Requirement Defect Mitigation Technique: An Early Stage Implementation

[1]Sandeep Kumar Nayak, [2]Raees Ahmad Khan and [3]Md. Rizwan Beg
[1] Department of Computer Science and Application, Integral University
Lucknow, Uttar Pradesh-226026, India

[2] Department of Information Technology, Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow, Uttar Pradesh-226025, India

[3] Department of Computer Science and Engineering, Integral University
Lucknow, Uttar Pradesh-226026, India

Abstract: Requirement defect identification and mitigation at early stage of Software Development Life Cycle (SDLC) is very cost effective than to later stages. The requirements analysis in requirement engineering process is critical and major foundation of requirement defect identification. A poor requirement analysis process may lead to software requirement specification (SRS) full of defects akin to misplaced, ambiguous, incompatible, misinterpreted, and incomplete requirements. In this paper, requirement defects are being identified and properly mitigated as per its severity in the requirement phase to get rid of major rework by spending extra cost and effort at the later stages. Here, a Defect Mitigation Technique (DMT) is proposed for mitigating the identified requirement defect and also the reliability of requirement is being assessed to deliver Reliable Requirement. The proposed algorithm is helping the DMT for its proper processing, defect mitigation and reliability assessment. The prime motive of this study is an effort to put off requirements stage defects from entering into later stages of SDLC.

Keywords: Software Development Life Cycle (SDLC)Requirement Defect, Software Requirement Specification (SRS), Defect Mitigation, Defect Mitigation Technique (DMT), Reliability assessment.

INTRODUCTION

Literature survey reveals that there is no such Framework is exist which may confirm reliability of Requirement document at early stage that means in Requirement Phase of Software Development Process. There is no such facility is available which may provide almost defect free requirement in the form of Reliable Requirement Specification document. Software defects are the basic reason for malfunctioning and software failure which imposes a direct impact on software reliability [Gong et al, 2003]. So the defects must take care of from the starting point of software development process. In the past decade of research software reliability engineering activities has became the need of overall process of software development life cycle [Hang et al, 2002]. Defect identification and mitigation according to their severity had become the essential need of software development to facilitate the software industries. Software defect cause analysis and preventive measure has become the essential criterion [Adeel et al, 2005] [Li et al, 2006] [Bean et al, 2008] for reliable software delivery.

PROPOSED DEFECT MITIGATION TECHNIQUE (DMT)

There are few techniques involved in mitigating the defects in developing software at early stage of requirement phase. A concrete and cost effective technique for defect mitigation in early stage is highly essential in modern era of software application and implications.

Corresponding Author: Sandeep Kumar Nayak, Assistant Professor, Department of Computer Science and Application, Integral University, Lucknow – 226 026, UP, INDIA. Contact No. +91-522-2890730 (O); +91-9935513331 (M).

Condition/Defect Stub		Condition/Defect Entry						
		Rules						
IF (Condition/ Defect)	Requirement Defect found?	Yes						No
	Defect in Requirement Scope?	Y	-	-	-	-	-	-
	Defect in Input Section?	-	Y	-	-	-	-	-
	Defect in Output Section?	-	-	Y	-	-	-	-
	Defect in Requirement Boundary?	-	-	-	Y	-	-	-
	Defect in Functional Component?	-	-	-	-	Y	-	-
	Defect in other section?	-	-	-	-	-	Y	
	No Defect found?	-	-	-	-	-	-	Y
THEN (Action/Mitigation)	Defect Mitigation through Variables							
	^ Refer to Decision Table 1	X	-	-	-	-	-	-
	^ Refer to Decision Table 2	-	X	-	-	-	-	-
	^ Refer to Decision Table 3	-	-	X	-	-	-	-
	^ Refer to Decision Table 4	-	-	-	X	-	-	-
	^ Refer to Decision Table 5	-	-	-	-	X	-	-
	Re-Inspection is required	-	-	-	-	-	X	-
	No Action is required	-	-	-	-	-	-	X

Condition /Defect Stub	Condition/Defect Entry			
	Rules			
	Yes			No
	Y	-	-	-
	-	-	Y	Y
	X	-	-	-
	-	-	X	X

Condition /Defect Stub	Condition/Defect Entry			
	Rules			
	Yes			No
	Y	-	-	-
	-	-	Y	Y
	X	-	-	-
	-	-	X	X

Condition /Defect Stub	Condition/Defect Entry			
	Rules			
	Yes			No
	Y	-	-	-
	-	-	Y	Y
	X	-	-	-
	-	-	X	X

