International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - TeLMISR 2015, ISSN : 2348-2273
Available Online at www.ijeecse.com

An Empirical Validation of Risk Assessment Framework

S. K. Pandey
Department of Electronics and Information Technology
Ministry of Communications & Information Technology, Government of India, New Delhi, India
santo.panday@yahoo.co.in

Abstract: ***In this digital era, Information and Communications Technology (ICT) has become a key enabler in most of the domains of our day-to-day lives. Computer networks have transferred our life into a fast and comfortable one but at the same time, it has posed various threats to the existing information systems due to open accessibility. Any information asset, when connected to the outside world, is vulnerable to attacks. The attacks are mainly caused by threats that have the potential to exploit vulnerabilities. Any type of damage to these assets causes risk and it is one of the most important factors to the organization/s. The risk of malicious attacks to the software security has considerably gone up and to prevent such risk is very necessary. The maxim 'sooner is better' has become the order of the day. Hence, this study was undertaken in view of the significance of risk assessment in the requirements phase of SDLC. In the absence of any roadmap/process/framework, in one of our earlier papers, Risk Assessment Framework (RAF) for assessing the risk in the requirements phase itself along with validation results has already been proposed. This framework has three major components: nine security policies checklists, weight for each attribute of all the policies and quantified risk estimation. As the validation was done on only one live project at that time; it was not enough to conclude about the effectiveness of the RAF. Hence, RAF is implemented on five more live projects of different software companies. This paper discusses the step by step implementation of RAF on those projects and provides the quantitative data of various steps. Finally, the results are also discussed, which ascertains the effectiveness of the RAF up to some more extent. Such a framework may prove to be appropriate in terms of software security assurance right from the inception itself.***

Keywords: Risk Assessment, Risk Assessment Framework, Information Security, Quantitative Assessment of Risk, Empirical Study of Risk Assessment.

I. INTRODUCTION

Modern technology is at the helm of development and progress. The progress has been achieved but this has some limitations too. These limitations are posing big threats and challenges and these are required to be addressed by software experts. Some of the software are being developed and put into test to thwart and minimize the risk. It is noteworthy to mention here that the assessment tests are to be conducted in the application of security measures. Time should be managed in order to maintain accuracy and speed up the security process. Scorpion's efforts are being attempted to develop secure software but these are not sufficient and satisfactory, as it may delay security assessments. Such 'delays' may count heavily towards security and quality assurance measures [1]. It is observed that the development in respect of early and accurate security estimation needs to be undertaken for holistic developments. It is imperative to have a potentially effective approach for an early, on time and accurate assessment of risk during software development life cycle.

Traditionally, risk can be defined as the potential harm caused if a particular threat exploits a particular vulnerability to cause damage to an asset, and risk analysis is defined as the process of identifying security risks and determining their magnitude and impact on an organization [2, 3]. NIST Guide for Security Certification and Accreditation [4] elaborates the definition to explore the entire process. Risk assessment comprises of three major areas, as: (i) Identification of threats to and vulnerabilities in the system; (ii) Potential impact or magnitude of harm that a loss of CIA (Confidentiality, Integrity or Availability) would have on enterprise operations or enterprise assets, should an identified vulnerability be exploited by a threat; and (iii) The identification and analysis of security controls for the information system [5].

At present, Risk assessment is an instrumental technique for managing Information Systems Security [6]. Various information security risk assessment methods are available that can be adapted and executed by the organizations, and each has different approaches to assess and monitor the information security risks [7]. A comparative study of the major existing frameworks, COBRA, CORAS, CRAMM, OCTAVE, SOMAP, and NIST Guide, along with strengths and weaknesses of each one has already been accomplished [8]. To surpass these weaknesses and realizing the need of a risk assessment methodology particularly for requirements phase of SDLC, a new framework RAF has been proposed and described in one of our previous papers [9]. At that time, RAF was implemented on only one project; hence, it was felt that to make the RAF more reliable, its implementation on a large sample is highly required.

Hence, in this paper, detailed implementation results of RAF on five live projects have been discussed along with a brief overview of RAF. This work unfolds and provides an integrated method to determine the risk in a quantitative manner that may be presented at the requirements phase itself.

Beyond this introduction on the background details, rest of the paper is organized as follows: Section II presents a brief discussion on the RAF Process, whereas in Section III, Implementation Mechanism is discussed. In Section IV, 'Experimental Validation and Results' on the SRS of five live projects have been given and 'Comparison of Results' are given in Section V. Finally, Section VI presents 'Conclusion and Future Research Directions' in the area.

II. RAF PROCESS

A prescriptive Risk Assessment Framework (RAF) is proposed for the risk assessment in the requirements phase of SDLC. By adapting RAF, a requirement engineer can assess the risk aspects of SRS in a right perspective. RAF is a cyclic process in which a number of steps/stages are involved to reach the ultimate objective. The architecture of RAF is given in the Fig.1.

