

Importance of Requirements Engineering in Making Your Project a Success

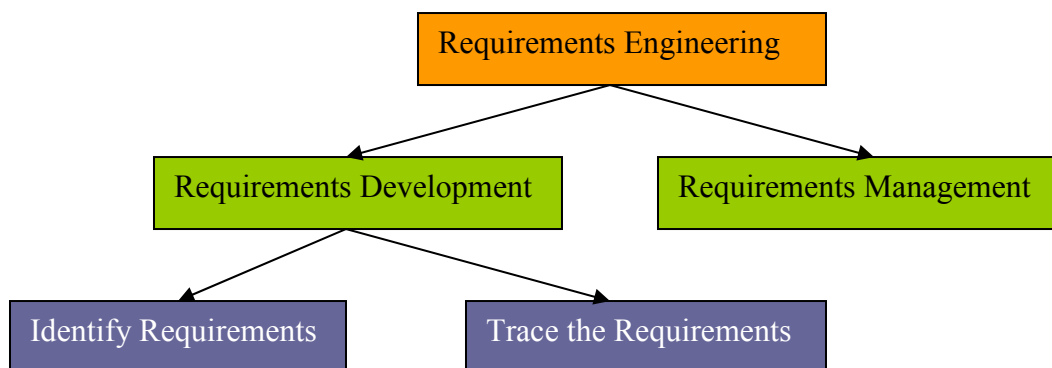
1 Introduction

"I'll go find out what they want, and the rest of you start coding." This caption from a cartoon is very close to the way some software organizations still treat the requirements specification process. In many companies, requirements management is a difficult challenge. They focus their efforts on design, development and testing but not on managing requirements. Some organizations may perceive requirements management as an activity only for organizations that have complex products and large staffs to support the effort.

Many good reasons exist for implementing a formal requirements management process like having the same understanding of the requirements in the whole team, finding defects earlier in the development process, keeping product definitions up to date, and communicating change to the development team.

2 Requirement Engineering

Various activities are performed in the software development life cycle with respect to requirements. The diagram illustrates the activities, which can be classified under requirements engineering.



Let us see the importance of each of these activities for a successful project.

2.1 Identify Requirements

Requirements engineering is primarily a communication, not technical, activity. Technical problems can begin early on if project participants don't have the common understanding of the requirements for the project

. Fred Brooks said:

“The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part of the system is more difficult to rectify later.”

Brooks, F., “No Silver Bullet: Essence and Accidents of Software Engineering”, *IEEE Computer*, Vol. 20, No. 4, April 1987, p. 10-19.



Imagine if instead of developing software, we are building an expensive house. Wouldn't we expect that before we started pouring the foundation, we would have detailed drawings that showed what the finished house should look like, inside and out? Of course we would.

How does building houses relate to developing software? When we build a house, there is usually a general contractor (on software projects we call this person the project manager), there are skilled trades people like carpenters, plumbers, electricians (on software projects there are developers, QA engineers, and tech writers). When building a house, it must pass several inspections to ensure that the work meets the national and local building codes (inspections on software projects). So, there are striking similarities between building houses and building software.

2.1.1 Getting the requirements right is the first step to getting the software right.

Without well-written, complete, and unambiguous requirements, it's difficult for the software engineers to know what to build and for testers to know what to test and to customers to what to expect.

Understanding the requirements is also a specific practice that has been recognized by the Software Engineering Institute (CMMI Level 2 Process Area)

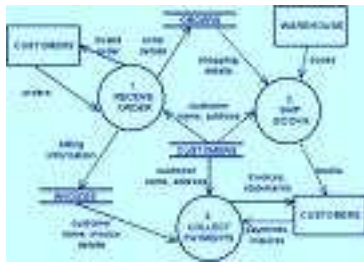
2.1.2 Define the Product's Business Requirements

Nearly everyone has worked on projects that suffered from scope creep, yet few of these projects had a documented scope or vision. How do we even know that the scope is creeping if no one ever clearly defined it? Documented business requirements will help to answer the first question to ask whenever someone proposes new functionality: "Is this in scope?"

2.1.3 Involve the Customer

The project team members are not experts in the specialized application domains for which the application is developed. Consequently, it is not reasonable to expect the engineers to make sound business or technical decisions on behalf of the customers, or to resolve conflicting requirements supplied by different end users, or to set priorities for the many requirements that might be collected. Hence the customer involvement is the most critical factor in software quality.

2.1.4 Requirements Analysis



A picture is worth thousand words. Requirements analysis includes decomposing high-level requirements into detailed functional requirements, constructing graphical requirements models, and building prototypes. Analysis models and prototypes provide alternative views of the requirements, which often reveal errors and conflicts that are hard to spot in a textual SRS.

Another important analysis activity is to prioritize the requirements.

