RESEARCH ARTICLE OPEN ACCESS

Requirement Analysis Practices Are the Foremost Way to Bridge the Dissonance between Project Success

Vallabhajosyula Bala Naga Venkata Sai [1], Kiran Kumar Reddi [2]
Research scholar [1], Rayalaseema University, Kurnool
Department of Computer Science& Engineering [2], KrishnaUniversity
Machilipatnam - India

ABSTRACT
The possibility of software projects failing can be attributed to various reasons like costs, scheduling and quality issues, and/or achieving of conceptual work is a difficult as establishing the detailed technical requirements. The scope of any project should represent the minimum amount of work needed to in order to fulfil the end user's requirements. If we do the less work when compare to the user's requirement it is not fulfilled to the client/customer. If we do the more work when compare to the user's requirements then it is wasteful work to the client/customer. Therefore, to get the perfect scope of the project/product from the client in right means and gathering the requirements rightly at first! And that means, and requirements asking the right people the right questions.

Keywords :— Software Project life cycle, Requirements, Design, software quality,

I. INTRODUCTION

In today's highly dispersed global software development, meeting the Customer needs is not so easy by doing the requirement engineering. Customer is always need to consider a important role in the project, the customer need to be in active in the throughout the project life cycle that it will leads to get the quality of the product. This doesn't mean that the project stake holders are unable to bring the success of the project and are not the reason for project failure. A customer involvement in the requirements and design phase strengthen the project from starting phase, leading to proper line of development at the all the later phases of the project for a quality end product.

This paper speaks on the following research questions for delivering a quality project/ product to customer.

1. Requirements, why requirements are needed?
2. Requirements, what are the characteristics of it?
3. Requirements, derived from customer are not know to him/her?
4. Requirements, change of requirements at the time of project progress?
5. Requirements, what is their role in the software quality?
6. How to overcome unreasonable timelines of the customer?
7. How much efficient is the project manager to fill the communication gap between the stake holders?

II. R^{5}HH PRINCIPLE FOR A SUCCESSFUL PROJECT

A. Requirements, why requirements are needed?

A requirement is an attribute of a product, service or system necessary to produce an outcome(s) that satisfies the needs of a person, group or organization. Requirements therefore define "the problem." In contrast, "the solution" is defined by technical Specifications.

Defining requirements is the process of determining what to make before making it. Requirements definition creates a method in which appropriate decisions about product or system functionality and performance can be made before investing the time and money to develop it. Understanding requirements early removes a great deal of guesswork in the planning stages and helps to ensure that the end-users and product developers are "on the same page."

Every project has some basic requirements that defines what the end users, customers, clients, developers, suppliers or business (i.e., stakeholders) require from it coupled with some needs of the system for efficient functioning requirement is a key factor during every software development as it describes what different stakeholders need to how the system will satisfy these needs. The client is generally expressed in natural language so that everyone can understand it will. It helps the analyst to better understand which elements and functions are necessary to develop that project/product

As per the Chaos report, on an average 50-60% of the projects are challenged and approximately 20% of the projects are failed leaving the rest as successful. Projects to be executed are always having one or other issues like insufficient requirement's or product being shelved before deployment for not given enough study on the requirement's. This lead to increase the dissatisfaction levels of the customer. Another point can be considered like, many a times by not giving proper attention to the design of the product, like

pseudo code analysis or care not given for the design of the product by properly considering the requirements. Top most priority needs to be given for the requirements and design phases, which helps in resolving many issues down the line in the project. As per chaos report the successful rate is 39% for agile and 11% for waterfall. [Chaos, 2015].

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Fig. 1 Chaos resolution report comparing Agile and waterfall.

Requirements engineering can be simply described as identifying a problem's context, locating the customer's requirements within that context and delivering a specification that meets customer needs with in that context.

