Empirical Validation of Reliable Requirement Specification Framework

[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:

In the view of changing technologies in requirement engineering the RRS framework is being introduced [1] for early stage requirement defect identification [2] and mitigation [3] during software development process to make requirement specification reliable. With the use of RRS framework at early stage of SDLC, requirement defect identification and their proper mitigation has become cheaper than to later stages. The principal purpose of this framework is to filter out requirements stage defects from entering into later stages of SDLC. In this paper, it is being tried to show the experimental design for validating RRS Framework and to give comparative analysis with petri net tool [9] which is meant for later stages of software development process. Under RRS framework two major techniques have been introduced, one for requirement defect identification [2] and other for requirement defect mitigation [3] which now being tried to validate through empirical design under pre-tryout, tryout and statistical analysis with hypothesis methodology.

Keywords: Reliable Requirement Specification (RRS), Software Development Life Cycle (SDLC)Requirement Defect, Requirement Specification, requirement defect identification, Defect Mitigation, Defect Mitigation Technique (DMT), Reliability assessment.

INTRODUCTION

Verification and validation are the major factors for the evaluation of any research work. With the help of implementation process the research work use to verify and after that verified research work has to be validated with the help of any existing or counterpart model or framework which deals with similar problem at any stage of development process may be taken for statistical analysis.

RRS framework is well designed [1] and implemented [2 and 3] to deliver almost defect free requirement in the form of reliable requirement specification document. The unmatched requirements become the software defects at later stages of software development process causes to software failure or direct impact on software reliability [4]. So the requirements must take care from the initial stage of software development process. The better treatment of requirement defect [5] at its initial phase became the essential need of software development to facilitate the software industries. Therefore, to identify causes of software defect and its respective preventive measures has become the essential criterion [6, 7 and 8] at each level of software development so that software reliability can be increase. RRS framework is conceptually as well practically helpful to handle all above mentioned problem at its level best.

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

1. THE RRS FRAMEWORK

Highly Reliable and complete requirement delivery for developing software became the most and first requisite in Software Development Process. Initial requirement defect identification and their mitigation is the key aspect in planning the development of the potential software.
In this research work it is being tried to identify potential requirement defect through defect identification technique under Defect Data Dictionary (D3) and setting defect severity as well their priority to mitigate. Here, a Framework for Reliable Requirement Specification (RRS) [1] is being introduced containing three major components 1) Input component in the form of Initial Requirement 2) Free Wheel Processing Assembly as the combination of Defect Identification Technique (D3) [2], Requirement Defect, Severity & priority and Defect Mitigation Technique (DMT) [3] 3) Output component in the form of Reliable Requirement Specification.

2. VALIDATION OF THE FRAMEWORK

In this paper, it is being tried to perform empirical validation with the help of complementary model called Petri Net for comparing the RRS framework. The RRS framework is meant for identifying and mitigating the requirement defect at the early phase or requirement phase of SDLC with its reliability assessment where the Petri Net is meant for identifying and resolving the defects at the later stage (coding and in further phases) of SDLC with the assessment of software reliability.

The validation part consists three major sections first is *Pre-tryout*: in which a small segment of requirement document is being taken to identify and mitigate the requirement defect with the help of both framework / tool and then after assess the reliability separately and compare it. Second is *Tryout*: in which a large requirement data of Electronic Fund Transfer (EFT) module for a financial sector is being taken to identify and mitigate the requirement defect with the help of both framework / tool and then after assess the reliability separately. Third is *statistical analysis*: in which the statistical analysis about both the framework / tool is executed and tried to establish and prove one of the hypothesis to better evaluate the researched framework.

2.1 Pre-Tryout

RRS framework priorily verified through implementation for showing its proper working under defect identification technique (D3) and DMT techniques. Now, a sample requirement data of faculty management system is taken for showing its working from *initial requirement* to *reliability assessment* (as per framework working style [1, 2 and 3]) which will further compare with its counterpart model Petri Nets for validation under pre-tryout process. Petri Nets are found to be powerful in modeling performance and dependability of computer and communications systems. Formally, a Petri net has 5 tuple [9].

