Reliability Assessment in Functioning of Requirement Defect Mitigation

Sandeep Kumar Nayak, Raees Ahmad Khan, and Mohd. Rizwan Beg

***Abstract*—There are few techniques involved in mitigating defects in developing software at early stages. A concrete and cost effective technique for defect mitigation in early stage is highly indispensable in modern era of software application and implications. In this paper, we are attentive over an effort to initiate a requirement defect mitigation algorithm including three phases working step by step under mitigation process. In the mitigation process flow, the mitigation algorithm may be competent to fix up the defects for delivering significant reliable requirement for the further phases of system development process. The implementation of defect mitigation process would also be appreciated by industry, software developers and innovators in future.**

***Index Terms*—Requirement mitigation process, mitigation algorithm, reliable requirement specification, requirement defect.**

I. Introduction

Decades have been passed for attempting to improve the reliability of the developed software by many researchers but the emergence of defects is still a big problem. Therefore it is necessary to introduce the best practices for identifying defects and their proper mitigation to deliver the reliable requirement for making reliable software. Defect identification and mitigation are limited to risk assessment on ad hoc basis [1], [2]. Some researchers entertain only frequently raised risks and provide a way to handle them. Researchers are interested in unleashing all the handling and avoidance mechanisms for software risks only [3]. Some study narrates only the effect of different patterns of requirements discovery on a software project [4]. In our previous study, free wheel processing assembly consists rotational free wheel structure (Fig. 1) named Defect Mitigation which takes the combined output of Requirement Defect and Severity & Priority to apply mitigation variables for mitigating the defect so that at the outer end a Reliable Requirement Specification (RRS) may deliver [5], [6].

In this paper we are giving an algorithm for mitigating the requirement defect as it identified by inspection technique [6].

Manuscript received May 1, 2012; revised August 14, 2012.

Sandeep Kumar Nayak is with Dept. of Computer Science & Application, Integral University, Lucknow-UP, India-226026 (e-mail: nayak.kr.sandeep@gmail.com).

Raees Ahmad Khan is with Dept. of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow-UP, India-226025.

Mohd. Rizwan Beg is with Dept. of Computer Science & Engineering, Integral University, Lucknow-UP, India-226026.

II. Defect Mitigation Process

This study is the extension of our previous work [5, 6] in which free wheel structure for Defect Mitigation has been mentioned for mitigating the requirement defects. Here mitigation process is executed through some steps as mentioned in Fig. 1. for delivering the Reliable requirement Specification.

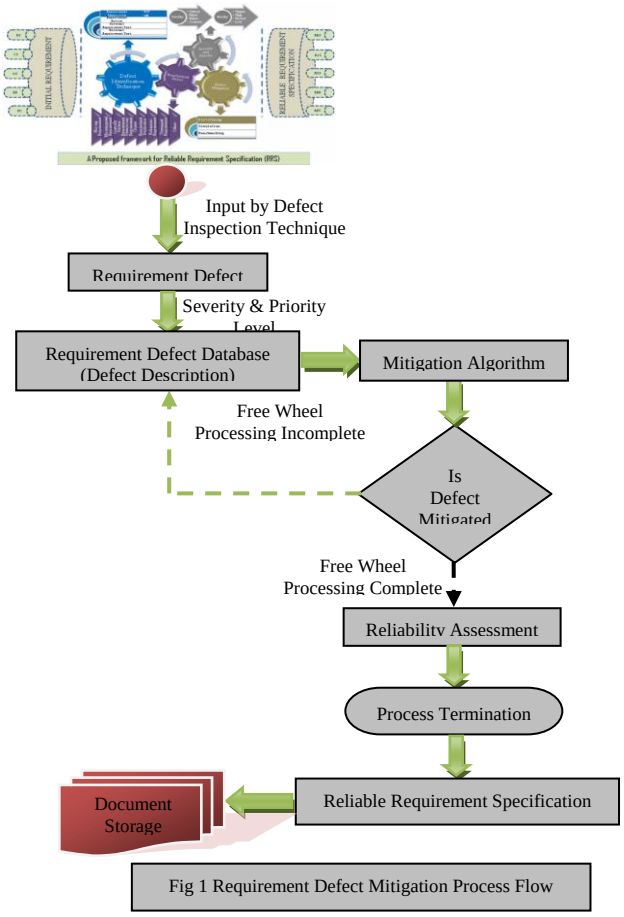

Fig. 1. Delivering the reliable requirement specification.

