

How to get software requirements right



**ASSIGN
BUSTER**

How to get Software Requirements right Abstract The importance of getting the software requirement right has made big difference to so many major projects in the IT industry. Many techniques and solutions have been developed for understanding and getting software requirements right first time in the software development life cycle, but fewer people are taking these techniques into account and this is the reason why most of the IT projects are failing. I define here how to get software requirements right and why it has become some important to get software requirements right.

Introduction Requirements Analysis is known as the process which allows us to understand the customer needs and expectation from proposed software. It is the first and one of the main stages in the Software Development Life Cycle model. Software requirements are functional and non-functional requirements of the system, which includes a brief description of how a system should behave as well as brief descriptions of a system's functions. Software requirements can state what an application is expected to do.

Outsource2India pointed out that " the software requirement analysis process includes the complex task of eliciting and documenting the requirements of all the users, modelling and analysing the requirements and documenting them as a basis for system design". There are different level and types of software requirement, which should be defined to get software requirements right. As shown in Figure 1, Business level requirements are there to define the business problem, business opportunities as well as what business is requesting as a solution of the problem.

A user level requirement looks at the functionality of the software from user's view. Product level requirements are where we define functional and

non-functional requirements of the system such as the functionalities of the software and the usability needs it should meet. The reason why Software Requirements is known to be very important and major role in developing a project is because it gives the developer a brief description of what he/she needs to develop. It gives them better understanding of a system, which results in better system.

Eliciting, analysing and writing good requirements is the hardest and most difficult part of building a software system because if you don't get the requirements right, it doesn't matter how well you do anything else. It is said Outsource2India that " Software companies are now investing time and resources into effective and streamlined Software Requirements Analysis Processes as a prerequisite to successful projects that align with the client's businessgoalsand meet the project's requirement specifications".

If the requirements of software are incomplete, software practitioners will end up building software that does not meet the needs of the customer and the user. State of the Art As mentioned earlier, requirement analysis is one of the most important phases in the entire software development life cycle. It is very important to identify and understand the software requirement before moving on to other phases of software development life cycle. It was said by Pooja. R. Gupta that " When project managers plan for the software development they need to plan for adequate time and resource allocation for this phase".

Experienced requirement analyst like Hooks and Ferry has said that result of getting requirement right, early in the project, can save one-third or more of the overall project budget, which again states how important it is to get

software requirements right in order to save time, money as well as achieve goals of the software. Another experienced requirement analyst Leffingwell points out that if you get your software requirements right at the beginning then it can save you the cost of fixing requirements errors as the cost of fixing requirements account for 70% to 80% of your rework costs.

Getting requirements right is the most critical aspect of the software development cycle. Software Productivity Centre mentioned that most of the companies come up short; studies point to a failure rate of more than 60% for IT projects, with poor requirements as one of the top five reasons. Moreover, when requirements are specified early in the lifecycle, 80% of the functionality is relatively unwanted by the users and 45% of these features are never used. These problems mean costly rework, compromised product quality, delayed delivery on current projects and start of new ones, lost credibility and lowered revenue opportunities.

Developing and managing requirements at the start of a software project can provide significant benefits to the whole outcome:

- Quicker and easier to develop and deliver high value products.
- Greater insights into development capacity and capabilities.
- Better understanding of the organisation and customer needs.
- Higher business and user satisfaction with the overall product.
- Closer alignment with business goals and expectations.
- Cost saving

- Result of the product is what the company actually required so get a product that meets the requirements of the system. Reduces rework and conflicts which would have come from unclear and ambiguous requirements.

How to get the requirements right? According to Pooja. R. Guptam, in order to get the software requirements right, it can be very beneficial to cover all

the following aspects of requirement analysis: 1. Define the problem and High level description of the solution: It is very important to understand the problem and the need for solution. The requirement analyst needs to understand how the new solution going to help and what benefits it is going to provide to the company. 2.

Cover needs of all stakeholders and users of the solution: At this point, you must identify and understand the needs of the stake holders as well the users who are ultimately going to use the new system because you need to produce the system, which meets user requirements. 3. Define what the solution should and should not do: It is very obvious that we might know what the solution should do but we always forget the we also need to understand and define what the solution will not do so that requirements can be stated in a manner that we won't be left with any imagination or doubts. 4.

Define the features required: This is one of the main tasks of a requirement analyst. It is very important to define the functional and non-functional requirements of the software solution before starting to design the solution. Defining the functional and non-functional requirement is not all that needs to be done; you must also make sure that they are the correct requirements. 5. Capture all supporting information: It's a good idea to include and record details of any processes followed, workflow, information flow, etc. It is also beneficial to record any other information which might be relevant to the solution.

Conclusion In my opinion, I believe software requirements play quite a big role in the whole software development life cycle. The experience of <https://assignbuster.com/how-to-get-software-requirements-right/>

developing such software project has made it so clear how important software requirement can be. Defining software requirements earlier in the project can provide the best solution of a system at the end. In my personal experience of developing software projects has given me the idea of how to get software requirements right, which are to firstly create requirement definition because it is the foundation of effective software delivery.

Secondly, define requirements among stakeholders to address their needs, business problems and the vision of the software so that stakeholder's goals and objectives can be met because if stakeholders requirements are not met or unclear then the developer will have difficulty building the solution without rework, which results in longer development lifecycles and higher costs. By investing more time on software requirements early in the software development process can save time, effort as well as money in the short and long terms.

By following these paths in developing software projects can lead to a successful system. As mentioned earlier about the types of requirements that should be identified before moving to other part of software life cycle. It is very important to define all the business level, user level and the product level requirements because a developer needs all of these requirements in order to develop the right software. It is a very idea to follow the software development life cycles, which defines that requirement analysis is the first process of developing a project.

In my previous projects, I believe a software development model has helped in defining the right requirements so it is a good advice to take this into consideration when designing any kind of software projects. Requirement

Definition process also helps in defining the right requirement. References
James A. Ward <http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=ART&ObjectId=9150&tth=DYN&tt=siteemail&iDyn=2>
Software Productivity Centre http://www.spc.ca/dc_software_requirements.htm
Pooja R. Gupta <http://ezinearticles.com/?Five-Tips-For-Getting-the-Software-Requirement-Analysis-Right-the-First-Time&id=1714473>

IBM http://www.utdallas.edu/~chung/RE/Getting_requirements_right-avoiding_the_top_10_traps.pdf
Brianna Smith, delivery engagement manager, Rational software, IBM Software Group
Lisa Garrity, technical professional, Rational software, IBM Software Group
Theresa Kratschmer, senior software engineer, Rational software, IBM Software Group <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1084/Getting-Your-Requirements-Right-Collaborate-With-Stakeholders-To-Work-Smarter.aspx>
My project Management <http://myprojectmanagement.com/category/requirement-analysis/>