level requirements are assigned to lower levels and and all the lower level requirements are derived from higher level requirements [8]. The traceability matrix is a table which is 2×2 in size for most scenarios. These tables show a relationship between any of the two baseline files about the changes in requirements. These relationships maybe be one to many or bijection and displays the complete relationship. This method is used continuously in high-level projects that require specific requirements according to the high-level plan, test plan and test cases. The method commonly used is, during the first step, an identifier for each requirement of a file is taken and placed on the leftmost top column of the 2×2 matrix. The identifiers of the most related files are placed across the topmost row horizontally. When a requirement in the left column is related to a requirement in the topmost horizontal, a cross marx (×) is placed in the respective position where the intersection takes place. The number of relationships is summed up for each row and each column and written on the 2nd row and the 2nd column that indicated the value of the two mapping items. A sample traceability matrix is shown in table I [8].

TABLE I: Sample Traceability Matrix

| Identifier | Reqs. Tests | A 0.3 | A 0.4 | B 1.1 | B 1.2 | C 2.1 | C 2.2 | D 1.0 |
|---|---|---|---|---|---|---|---|---|
| Test Cases | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 3 |
| Implicit Tests | 7 | | | | | | | |
| 1.0.1 | 2 | | | × | | × | | |
| 1.0.2 | 2 | × | | | | × | | |
| 1.1.1 | 1 | | | | | × | | |
| 1.1.2 | 1 | | | × | | | | |
| 1.2.1 | 1 | × | | | | | | |
| 1.2.2 | 2 | | × | | | × | | |
| 1.3.1 | 3 | | | × | × | | × | |

The purpose of this matrix is to assist in ensuring that the requirement objectives are met by correlating each requirement with the object through the traceability matrix made for the requirement. In the forward trace, the matrix is used to validate that the affirmed requirements are distributed to system components and other deliverable products. It can also be used to conclude the source of the requirement in the backward trace. Requirement traceability matrix embraces outlining to things that convince the requirements such as design materials, capabilities, manual processes and analysis. This matrix is also utilized to make sure that all expected requirements are gathered. It is also used to establish the influence between system components when a modification is required. The capability to directly locate influenced components in the project allows the designers and developers to modify the project while ensuring maximum benefits while reducing costs and providing proper to-do estimates [8].

### A. *Tools for Requirements Management*

Requirements management tools are used to facilitate the process of requirements management. A toolkit can consist of one or more tools, each designed to keep track of processes that the human mind is simply incapable of doing. Following are a list of criteria that needs to be considered when designing a requirements management tool [1].

- Identification of 'individual' requirements
- Assignment to a destination and sorting of requirements
- Identification of requirement groups (collection), revision and base lining
- Provision of a basic data interface

### B. *Benefits of using Tools*

Following is a list of benefits that can be achieved through the use of specific requirements management tools instead of general purpose tools [9].

*1) Structured Requirements:* Only specific tools allow the gathering of 'structured' requirements. This means that it is possible to define attributes that would be helpful to track individual requirements and make sure that each requirement has its own set of attributes

*2) Save Time:* Good requirements management tools can save a lot of time by properly managing the software requirements. Since these tools are automated for most of the management tasks such as creating automated documentation, the time saving will be considerable

*3) Less Stress:* Most gathering and tracking of requirements are very chaotic in nature. A good requirement management tool can eliminate a lot of the unnecessary stress associated with the process

*4) Work flow and Best Practices:* Built-in methods in these specialized tools enable an efficient work flow while adhering to best practices in the field automatically

*5) Easy to Collaborate:* A good requirements management tool enables collaboration among external and internal stakeholders effectively. This is one area where general purpose tools lack significantly

*6) Increase in Precision of Requirement:* Good tools increase the exactness between customer requirement and the project output. If offers trouble free implementation in such a way that the recording and the supervision of the customer requirement is easily understandable

*7) Cost Reduction:* Tools have the ability to reduce maintenance costs, training costs, deployment costs and well as reduce the deployment risks and the time required

*8) Added Benefits:* These tools can increase the collaboration between the team and support services and thereby, deliver better solutions in lesser times

## V. Conclusion

Depending on the literature available, it is clear that the use of requirements management techniques is highly beneficial to the software industry. Usually, the process begins with the analysis and elicitation of the objectives and the constrains of the project and the organization. Once this portion is complete, the next step is using change control management. Change control management is an important process that can deliver immense benefits. By using the traceability matrix, it is easy to identify the relationship between different requirements of the project. Once these objectives are clear, it is possible to address these requirements issues both in the forward and backward traces. The process of requirements management can further be enhanced through the use of specific software tools that can be utilized to save both time and money while increasing the quality of the output.

### A. *Recommendations*

While it is difficult to present a global recommendation for all projects, it is obvious that the use of these techniques have a direct positive impact on the productivity. With this in mind, each software company must try their best to enhance their capabilities using these methods. However, it is doubtful whether sophisticated software tools will be beneficial to all software projects. One major issue regarding these tools is the cost factor. It is possible for the tool cost to be as large as the budget in some cases. Due to this reason, it is important to properly understand the scope of the project. If the project involves the designing and implementation of a large system which requires regular updates, or if the same system is to be sold to multiple markets with minor customisations, it is important to invest on these specific tools. However, regardless of the project size, it is always important to invest time on the requirements management process. If the budget is small, it is best to use an all-purpose software such as Microsoft Excel, but nonetheless, the task must still be done. As it was mentioned, requirements management has a direct impact on the documentation process, and even if the developers are developing something as small as a mobile application, it is important to understand the significance of the documentation. Based on these points, it is highly recommended for all software developers to adopt principles of software requirements management to their development life cycle, regardless of the size of the project.

### B. *Future Work*

This paper presented a review of literature in order to highlight the pros and cons of software requirements management. Through the review, it is clear that software requirements management has become an integral part of the software development life cycle. Future work in this regard should include a review of the actual software packages that are being used in the industry at the moment for the task of software requirements management. Such an analysis should quantify the possible parameters of these applications. Given the highly dynamic nature of the field, it will obviously be futile to attempt naming the best software available, however, it is possible to obtain some qualitative data from the end users from various fields and present a recommendation as to which software package is the best for a given task.

## REFERENCES

[1] D. Leffingwell and D. Widrig, *Managing software requirements: a use case approach*. Pearson Education, 2003.

[2] K. L. Evans, R. P. Reese, and L. Weldon, "Unit information management practices at the joint readiness training center," DTIC Document, Tech. Rep., 2007.

[3] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[4] B. W. Boehm, "Software risk management: principles and practices," *Software, IEEE*, vol. 8, no. 1, pp. 32–41, 1991.

[5] M. Dumas, W. M. Van der Aalst, and A. H. Ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. John Wiley & Sons, 2005.

[6] S. G. Eick, T. L. Graves, A. F. Karr, J. S. Marron, and A. Mockus, "Does code decay? assessing the evidence from change management data," *Software Engineering, IEEE Transactions on*, vol. 27, no. 1, pp. 1–12, 2001.

[7] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.

[8] B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *Software Engineering, IEEE Transactions on*, vol. 27, no. 1, pp. 58–93, 2001.

[9] S. J. Andriole, *Managing systems requirements: methods, tools, and cases*. McGraw-Hill Companies, 1996.