A. Requirement Defect

Whenever the initial requirement enters into the Free wheel processing assembly, entertains by Inspection Technique to deliver the requirement defects then the severity and priority level be assign to each and every requirement defect as TABLE I:

TABLE I: Severity and Priority Level and Their Combination

Severity	Critical (S1)	Major (S2)	Minor (S3)	Cosmetic (S4)
Priority	Urgent (P1)	High (P2)	Modest (P3)	Low (P4)
Severity & Priority	S1 + P1	S1 + P4	S2 + P2	S2 + P1	S3 + P3	S4 + P4

B. Requirement Defect Description

Requirement defects must be contained in a tabular form within the database which may follow a template of specific attributes, (TABLE II**)** such as:

TABLE II: DEFECT TEMPLATE WITH SPECIFIC ATTRIBUTES

Defect Position	Defect Indicator	Defect Cause	Defect Mitigation Cost	Defect Severity	Defect Priority	Defect Definition
Function Requirement	Functional Actor Missing	Inadequate Requirement Collection	Effort & time taken in defect identification & Mitigation	Critical (S1)	Urgent (P1)	a) Actor executes the operational Task. b) Mitigating action required immediately.

Above mentioned specimen in the template is given only for one type of defect within functional requirement domain. Here, stored defect is assigned severity and priority for further mitigating action.

C. Mitigation Algorithm

In our previous study we identified many of the requirement type in the requirement document but due to specific elaboration here only five vital requirement types and respective defects is taken as in fig 2. The mitigation variables with respect to requirement defects are taken for fixing the defects.

```
// A[m] =   1   2   3   4   -   -   -   m
            20  25  30  35  40  45  50  55

// B[n] =   1   2   3   4   -   -   -   n
            20  30  25  35  50  45  40  55

  d (i j) =
//No. of Defects in RS, IS, OS, RB, FC:          1------p
//No. of Defects in RSM, ISM, OSM, RBM, FCM:  1------q
  A (i) ← 20
     For (int i=2; i <= m; i++)
     {A (i) ← A (i-1) + 5 ;}
         For (int j=1; j <= n; j++)
          {B (i) ← dm (j) ;}
// Phase 1
      For (int i=1; i <= m; i++)
          {For (int j=1; j <= n; j++)
            {If (A (i) = B (j))
              Then
           {Start matching A (i)
                  Break ;}}}
// Phase 2
           Matching A (i)
          {For (int i=1; i <= p; i++)
              {For (int j=1; j <= q; j++)
                 {If (d (A (i)) = dm (A (j)))
                   Then
                     {Mitigate d (i);
           }            Break ;}}}
// Phase 3
          Mitigate (d, i)
            {If (priority = P1 or P2)
                  Fix now;
          Else
               If (priority = P3)
                   Fix later;
                   {No Action ;}
```

Structure as an array A (m) & B (n) respectively. This algorithm is working in three phases, Phase 1: This will match the requirement defects and respective mitigation variables under requirement type (such as: RS matches with RSM). Phase 2: After phase 1individual defect will search its respective mitigation variable under specific requirement type. Phase 3: In this phase mitigation variables may perform three tasks (fix up, fix later or no action).

D. Reliability Assessment and Process Termination

As per the definition of reliability,

$$R(t) = 1 - F(t);$$

F (t) is the probability of function failure during time t. [7], [8].

Similarly, the number of mitigated defects (MD) may be assessing through the number of non mitigated defects so the percentage also.

