Hugh Hoang 1833106

Software and product failure is a broad, multi-faceted and identified "changing and/or complicated area of exploration. That requirement definition, or identifylng the problem to solve for, Is one of the thorniest aspects of building software. The fast pace of technology evolution in computing, a phenomenon we now broadly refer to as Moore's Law, has also meant that few projects are ever repeated with comparable technologies or under comparable circumstances. An early criticism of agile approaches was that it avoided the problem of poorly defined requirements by simply not seeking it. In short, agile techniq ues on their own often require augmentation and adjustment it is important notto ig nore the role of software architecture and team organization in creating flexible systems that can be adjusted rapidly and evolved to meet changing requirements. A long running software principle, known as divide-and-conquer. calls for problem decomposition to where the individual components become more tractable and manageable. Project failure is expensive. Sunk "Project failure is expensive. costs In a failed project add to the Sunk costs in a failed project enormous missed opportunity cost add to the enormous missed of a successful project opportunity cost of a successful project.Even with the best methodologies, software architectures and organizational approaches, software developers would do we l to invest more time and effort into understanding the requirements of the systems they plan to build, and investing more time and effort In the infrastructure and processes to test these systems. Requirements must be determined and agreed to by the customers, users, and suppliers of a software product before the software can be built. What the software must do to add value for its stakeholders. These functional requirements define the capabilities of the software product. Business requirements define the business problems to be solved or the business opportunities to be addressed by the software product. User requirements look at the functionality of the software product from the user's perspective. The product's functional requirements that define the software functionality must be built into the product to enable users to accomplish their tasks, thereby satisfying the business requirements.Eliciting, analyzing, and writing good requirements are the most difficult parts of software engineering. There are many issues that can have a negative impact on software development projects and products if practitioners don't do a good job of defining their software requirements. These issues include: Incomplete requirements, Lack of user involvement,Requirements churn,Wasted resources , Gold plating, Inaccurate estimates. The acquirer type stakeholders can be divided into two major groups. First there are the customers who request, purchase, and/or pay for the software product in order to meet their business objectives. The second group is the users, also called end-users, who actually use the product directly or use the product indirectly by receiving reports, outputs, or other information generated by the product.The suppliers ofthe software product include individuals and teams that are part ofthe organization that develops the software product or are part ofthe organizations that distribute the software product or are involved in other product delivery methods. Identifying and considering the needs ofall ofthe different stakeholders can help prevent software product requirements from being overlooked. The requirements analyst will never know as much about a stakeholder's work as that stakeholder. By identifying and involving key stakeholders, the analyst gains access to their experience base and domain knowledge. Requirements development encompasses all the activities involved in identifying, capturing, and agreeing upon the requirements.