Condition /Defect Stub	Condition/Defect Entry			
	Rules			
	Yes			No
	Y	-	-	-
	-	-	Y	Y
	X	-	-	-
	-	-	X	X

Condition /Defect Stub	Condition/Defect Entry			
	Rules			
	Yes			No
	Y	-	-	-
	-	-	Y	Y
	X	-	-	-
	-	-	X	X

Proposed Defect Mitigation Technique (DMT)
Fig 1

Defect Collection Pool

Refer by Master Table

Defect Data Dictionary

Condition/Defect Stub		Condition/Defect Entry						
		Rules						
IF (Condition/ Defect)	Requirement Defect found?	Yes						No
	Defect in Requirement Scope?	Y	-	-	-	-	-	-
	Incorrect portrayal for Product?	-	Y	-	-	-	-	-
	Business Case, behavior or Mission not defined?	-	-	Y	-	-	-	-
	Unclear needs, goals, objectives?	-	-	-	Y	-	-	-
	Operational Concepts & interface description missing?	-	-	-	-	Y	-	-
	Defect in other section?	-	-	-	-	-	Y	
	No Defect found?	-	-	-	-	-	-	Y
THEN (Action/Mitigation)	Defect Mitigation through Variables	X						
	Provide required Operational activity & interfaces	-	-	-	-	X	-	-
	Inject clear & complete product needs, goals, objectives	-	-	-	X	-	-	-
	Redefinition of Product details	-	X	-	-	-	-	-
	Correction in definition of product behavior, mission, business cases	-	-	X	-	-	-	-
	Re-Inspection is required	-	-	-	-	-	X	-
	No Action is required	-	-	-	-	-	-	X

Specimen of Requirement Scope Defect Mitigation
Fig 2

Here, an initiative step is taken to mitigate the requirement defect at the same phase with the help of Decision Table and Defect Data Dictionary. Defect Mitigation Technique (DMT) is implemented in the requirement phase so that maximum of the defect might be resolve as soon as possible for preventing them to carry forward in the further phases. DMT has a master decision table along with five other individual decision table attached through their specific references. The mitigation algorithm may be competent to fix up the defects for delivering significant reliable requirement for the further phases of system development process.

Definitions of Algorithm

Definition 1:

For a Decision Table DT = {R, C, D, V, *f* } where,

R: non-empty set of requirements $\{r_1, r_2, r_3, \text{-----------}, r_n\}$

C: non-empty set of requirement defect in form of conditional attributes, where

$$C = \bigcup_{i=1}^{m=5} C_i$$

D: non-empty set of requirement defect mitigation variables in Defect Data Dictionary in form of Decision attributes, where

$$D = \bigcup_{i=1}^{m=5} D_i$$

V: $V = V_t$ for $\forall\, t \in C$ and V_t is known as value range of requirement defect attribute

f: such that, $f : C \times D \rightarrow V$ is an action information function, which shows the action information value array for each defect attribute of each object, where

If $x \in C$ and $a \in D$ then $f(x, a) \in V$

Definition 2: Given a Decision Table DT = {R, C, D, V, *f* }, let $DF = (m_{ij})_{nxn}$ be a parse matrix within the decision table, where elements are defined as follows:-

$$m_{ij} = \begin{cases} \{a_k \mid a_k \in C \\ \Phi \text{ else} \end{cases} \quad \text{Where } i = j$$

Definition 3: Given a Decision Table DT = {R, C, D, V, *f* }, let $(m'_{ij})_{nxn}$ be a matrix w.r.t. Mitigation Variables (*Mit*) within the decision table, where elements are defined as follows:-

$$m'_{xy} = \begin{cases} \{d_m \mid d_m \in D \\ \Phi \text{ else} \end{cases} \quad \text{Where, } x = i \text{ of } m_{ij}$$

Definition 4: Initialization of count = 0.

$\forall\, d_k \in Mit$

Count = Count + number of d_k

Algorithm: An algorithm for proficient requirement defect mitigation through mitigation variables based on Decision table.

Input: Decision Table DT = {R, C, D, V, *f* } where, domain for requirement is

$R = \{r_1, r_2, r_3, \text{-----------}, r_n\}$,
$C = \{C_1, C_2, C_3, C_4, C_5\}$ is the conditional attributes,
$D = \{D_1, D_2, D_3, D_4, D_5\}$ is the decision attributes,

Output: Requirement Domain R without any defect or Defect Free Requirement R known as Reliable Requirement Specification.

