Pak. j. eng. technol. sci.
Volume 6, No 1, 2016, pg 1-6
ISSN: 2222-9930 print
ISSN: 2224-2333 online

Requirement Correctness Problems and Strategies for Web Applications

Muhammad Athar Naeem
Bahria University
Karachi, Pakistan
m_atharnaeem@hotmail.com

Engr. Usman Waheed
Bahria University
Karachi, Pakistan
uwaheed@bimcs.edu.pk

Syed Furqan Ali Raza
Bahria University
Karachi, Pakistan
Naqvisyed092@gmail.com

ABSTRACT

Correctness means that application planned tasks as defined by its specification. This research paper conversing that if poor requirements are not executed in the application there is some difficult existing in the application. Some other resembling problems like ambiguous requirements and inappropriate constraints also exist in the application. For these problems some solution techniques provided like documentation, verifying techniques of requirements, and use case of completeness validation.

KEYWORDS---Requirements quality, Ambiguous requirements, Constraints, verification techniques, Use case validation.

1. INTRODUCTION

Correctness in application perspective can be defined as the adherence to the specifications that regulate how users can interact with the application and how the application should behave when it is used appropriately. Requirement correctness is the bottom part of the web application. This research paper discusses the correctness in application that is appropriate for the interest of the web application. [1, 2]

2. PROBLEM

In reality, there are a number of problems with this correctness in application. These problems can cause suspensions and errors in the rest of the application. This research discusses some of the more common problems. These problems appear if requirement is not correct. [3]

They often expose that the major cause of application failures is poorly defined requirements. Ambiguous requirements lead to confusion, wasted effort and rework. A constraint is something that plays the part of a physical, social or financial restriction in an application if constraints are not well proposed application suffers failure.

2.1 Poor requirements quality

This problem arises because many requirements engineers who are inadequately trained, have inadequate access to stakeholders and other sources of the requirements, and who are given inadequate resources or authority to properly engineer the requirements. [4]

Example: Online Order rejections shall be less than 99% and the website shall be auto maintainable.

2.2 Ambiguous requirements

These requirements are not cohesive, incomplete, inconsistent, incorrect, out-of-date, specified using technical jargon rather than the terminology of the user or business/application domain. [5]

Example: The website shall be able to provide patient tumor data for the past five calendar years.

2.3 Inappropriate constraints

Many requirements are not actually mandatory. Instead, too many of them are architecture, design, implementation, and installation/configuration constraints that are unnecessarily specified as requirements. [6]

Example: Website only opens on the desktop computer but inappropriate constraint defines that it is available on mobile.

The problem discussed above have some solution techniques if documentation is correct that the above problem not occurred and we have some verification techniques for using these we take correct requirements. Determining if a goal will be completely achieved is a function of confirming that it will be completely enabled. This requires us to review the requirements that have been written to support the goal.

3. STRATEGIES

Some solution strategies use for the solving problem discuss in the research paper these solution in very prop rate for the problem handling if requirement in not correct.

3.1 Documentation

Documentation is very important factor of any project. It requires proper requirement gathering and understanding of what client wants. Proper documentation aids to reduce the poor quality of requirement gathered and ambiguity in requirement. [10] In documentation process, the elicitation is key technique which is used to recorded user requirement according to clients need. As the correctness of documentation is concerned it is recognized that after implementation the maintenance is done according to the documentation. [7, 11]

3.2 Verification Techniques

All requirements should contribute to achieving our goal [12]. If a requirement does not help us achieve the goal that it supports (in a structured requirements framework), it is not a correct requirement. Either the requirement is placed under the wrong goal, or it truly doesn't belong. In verification techniques use four fundamental methods Inspection, Demonstration, Test and Analysis in requirement correctness. [8, 13, 14]

3.3 Use Cases of Completeness Validation

Requirement can be accomplished by reviewing the use cases that are designed to enable it [15]. If the use cases are necessary and sufficient, then the requirement is correct. Each requirement is supported by one or more use cases that way it is easy to method for the correctness in requirement. [9]

4. BENCHMARK

Table 1 shows that poor requirement quality problem can be solve by proper documentation through requirement elicitation. Another solution for this problem is through verification techniques verifying the goal is achievable. The next problem discussed is ambiguous requirements. This is solved by proper documentation strategy by recording all the requirements gathered and even those which have been replaced and edited. Verification techniques also solve the same problem by inspection and demonstration. The last constraint which can be solved through a verification technique called test analysis as well as another strategy called the use case of competence validation technique called reviewing.

	Documentation	Verification Techniques	The use case of competences Validation
Poor Requirement Quality	Requirement elicitation [10]	Achieving the goal [12]	
Ambiguous Requirements	Record requirement [11]	Inspection Demonstration[13]	
Inappropriate Constraints		Test Analysis[14]	Reviewing [15]

Table: 1 *Problem and Strategies in Requirement Correctness*

5. CONCLUSION

In this paper we provided a theoretical underpinning for the pragmatic view of correctness, thus introducing more rigors into the process of requirements evolution. In detail, we have described which kind of solution techniques must be carried out at each problem during the evolution of the requirements. We have also proposed various ways in which our solution can be applied to real-life circumstances, both for validation purposes and as a supporting technique during requirements negotiation and prioritization. Furthermore, we hope that this work will bring to the attention of requirements engineers the importance of considering the three solutions we provide every requirement engineer follow these three steps in requirements correctness

Reference

1. Software Quality Engineering Testing, Quality Assurance, and Quantifiable Improvement.2005 edition-5 Jeff tian.
2. www.tutorialspoint.com/software_testing_dictionary/correctness.html
3. Software Constraints for Large Application Systems DAG I. K. SJØBERG1, RAY WELLAND2 AND MALCOLM P. ATKINSON2
4. Grant, Delvin. "Shortening requirements engineering: Dual imperative action research study." *The Journal of Computer Information Systems* 53.2 (2012): 1-8.
5. Wang, Yue, et al. "Automatic detection of ambiguous terminology for software requirements." *Natural Lanquage Processing and Information Systems*. Springer Berlin Heidelberg, 2013. 25-37.
6. Li, Yang, et al. "Requirements engineering for scientific computing: a model-based approach." *e-Science Workshops (eScienceW), 2011 IEEE Seventh International Conference on*. IEEE, 2011.
7. Génova, Gonzalo, et al. "A framework to measure and improve the quality of textual requirements." *Requirements Engineering* 18.1 (2013): 25-41.
8. http://tynerblain.com/blog/2006/07/06/requirement-completeness-validation-with-use-cases/
9. Machniak, Martin J. *Development of a quality management metric (QMM) measuring software program management quality*. Diss. Monterey, California: Naval Postgraduate School, 1999.
10. Lethbridge, Timothy C., Janice Singer, and Andrew Forward. "How software engineers use documentation: The state of the practice." *Software, IEEE*20.6 (2003): 35-39.
11. Detecting Ambiguities in Requirements Documents Using *www.cas.mcmaster.ca/wise/wise01/KamstiesBerryPaech.pdf*
12. Balci, Osman. "Verification, validation, and certification of modeling and simulation applications: verification, validation, and certification of modeling and simulation applications." *Proceedings of the 35th conference on winter simulation: driving innovation*. Winter Simulation Conference, 2003.
13. tynerblain.com/blog/2006/07/06/requirement-completeness-validation-with-use-cases/
14. tynerblain.com/blog/2006/07/06/requirement-completeness-validation-with-use-cases/
15. Verification, Validation and Testing in Software Engineering edited by Aristides Dasso, Ana Funes