2.1.4.1 Prioritize the requirements

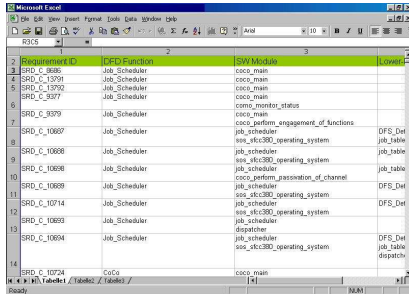
Prioritization helps the project manager to plan for staged releases, make trade-off decisions, and respond to requests for adding more functionality. Every project must establish the relative priorities of the requested features or functional requirements. This helps the developers to implement the highest priority requirements first. In the worst case of schedule slippage, the customer's highest priority requirements would have been implemented.

The objective of identifying requirements is to communicate a shared understanding of the new product among all project stakeholders. This understanding is captured in the form of a textual SRS written in natural language, augmented by appropriate analysis models. This document should be base lined and made available to the entire team.

2.2 Trace the requirements

Consider the following questions:

1. Has the system been adequately tested?
2. How will a change to a requirement impact other requirements and work products?
3. For testers: Which all requirements are implemented by the unit/module under test?



Req ID	Req Description	Work Product	Owner
BRD_C_0000	Job Scheduler	coco_main	
BRD_C_1001	Job Scheduler	coco_main	
BRD_C_1002	Job Scheduler	coco_main	
BRD_C_1003	Job Scheduler	coco_main	
BRD_C_1004	Job Scheduler	coco_main	
BRD_C_1005	Job Scheduler	coco_main	
BRD_C_1006	Job Scheduler	coco_main	
BRD_C_1007	Job Scheduler	coco_main	
BRD_C_1008	Job Scheduler	coco_main	
BRD_C_1009	Job Scheduler	coco_main	
BRD_C_1010	Job Scheduler	coco_main	
BRD_C_1011	Job Scheduler	coco_main	
BRD_C_1012	Job Scheduler	coco_main	
BRD_C_1013	Job Scheduler	coco_main	
BRD_C_1014	Job Scheduler	coco_main	
BRD_C_1015	Job Scheduler	coco_main	
BRD_C_1016	Job Scheduler	coco_main	
BRD_C_1017	Job Scheduler	coco_main	
BRD_C_1018	Job Scheduler	coco_main	
BRD_C_1019	Job Scheduler	coco_main	
BRD_C_1020	Job Scheduler	coco_main	
BRD_C_1021	Job Scheduler	coco_main	
BRD_C_1022	Job Scheduler	coco_main	
BRD_C_1023	Job Scheduler	coco_main	
BRD_C_1024	Job Scheduler	coco_main	
BRD_C_1025	Job Scheduler	coco_main	
BRD_C_1026	Job Scheduler	coco_main	
BRD_C_1027	Job Scheduler	coco_main	
BRD_C_1028	Job Scheduler	coco_main	
BRD_C_1029	Job Scheduler	coco_main	
BRD_C_1030	Job Scheduler	coco_main	
BRD_C_1031	Job Scheduler	coco_main	
BRD_C_1032	Job Scheduler	coco_main	
BRD_C_1033	Job Scheduler	coco_main	
BRD_C_1034	Job Scheduler	coco_main	
BRD_C_1035	Job Scheduler	coco_main	
BRD_C_1036	Job Scheduler	coco_main	
BRD_C_1037	Job Scheduler	coco_main	
BRD_C_1038	Job Scheduler	coco_main	
BRD_C_1039	Job Scheduler	coco_main	
BRD_C_1040	Job Scheduler	coco_main	
BRD_C_1041	Job Scheduler	coco_main	
BRD_C_1042	Job Scheduler	coco_main	
BRD_C_1043	Job Scheduler	coco_main	
BRD_C_1044	Job Scheduler	coco_main	
BRD_C_1045	Job Scheduler	coco_main	
BRD_C_1046	Job Scheduler	coco_main	
BRD_C_1047	Job Scheduler	coco_main	
BRD_C_1048	Job Scheduler	coco_main	
BRD_C_1049	Job Scheduler	coco_main	
BRD_C_1050	Job Scheduler	coco_main	
BRD_C_1051	Job Scheduler	coco_main	
BRD_C_1052	Job Scheduler	coco_main	
BRD_C_1053	Job Scheduler	coco_main	
BRD_C_1054	Job Scheduler	coco_main	
BRD_C_1055	Job Scheduler	coco_main	
BRD_C_1056	Job Scheduler	coco_main	
BRD_C_1057	Job Scheduler	coco_main	
BRD_C_1058	Job Scheduler	coco_main	
BRD_C_1059	Job Scheduler	coco_main	
BRD_C_1060	Job Scheduler	coco_main	
BRD_C_1061	Job Scheduler	coco_main	
BRD_C_1062	Job Scheduler	coco_main	
BRD_C_1063	Job Scheduler	coco_main	
BRD_C_1064	Job Scheduler	coco_main	
BRD_C_1065	Job Scheduler	coco_main	
BRD_C_1066	Job Scheduler	coco_main	
BRD_C_1067	Job Scheduler	coco_main	
BRD_C_1068	Job Scheduler	coco_main	
BRD_C_1069	Job Scheduler	coco_main	
BRD_C_1070	Job Scheduler	coco_main	
BRD_C_1071	Job Scheduler	coco_main	
BRD_C_1072	Job Scheduler	coco_main	
BRD_C_1073	Job Scheduler	coco_main	
BRD_C_1074	Job Scheduler	coco_main	
BRD_C_1075	Job Scheduler	coco_main	
BRD_C_1076	Job Scheduler	coco_main	
BRD_C_1077	Job Scheduler	coco_main	
BRD_C_1078	Job Scheduler	coco_main	
BRD_C_1079	Job Scheduler	coco_main	
BRD_C_1080	Job Scheduler	coco_main	
BRD_C_1081	Job Scheduler	coco_main	
BRD_C_1082	Job Scheduler	coco_main	
BRD_C_1083	Job Scheduler	coco_main	
BRD_C_1084	Job Scheduler	coco_main	
BRD_C_1085	Job Scheduler	coco_main	
BRD_C_1086	Job Scheduler	coco_main	
BRD_C_1087	Job Scheduler	coco_main	
BRD_C_1088	Job Scheduler	coco_main	
BRD_C_1089	Job Scheduler	coco_main	
BRD_C_1090	Job Scheduler	coco_main	
BRD_C_1091	Job Scheduler	coco_main	
BRD_C_1092	Job Scheduler	coco_main	
BRD_C_1093	Job Scheduler	coco_main	
BRD_C_1094	Job Scheduler	coco_main	
BRD_C_1095	Job Scheduler	coco_main	
BRD_C_1096	Job Scheduler	coco_main	
BRD_C_1097	Job Scheduler	coco_main	
BRD_C_1098	Job Scheduler	coco_main	
BRD_C_1099	Job Scheduler	coco_main	
BRD_C_1100	Job Scheduler	coco_main	