Begin:

Step 1: Let the Requirement Defect Matrix DF = Φ and Mitigation Variable Matrix = Φ;

Step 2: According to Definition 1, compute entries of decision table as $R \rightarrow C, D$;

Step 3: for, $\forall\, C_n \in C$ and For n = 1; $n \leq 5$; n++;
Step 3.1: for, $\forall\, r_i, r_j \in R$
And if $f(r_i, a) \in C$ (by Definition 1)
Then for $\forall\, a_k \in C_n$,
$m(a_k) \rightarrow m(r_i)$ (by Definition 2)
Step 3.2: for matrix DF: $\forall\, a_k \in C_n$ replaces $r_i \in R$ so,
for matrix m'_{xy} w.r.t. *Mit* $\exists\, d_l \in D_n$ (by Definition 3)
Step 3.3: if $m_{ij} = \{a_k \mid a_k \in C_n$
$\exists\, m'_{yj} = \{d_l \mid d_l \in D_n$
Then fix matrix element of m_{ij} in DF

Step 4: for each attributes C in defect matrix DF, computing the count of each mitigation attribute, as
$\forall\, C_n \in C$ and $\forall\, D_n \in D$,
For n = 1; $n \leq 5$; n++;
Step 4.1: count the attributes of D_n in matrix *Mit* satisfying the maximal number of C_n in matrix DF;
count = count + $(\text{count})_n$

Note: Count = $\text{Count}_1 + \text{Count}_2 + \text{Count}_3 + \text{Count}_4 + \text{Count}_5$ will give the total number of mitigated requirement defect

Step 5: if matrix *Mit* is not empty, then turn to step 3;
Else output is achieved.

Step 6: According to Reliable Requirement Specification i.e. *Mit*, the maximum possible set of defects are mitigated.

End

DMT Activities: To achieve the better performance, Defect Mitigation Technique executes some of the vital activities are as follows:-

- It contains a master decision table which gives the reference of other five decision tables depending upon the type of requirement defect (as mentioned in fig 2).
- Decision Table contains two quadrants of conditions, one is for Requirement Defect and one is for Action Strategy whereas two other quadrants have their respective entries depending upon the rule satisfaction.
- Condition Stub (Defect Stub): In the first quadrant statement introduces one or more conditions in the form of requirement defects which comes from the defect collection pool. These defects may be treated as the factor for taking decisions.
- Condition Entry (Defect Entry): In the second quadrant of decision table condition entries are meant for completing the condition statement (defect statement). The entries may be “Yes”, “No” or “don’t care” depending upon the defect rules.
- Action Stub (Solution Stub): In the third quadrant statement introduces one or more mitigation variables from Defect Data Dictionary in the form of action strategy for requirement defects (Defect Stub). These action strategies may be treated as the steps to be taken when a certain condition of conditions exists.
- Action Entry (Solution Entry): In the fourth quadrant of decision table action entries are meant for completing the action strategy statement. The entries may be “Yes”, “No” or “don’t care” depending upon the action strategy rules.
- An individual requirement defect may be mitigated as per their matching mitigation variable found in the action stub under any classified requirement (Requirement Scope, Input Section, Output Section, Requirement Boundary and Functional Requirement; as mentioned in Fig 2) [Nayak et al, 2011].
- Most severe requirement defect have their higher priority to be mitigated.
- Residue requirement defect may have lower severity and priority at this stage and move forward without any harm for further phases.
- Lastly the comparative reliability before and after defect mitigation will assess to deliver reliable requirement.

TOOL IMPLEMENTATION

A rigorous analysis of RRS framework [Nayak et al, 2012], D3 tool and DMT is being performed with six live projects for verification and validation of this research. The critical analysis and assessment have been executed through the proposed technique of this study and the resultants of data analysis of these projects are mentioned in the Table 1, Graph1, Graph2 and Graph3.

CONCLUSION

The proposed framework for Reliable Requirement Specification (RRS: D3 Tool, Defect Data Dictionary, Defect Mitigation Technique) has been implemented and validated with the help of *Live Projects* developed by UG and PG level professions in different renowned software development organization. The consequences of the framework through implementation and validation give an optimistic gesture to identification and mitigation of requirement defect at the same phase which may help to convey a reliable requirement for the further phases of Software Development Process. Therefore this research gives the impression to put into practice at large scale of industrial data for the betterment of software development process. The future direction of this research may comprise:

- To implement RRS framework on projects having large scale of industrial data.
- To analyze the improvement through implementation of this framework at large scale of industrial data.
- To analyze the affected parameters though using this approach.

REFERENCES

Gong Y.Z., “On software’s defects,” Journal of Academy of Armored For Engineering, Vol.17, No.1, pp.60-63. 2003.

Hang X.Z., Software Reliability, Safety and Quality Assurance. Beijing: Electronics Industry Press, 2002.