PN = (P, T, A, M, μ0) where
P is a finite of places (circle representation)
T is a finite set of transitions (bar representation)
A is a set of arcs connecting.
M is a multiplicity associated with the arcs in A
μ0 is the marking that denotes the number of tokens for each place

The Generalized Stochastic Petri Nets model for quantitative reliability prediction for process oriented development is depicted in Fig. 1.0 which represents generalized stochastic Petri nets model [10] contains places, transitions and tokens for assessing the software at its coding phase of SDLC. Through this model the requirements of faculty management

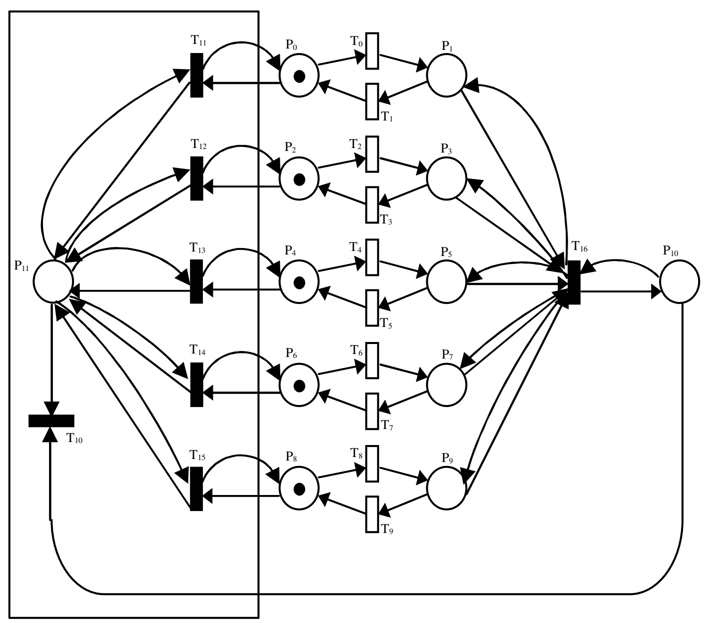

Fig. 1.0: Generalized stochastic Petri nets model
Adopted from [10]

system may be assessed but at its coding phase which comes further with respect to requirement phase of software development process. This is also helpful in estimating the reliability of software through defect management. The working of Petri Nets may notice through Table 1 containing sample data of requirements with 11 identified defects at its first appearance and lower reliability count but after the defect removal the reliability increases upto 90.7692.

Table 1: Sample Data with Petri Net

Petri net Implementation	Total Number of Requirement	Total Number of Requirement Defect	Reliability in %	Unreliability in %
Execution of P: Places, T: Transitions	**65**	**11**	**83.0769**	**16.9231**
	65	**06**	**90.7692**	**09.2308**

Whereas with the use of RRS framework same sample requirement of faculty management system is being assessed under D3 defect identification technique and DMT defect mitigation technique. After being identify the maximum possible number of requirement defect through D3 technique it can be observed that the reliability of requirement document is so much of lesser for dealing with further phases of SDLC. But as soon the DMT technique is employ on the requirement defects they decreases upto least count of severe defects and increases the reliability of requirement upto 92.3077 (Table 2). The defects which are not mitigated (but fixed for a time being) have least severity level and so the priority which are not effecting the overall requirement document. Therefore it may conclude that i) reliability assessed by Petri Net is lesser than to reliability assessed by RRS framework; ii) Petri Nets works at coding phase whereas RRS framework in requirement phase so it is much more capable to identify and mitigate requirement defects to reach maximum level of reliable requirement specifications.

Table 2: Sample Data with RRS Framework

RRS Implementation	Total Number of Requirement	Total Number of Requirement Defect	Reliability in %	Unreliability in %
Implementation of D3 Technique	**65**	**12**	**81.5385**	**18.4615**
Implementation of DMT	**65**	**05**	**92.3077**	**07.6923**

2.2 Tryout

In this section, the similar processing of validation has been followed apart from its statistical analysis. Here, a large module EFT from financial management project is taken for further assessment of the RRS framework with respect to Petri Nets model. The module has a certain number of requirements (Table 3 and Table 4) containing some of the requirement defects. These requirement defects will be entertain by both the model of Petri Nets [10] and RRS framework for removing it properly and further assessing the reliability.