It is not easy to answer these questions without a suitable traceability matrix. If you have traced the requirements, then answering these questions becomes easy. Tracing requirements entails documenting contextual links between the various requirements and between the work products developed to implement and verify the requirements. Traceability matrix serves as a

graphical representation of traceable relationships between requirements and work products.

Let us try to answer the above questions with the help of a traceability matrix:

1. Tracing customer requirements through development and testing verifies that the customer requirements are implemented and tested, diminishing the risk of missing stated or derived requirements, especially when developing large, complex systems.
2. In case of a change in the requirement, traceability matrix helps to identify all the requirements and components of the process such as design specifications, code and test cases that need modification to fulfill the change request. This ensures that all affected work products are modified to support the change request.
3. During testing, it is important for the testers to know the requirements implemented by the unit/module they are testing. Without a well-documented traceability matrix it is not possible for the tester to find the requirements using which he/she has to write the test cases. If the tester does not recognize the requirements for the unit/module under test, then the test cases he writes to test the unit/module will not be adequate.

2.3 Requirements Management

The project will go haywire if the requirement changes are not analyzed properly. If there is a requirement change at any phase of the project and if the change is accepted without analyzing it then it might create a very big problem for the project. Hence before accepting any requirement change it is necessary to analyze the effort required

to implement the change and to decide whether the change is within the defined scope. Without a proper defined requirement management process it is very difficult to manage the changes in the requirements.

It is important for every organization to document a process that describes how a proposed change will be submitted, evaluated, decided upon, and incorporated into the requirements baseline. The change control process can be supported with a problem- or defect-tracking tool, but a tool is not a substitute for a documented process.

It is also important to establish a change control board of the decision-makers who approve or reject each proposed change. This board could be just one person. More typically, a body representing diverse perspectives (such as development, management, customer, and testing) is appropriate.

3 Epilogue

Without a suitable requirement engineering process it is very difficult for the organizations to deliver successful projects. Although there are plenty of obstacles to put requirements engineering process in place and plenty of reasons to not follow the process, the advantages of conforming to the process outnumber them. Instead of manually performing the requirements engineering activities, they can be automated with the help of an appropriate tool. However that is a topic for another paper.

4 References

1. <http://www.unicom.co.uk/>
2. <http://home.jtan.com>
3. <http://www.processimpact.com/>

Pankaja P K

Manager - Software

She heads the SmartWorks and ReMa teams at Accord. She also manages DO178B verification projects.

Write to her at [feedback at ReMa](#)