The Significance Level of mitigated defect may also be drawn with the help of below given two equations (1) & (2).

$$MD = N - NMD \quad (1)$$

$$\text{Significance Level} = MD / N \quad (2)$$

MD: Number of Mitigated Defects
N: Total Number of Identified Defects
NMD: Number of Non Mitigated Defects

After the assessment of significance level of overall requirement document the mitigating action as well free wheel assembly process will stop working.

E. Reliable Requirement Specification

The mitigation process in this study has limitations to evaluate whole requirement defect. Several requirements are bounded for self efficacy completely those are depends on severity and priority of respective defect. In this reference, the mitigation algorithm may play a vital role to examine the requirement defect as well as for implementation for mitigation and so forth the Reliable Requirement Specification may deliver. This reliable requirement document may contain in specified storage place.

III. CONCLUSION AND FUTURE WORK

Requirement defect mitigation and its Reliability assessment against quality software entails recording of defects within the Requirement Defect database and assigns their Severity & Priority level. Therefore this study exposed the process to analyze requirement defects and their mitigation through comparison of defects and their equivalent mitigation variables through mitigation algorithm.

Through mitigation algorithm half of the overall defects may be mitigated in the first pass and in second pass half of the rest part and if third pass is needed then few of the residue defects will be mitigated. Our future work is to implement and validate our framework to RRS and this mitigation process with algorithm so as to show its usefulness in the software industry for developing reliable software.