As Cheng and Atlee describe [1] "Successful Requirements Engineering involves understanding the needs of all stakeholders; understanding the contexts in which the to-be-developed software will be used; modelling, analyzing, negotiating, and documenting the stakeholders' requirements; validating that the documented requirements match the negotiated requirements; and managing requirements evolution". Looking at this definition of the success of requirements engineering processes, we notice that the concept understanding is the single most important aspect of requirements engineering. Central question here: "Does the requirements engineer understand what a stakeholder needs?". It's this understanding what makes or breaks the success of requirements engineering processes. Requirements engineering is difficult [1]. In general, the type of challenges concerning requirements engineering are slightly different from those concerning software engineering. Requirements engineering challenges reside mainly in the problem space, where software engineering challenges reside in the solution space. This leads us to another requirement engineering definition [15]: "Requirements engineering is about defining precisely the problem that the software is to solve"

B. Requirements, what are the characteristics of it?

The importance of the good requirement gathering/engineering may not be obvious at first glance and, indeed, even the purpose of requirements can be rather opaque, Requirements engineering represent a translation of customer needs. Documenting good requirements early in the process, helps to ensure the project stays on track and delivers the expected and agreed upon result.

Requirements engineering is difficult and time-consuming, but must be done well if the final product or system is to be judged by the end users as successful. From the International Council of Systems Engineers (INCOSE) Requirements Working Group, here are seven attributes of good requirements:

Approach to requirements	**Runaway probability**
Projects that Ignore requirements	7 out of 10
Projects that Capture requirements in document form	5 out of 10
Projects that Capture requirements in a database	4 out of 10
Projects that capture Requirements with a tracking tool	4 out of 10.

Fig. 2 Approach to requirements

the stakeholders. Project scope creeps, higher the project cost and delay the project. Standish Group defined project success as being delivered on time, staying on budget, and reaching completion with all planned features intact. Quick project completion, staying on budget and timeliness delivery of project can be achieved by practicing the following method.

Necessary	Can the system meet prioritized, real needs without it? If yes, the requirement is not necessary.
Verifiable	Can one ensure that the requirement is met in the system? If not, the requirement should be removed or revised.
Unambiguous	Can the requirement be interpreted in more than one way? If yes, the requirement should be clarified or removed. Ambiguous or poorly worded requirements can lead to serious misunderstandings and needless rework
Complete	Are all conditions under which the requirement applies stated? In addition, does the specification include all known requirements?
Consistent	Can the requirement be met without conflicting with any other requirement? If not, the requirement should be revised or removed.
Traceable	Is the origin (source) of the requirement known, and is there a clear path from the requirement back to its origin?
Concise	Is the requirement stated simply and clearly?

Factors to consider in developing a set of requirements

Ask potential users of the projects deliverables what they need. Consider each potential user group separately as their needs will be different.

Define and prioritize the required technical functions for each deliverable.

Be sure that organization process is mature and stable, and then be sure that the deliverables can support the processes. This may suggest a less sophisticated deliverable, or alternatively indent addition requirements for process improving and training.

Consider how the deliverables should integrate with other tools and processes used by the organisation and their wider business.

Define how the deliverables should work once completed and the degree of standardization or flexibility needed both in the short term and the longer term.

Consider what training the users might need to be able to use the deliverables properly. Who will be responsible for the training needs analysis, developing the training materials, training the trainer and training the users?

Decide on scalability and the ability to manage the full spectrum of current needs and likely longer-term needs. Should the deliverable be a 'one-size-fits-all solution or focused on a particular type or size of need/problem/requirement. The cost and complexity of the project will vary significantly.

Think about what ongoing support the organisation might need from external suppliers and vendors.
Assess how the chosen solution might need to grow with your business.

Determine the budget, including purchase/development of the deliverable, possible customization, piloting, data cleansing/migration, training, communication and roll-out. But cost should not be the driving consideration- you will get what you pay for and you should buy what you need based on the best value proposition

C. ***Requirements, derived from customer are not know to him/her***

Possibly the most common problem in the requirements anaylysis phase is that customers have only average idea of what they need, and it's up to you to ask the right questions and perform the ananlyis necessary to turn the customer vision into a software requirement specification document. To achieve this, we should follow the below points.
Spend your maximum time to understand the goal, objectives and scope of the project and the starting phase of the project.