Table 3: Sample Data EFT module with Petri Net [10]

Petri net Implementation	Total Number of Requirement	Total Number of Requirement Defect	Reliability in %	Unreliability in %
Execution of P: Places, T: Transitions for Defect Identification & Removal	**214**	**17**	**92.0561**	**07.9439**

Table 4: Sample Data EFT module with RRS Framework

RRS Implementation	Total Number of Requirement	Total Number of Requirement Defect	Reliability in %	Unreliability in %
Implementation of D3 Technique	**214**	**40**	**81.3084**	**18.6916**
Implementation of DMT	**214**	**15**	**92.9907**	**07.0093**

It may be observed (Table 3) that the final outcome from the Petri Net model in the form of requirement defect is 17 which represents the reliability of requirement as 92.0561. Whereas, the final outcome of RRS framework (Table 4) with the same data is 92.9907 as reliability of requirement containing 15 lowest severe defects which is more than that of Petri Net's assessment. Therefore it may conclude that RRS framework performs better at the early stage of requirement defect identification and mitigation which may also be examine through statistical analysis. The tryout validation does not reflect that Petri Net is not capable to handle the defect, but it shows that RRS framework is significantly more competent to handle requirement defect at its early stage of SDLC with respect to Petri Net.

3. STATISTICAL ANALYSIS

The statistical analysis use to start with analysis plan which describes how to use sample data to accept or reject the null hypothesis. Under this analysis plan two major things should specify, one is *Significance Level*: frequently, the most of the researchers use to choose significance level equal to 0.01, 0.05, or 0.10, but any value persists between 0 and 1 can be taken and here for this research validation 0.10 is being taken as significance level; second is *Test Method*: any statistical method may choose for assessing the hypothesis and here *chi-square test for independence* is being taken to determine whether there is a significant relationship between two elaborated frameworks. The state of hypothesis is as follows:-

H_0 (null hypothesis): The RRS framework is more reliable and valid in context with Petri net tool for assessing the reliability of software requirement at the early stage or requirement phase of SDLC.

H_a (alternative hypothesis): The RRS framework is less reliable and valid in context with Petri net tool for assessing the reliability of software requirement at the early stage or requirement phase of SDLC.

After being stated the two hypothesis and the chi-square test method there is a need of sample data as mentioned above in Table 3 and Table 4, along with this sample data some of the key parameters have to find out such as degree of freedom, expected frequencies and test statistic associated with the test statistic.

Degrees Of Freedom:

$$\mathbf{DF = (r - 1) * (c - 1)}$$

Where, r is number of rows and c is number of columns used in sample data table (Table 5).

Expected frequencies: The expected frequency value counts are computed separately for each framework. Compute r * c expected frequencies, according to the following formula.

$$\mathbf{E_{r,c} = (n_r * n_c) / n}$$

where $E_{r,c}$ is the expected frequency value count for level *r* of one framework and level *c* of another framework, n_r is the total number of sample observations at level r of one framework, n_c is the total number of sample observations at level *c* of another framework, and n is the total sample size.

Test Statistic: Since the statistical method chi-square test is taken for assessing the above mentioned hypothesis so the chi-square (X^2) will be defined as:-

$$\mathbf{X^2 = \Sigma [(O_{r,c} - E_{r,c})^2 / E_{r,c}]}$$

where $O_{r,c}$ is the observed frequency count at level *r* of one framework and level *c* of another framework, and $E_{r,c}$ is the expected frequency count at level *r* of one framework and level *c* of another framework.

The sample data:-

Table 5: Sample Data for Statistical Analysis

Frame work	Total No. of Requirement	Total No. of Requirement Defect	Reliability of Requirement	Row Total
RRS	**214**	**15**	**92.9907**	**321.9907**
Petri Net	**214**	**17**	**92.0561**	**323.0561**
Column Total	**428**	**32**	**185.0468**	**645.0468**

Analysis:-

Degree of Freedom,

$DF = (r - 1) * (c - 1) = (2-1)*(3-1) = 2$

Expected frequencies,

$E_{r,c} = (n_r * n_c) / n$

$E_{1,1} = (321.9907*428) / 645.0468 = 213.6465$
$E_{1,2} = (321.9907*32) / 645.0468 = 15.9736$
$E_{1,3} = (321.9907*185.0468) / 645.0468 = 92.3706$
$E_{2,1} = (323.0561*428) / 645.0468 = 214.3535$
$E_{2,2} = (323.0561*32) / 645.0468 = 16.0264$
$E_{2,3} = (323.0561*185.0468) / 645.0468 = 92.6762$