Adeel K., S. Ahmad, and S. Akhtar, “Defect prevention techniques and its usage in requirements gathering - industry practices,” IEEE Engineering Sciences and Technology, Vol.8, pp.1-5, 2005.

Li M., He X.Y., and Sontakke A., “Defect prevention: a general framework and its application,” IEEE Quality Software, Vol.10, pp.281-286, 2006.

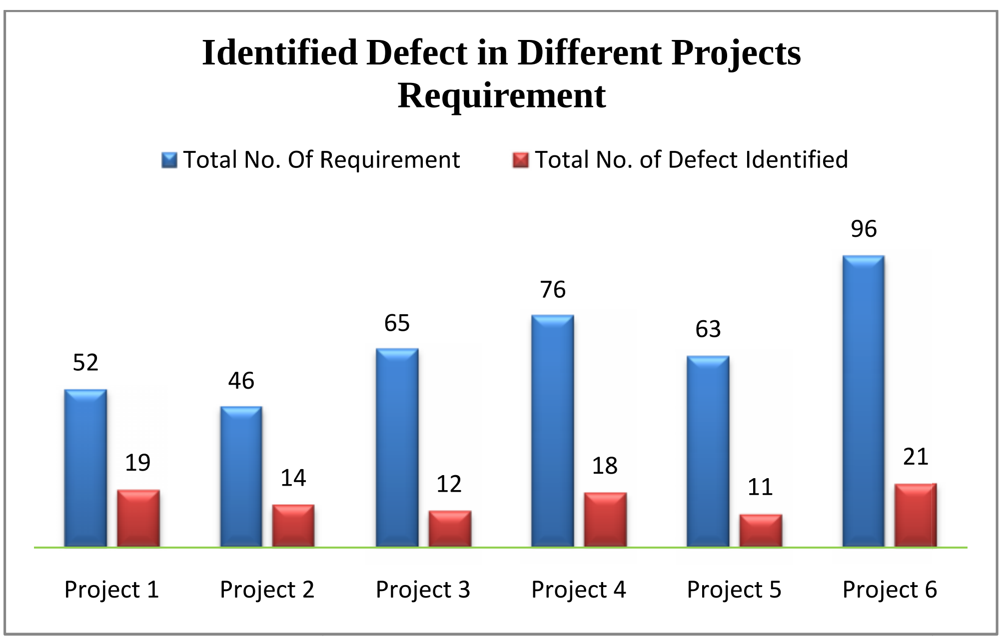
Bean E., “Defect prevention and detection in software for automated test equipment,” IEEE Instrumentation & Measurement Magazine, Vol.8, pp.16-23, 2008.

Nayak S.K., Khan R.A., Beg R.,”Requirement Defect Identification and Their Mitigation through Severity and Priority”, International Conference on Communication and Electronics Information. Thomson ISI. PP: 427-431, 2012

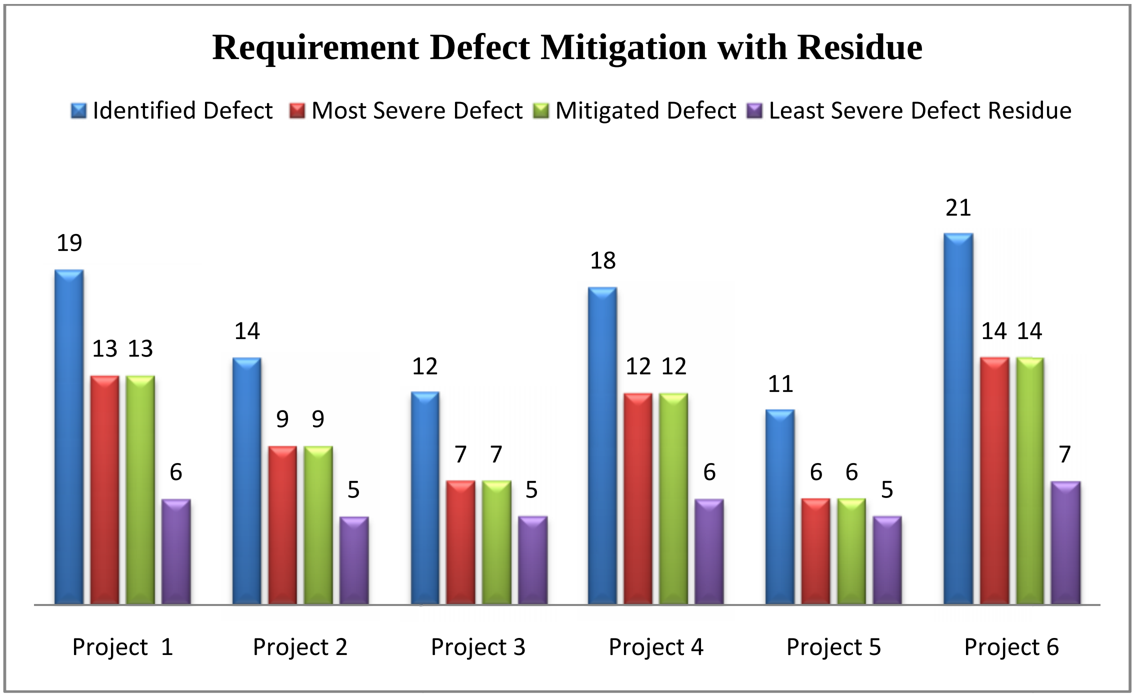
Nayak S.K., Khan R.A., Beg R.," Reliable Requirement Specification: Defect Analysis Perspective", 4th International Conference on Recent Trends in Computing, Communication and Information Technologies Springer – LNCS in CCIS. pp: 740-751, 2011