Attempt to write a solid vision of the project, which is having both he functions and the user benefits and need to cover all the business problems which the customer is expected to solve.

Send to our client/customer to read and analyze/re think about to completed software requirements specifications to align expectations and discuss that both parties have a clear understanding of the product deliverables

D. ***Requirements, change of requirements at the time of project progress.***

The second most common and important problem with the software projects is that the requirements defined at the beginning phase change at the project progresses. This is due to the development progresses and prototype are developed, at that time customer can see the problems and compare with the original requirements and do the necessary corrections.
It may also occur because of the changes in the domain conditions/ technical problems, it may reshape/rephrase the problem and hence the necessity to provide a change /different solution that will be proposed earlier by the company.

Good Project managers should be aware of the above-mentioned problem/possibility and should be have a backup plan to deal with these kinds of changes.

To solve the above problem

We should have a pre-defined process to receiving, analyzing and implementing /incorporating the CR (Change Requests), which did not affect to the entry point into this process.

We need to set a milestone for each development phase beyond which certain changes are not to be permit, for example If a module/phase reaches the maximum level of requirements done in that phase, we don't need to allow the changes up to that phase is to complete.

Ensure that change requests and approvals are clearly communicate with all the stakeholders and get the feedback with their rationale, after that we need to update the master plan accordingly.

E. ***Requirements, what is their role in the software quality?***
Requirements are play a vital role in the software quality. Quality management is a complex job, involving the co-ordination of many processes and techniques. Some of these are manually dependent and required for every project.

Quality is heavily depended on requirements, each time there is a change, quality is revisited to reflect the changes in the project/product on account for them in the quality assurance.

F. How to overcome unreasonable timelines of the customer?

The above-mentioned problem is the common problem. Customer said/requested that its' an important job we need to complete this in a minimum time frame, the project manager/company will agree to such timelines before actually performing a detailed analysis on the work what we have to do that is scope of the project and also without proper planning (Cost, Resources, technical possibilities), with the agree of the project manager/company the resources are start working and complete it with may/may not without proper quality testing/inspection on the work.

To solve this problem, you should.

Convert the SRS into the perfect project plan with detailed tasks and the resources needed to accommodate at each stage which will need to deal in all the possible conditions (Good, better, Best, and worse) conditions.

The project manager needs to take the project plan account of available resource constraints and keeps sufficient time for the testing and the quality analysis of the product.

G. ***How much efficient is the project manager to fill the communication gap between the stake holders?***

Stakeholders and communicate with each other because they come from different platforms, and some have understood the technical terms and someone did not have in the same way. This can lead to confusion and a miscommunication especially at the time of requirements phase, which needs to take care by project manager.

To Solve the above problem

Make a note of every meeting minute and send to all the stake holders, in a detailed way and stick to them consistently.

III. CONCLUSION

If a project is success is determined by whether or not quality targets are met with in the budget and on time, product success is determined by whether or not customer expectations are met.
Functional requirements that do not meet customer expectations may yield a successful project that results in an unsuccessful product/project. Requirement analysis practices are the foremost way to bridge the dissonance between project success. it is not necessarily having complete requirements at the start of the project but rather completing the requirements at any of the stage during the project execution and the projects which has a central repository for requirements are more likely to succeed.

References

[1] *Chaos https://www.standishgroup.com/*

[2] *SpecIndia blog, https://www.spec-india.com/blog/from-waterfall-to-agile-to-devops-a-cultural-and-technological-shift/*

[3] Pohl, K. (2007). *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. Heidelberg: dpunkt.verlag GmbH.

[4] Pressman, R. S. (2004). *Software engineering: A practitioner's approach*. (6th ed.). New York, NY: McGraw Hill.

[5] Pressman, R. S. (2009). *Software engineering: A practitioner's approach*. (7th ed.). New York, NY: McGraw Hill.

[6] Robertson, S., & Robertson, J. (2006). *Mastering the requirements process*. (2nd ed.). Westford, MA: Pearson Education Inc.

[7] Wiegers, K. E. (2003). *Software requirements*. (2nd ed.). Redmond, Washington: Microsoft Press.