Chi-square,

$X^2 = \Sigma [(O_{r,c} - E_{r,c})^2 / E_{r,c}]$

$X^2 = (214 - 213.6465)^2 / 213.6465 + (15 - 15.9736)^2 / 15.9736 + (92.9907 - 92.3706)^2 / 92.3706 + (214 - 214.3535)^2 / 214.3535 + (17 - 16.0264)^2 / 16.0264 + (92.0561 - 92.6762)^2 / 92.6762$

$X^2 = 0.1281$

So, it may say that observed chi-square value (X^2) is **0.1281**

Since for this research validation 90% level of confidence is taken, which means alpha (significance level) = 0.10
With the use of standard table critical value of chi-square for the problem's degree of freedom (DF = 2) and significance level (0.10), can be taken to compare the observed chi-square value as mentioned below:-

- If observed chi-square < critical chi-square, then frameworks are not related (H_0 is true)

- If observed chi-square > critical chi-square, then frameworks are not independent (and hence may be related) (H_0 is false).
- The observed chi-square value is **0.1281** where critical value of chi-square is **4.605**

So, it may say that first axiom is acceptable as the value of observed chi-square (0.1281) is less than the critical value of chi-square (4.605). Therefore, the null hypothesis is acceptable and so forth the RRS framework is more reliable and valid in context with Petri net tool for assessing the reliability of software requirement at the early stage or requirement phase of SDLC.

4. CONCLUSION

In this paper, the hypothesis methodology with statistical analysis is used for the proper validation of RRS framework. In which a specimen requirement of faculty management system is taken for analyzing the working of defect identification technique, assigning defect weightage through severity and priority, defect mitigation technique and reliability assessment through different metrics under pre-tryout process. Secondly, a large requirement data of Electronic Fund Transfer (EFT) is taken for analyzing same working of RRS Framework and compare with the working of Petri Net tool. Under statistical analysis the hypothesis methodology is used with the help of chi-square test procedure. The validation process of RRS framework is reflecting positive and enhanced resultant in comparison with Petri Net which justifies the better performance of the newly developed framework. So, the comparative analysis of RRS framework and Petri Net tool, used in later stage of SDLC, signifies that RRS framework is applicable to make the requirement specification reliable at early stage.

REFERENCES

[1] Nayak SK, Khan RA, Beg R. “Requirement Defect Identification And Their Mitigation Through Severity And Priority” International Conference on Communication and Electronics Information. ASME Press (USA), Indexed in Thomson ISI.2012 PP: 427-431

[2] Nayak SK, Khan RA, Beg R. “Requirement Defect Identification: An Early Stage Perspective”, International Journal of Computer Science Issues, Vol. 9, issue 5, No.1, September 2012, ISSN (online): 1694-0814, Indexed in Elsevier & Ei Compendex, (www.ijcsi.org)

[3] Nayak SK, Khan RA, Beg R. “Requirement Defect Mitigation Technique: An Early Stage Implementation”, International Journal of Computer Science Issues, Vol. 9, issue 5, No.1, September 2012, ISSN (online): 1694-0814 Indexed in Elsevier & Ei Compendex, (www.ijcsi.org)

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

[5] 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

[6] 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.

[7] 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.

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

[9] Murata T., "Petri Nets: Properties, Analysis and Applications", Proc. Of IEEE, Vol. 77 No. 4, pp. 541-560, 1989.

[10] Mohan K.K., Verma A.K. and Srividya A. “Software Reliability Estimation Through Black Box and White Box Testing at Prototype Level” 2nd International Conference on Reliability, Safety & Hazard (ICRESH-201O) pp: 517-522, 2010.

Sandeep Kumar Nayak, Assistant Professor, Department of Computer Science & Application, Integral University, Lucknow – Uttar Pradesh (INDIA). He did Masters in Computer Application from Uttar Pradesh Technical University after obtaining B.Sc in Computer Science, recently submitted Ph.D. thesis in Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow - Uttar Pradesh (INDIA). He published more than ten research papers in reputed and indexed international journals and conference proceedings. He is the member of many professional bodies and also invited as session chair in an international conference organized by Elsevier & IACSIT in January, 2012.

Raees A. Khan, Associate Professor & Head, Department of Information Technology, 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 & PhD in Computer Science & Engineering. Presently he is working as Professor & Head Dept. 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.