Table 1

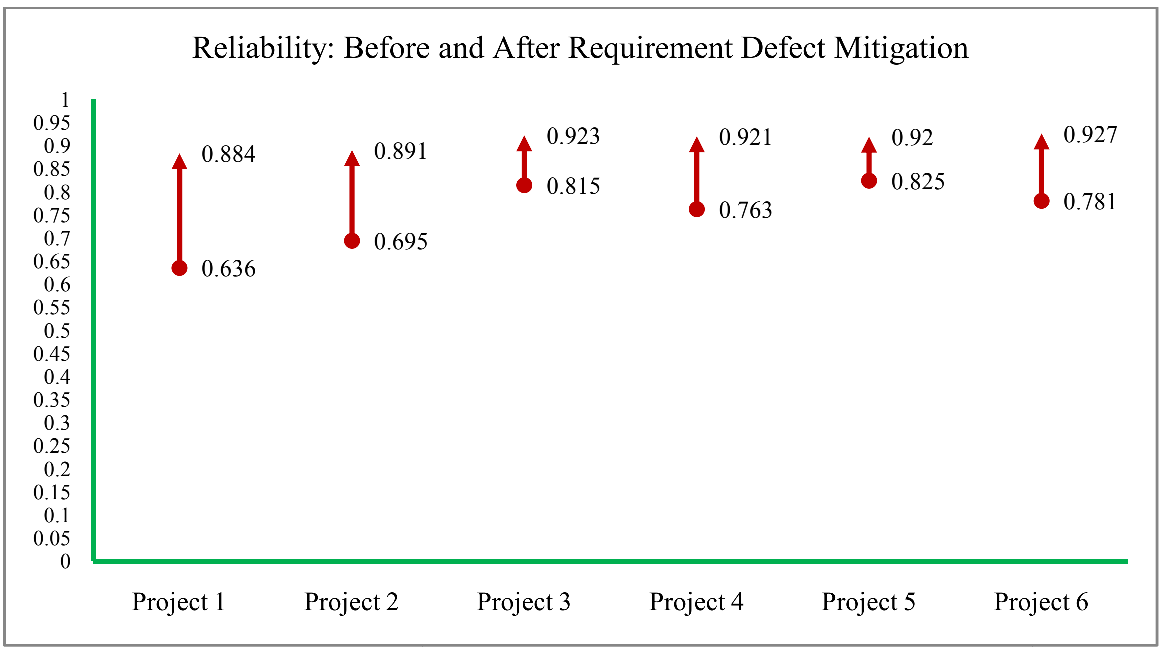
Project	No. of Requirement Gathered	Identified Requirement Defect	Reliability before Defect Mitigation	No. of Mitigated Defect (1st Pass)	Reliability after Defect Mitigation (1st Pass)	No of Non Mitigated Defect (1st Pass)	No. of Mitigated Defect (2nd Pass)	Reliability after Defect Mitigation (2nd Pass)	No of Non Mitigated Defect (2nd Pass)	Increase in Reliability
P1	52	19	**0.636**	10	0.826	09	03	0.884	06	24.8%
P2	46	14	**0.695**	09	0.891	05	-	-	-	19.6%
P3	65	12	**0.815**	05	0.892	07	02	0.923	05	10.8%
P4	76	18	**0.763**	08	0.868	10	04	0.921	06	15.8%
P5	63	11	**0.825**	06	0.920	05	-	-	-	9.5%
P6	96	21	**0.781**	11	0.895	10	03	0.927	07	14.6%



Graph 1



Graph 2



Graph 3