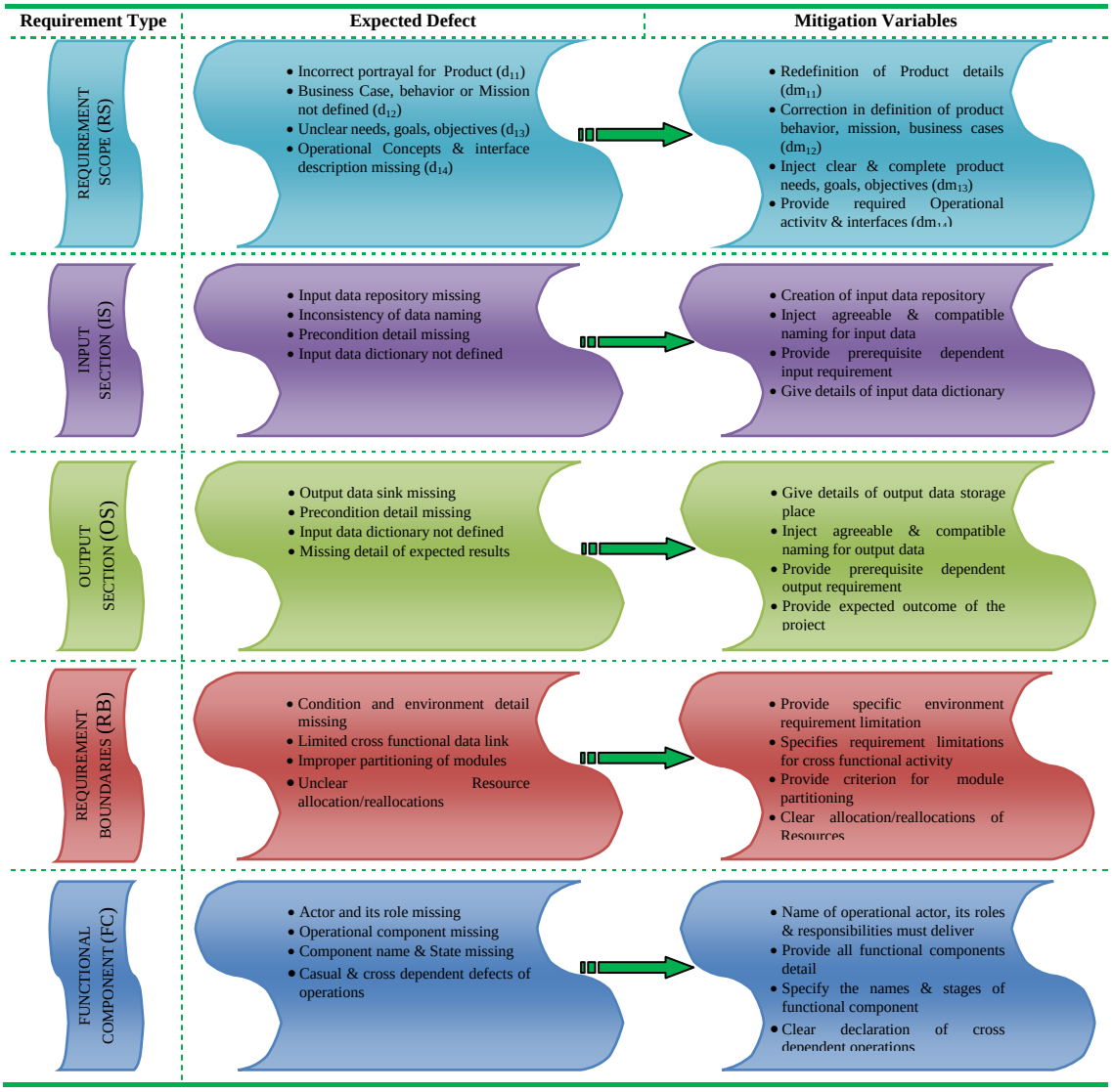

Fig. 2. Expected requirement defect and mitigation variables

REFERENCES

[1] E. Hossain, M. A. Babar, H. Y. Paik, "Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework," *16th Asia-Pacific Software Engineering Conference. IEEE*, pp. 457-464, 2009.

[2] T. J. Wilmering, "Careful What You Wish For – Risk Mitigation for Unplanned Behaviors in Distributed Diagnostic Software Systems," *IEEE*; 137-146, 2007.

[3] B. Shahzad and A. S. Al-mudimigh, "Risk Identification, Mitigation and Avoidance Model for Handling Software Risk," *Second IEEE International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 191-196, 2010.

[4] R. Thakurta, R. Roy, and S. Bhattacharya, "Impact of Requirements Discovery Pattern on Software Project Outcome: Preliminary Results," *42nd Hawaii International Conference on System Sciences. IEEE*, pp. 01-06, 2009.

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

[6] S. K. Nayak, R. A. Khan, R. Beg, "Requirement Defect Identification And Their Mitigation Through Severity And Priority," *International Conference on Communication and Electronics Information.Thomson ISI*, 2012.

[7] S. Xu, "An Accurate Model of Software Reliability," *13th IEEE International symposium on Pacific Rim Dependable Computing*, pp. 77-84, 2007.

[8] S. Xu, "Reconsideration of Software Reliability Measurements," *16th IEEE Asian Test Symposium*, pp. 159-162, 2007.

Sandeep Kumar Nayak is an Assistant Professor in Department of Computer Science & Application, Integral University, Lucknow – Uttar Pradesh (INDIA) Sandeep Kumar Nayak did MCA from Uttar Pradesh Technical University & pursuing Ph.D. (Information Technology) from Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow - Uttar Pradesh (INDIA). He has published six research papers in reputed and indexed International conference proceedings. He is member of some professional bodies.

Raees A. Khan is an Associate Professor and the Head of Department of InformationTechnology, BBA University, Lucknow (INDIA) Dr. Raees A. Khan done MCA from Punjab University & Ph.D. (Computer Science) from Jamia Millia Islamia (Central University), New Delhi. He has written one book and coauthored one book, published numerous articles, several papers in the National and International Journals and conference proceedings. The area of expertise are Software Quality Assurance, Software testing, Software Security. He has written two projects on Software Security. He is a member of various professional bodies.

Prof. Md. Rizwan Beg is M.Tech & Ph.D in Computer Sc. & Engg. Presently he is working as Professor & Head Deptt. Of Computer Sc. & Engineering. and Information Technology, He is having more than 16 years of experience which includes around 14 years of teaching experience. His area of expertise is Software Engineering, Requirement Engineering, Software Quality, and Software Project Management. He has published more than 40 Research papers in International Journals & Conferences. He is member of large member of International professional societies & also member of Advisory Board for various institutions in India.