RAF is a security risk assessment framework for requirement phase. By going minutely its various stages, requirement engineers would be able to assess the risk aspects of the requirements. RAF will be operated on SRS, prepared by requirement engineers. The impetus acknowledged is on security policies and its checklists. In each security policy, various attributes are identified based on the checkpoints and then the respective weights of each one is also assigned through an estimation using expert surveys. A mathematical formulation has been proposed for the calculation of the risk. Then the tolerance level of the risk is also assessed and accordingly, suitable countermeasures/ mitigation techniques can be applied in a smooth manner. If risk is acceptable according to the time, type of project as well as resources available, then SRS could be delivered for design phase. For more details, our earlier paper may be referred [9].

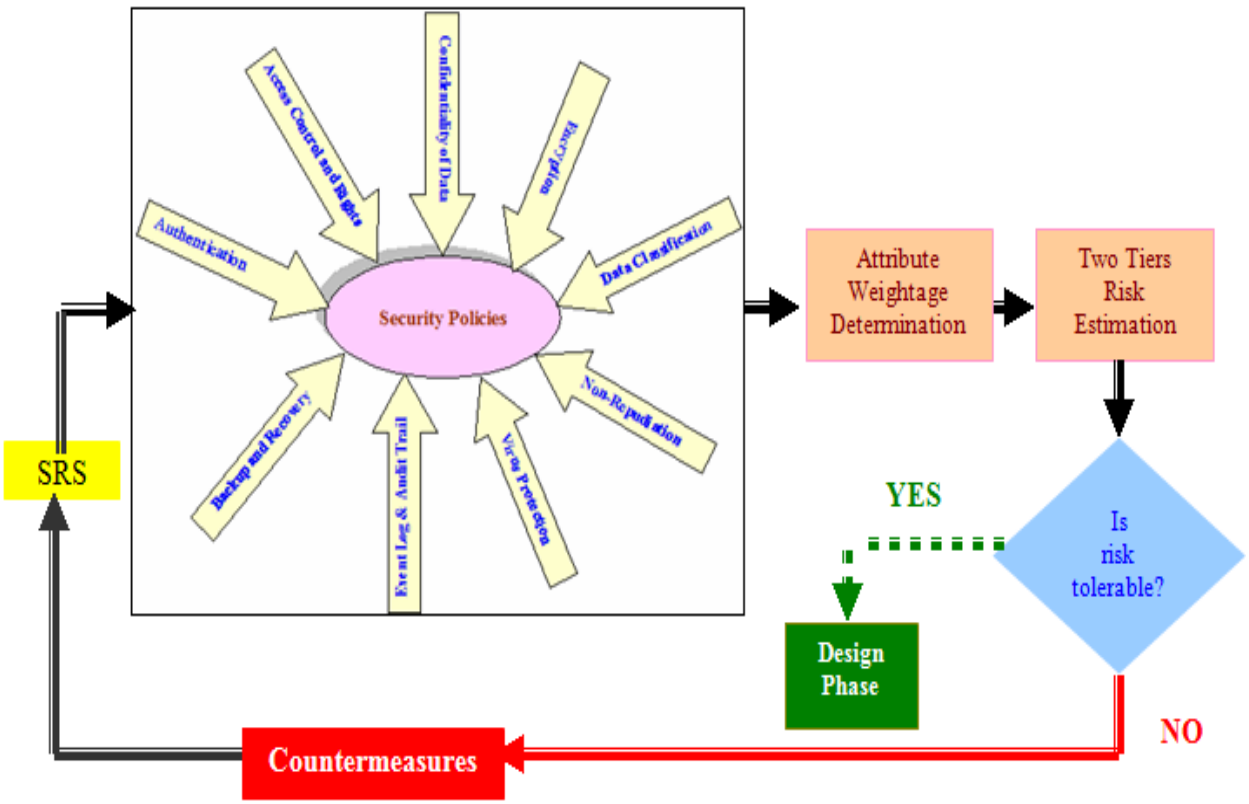

Fig. 1. Architecture of the RAF

1. Security Policies: In general terms, security policy typically describes principles or rules to guide decisions and achieve rational outcomes for assuring information Systems for organization or other entity. These are of prime importance for risk management, fraud prevention, and information teams. The security policy addresses constraints on functions and flow among them, constraints on access by external systems and adversaries including programs and access to data by people. Sound security policy architecture protects organization from attacks as well as accidental internal leakage of information, and data mishandling, whereas a poorly documented or ambiguous policies may result into a production delay and confuse the security team members which results in higher cost and effort. These nine security policies are shown in the first part of this figure. For this purpose, a checklist is proposed for each of the security policies. Due to page limitations, we are not giving the complete checklists in this paper. Some of these checklists have already been published [10, 11, 12, 13, 14]. Serial No. of checkpoints/attributes of the policies does not refer any priority of the attribute; it is used for only convenience of the presentation.

2. Determination of the weights of the attributes: After the designing of these checklists, it was felt by a team of experts that each checklist contains various attributes corresponding to each checkpoint and hence their weight for that security policy may not be the same rather it will be definitely different. Thereby, a process of estimating component weights (e.g. quantified impact) initiated through well structured expert survey by area experts/practitioners. The feedback was collected on the following issues:

- Checklists' relevance to the purpose,
- Analysis of the checklists' quality which include following heads:
    - importance of the attribute
    - Potential utility for evaluation practice
    - Completeness/coverage of attributes
    - Relevance of all the attributes

In the rightmost column of each checklist, to assign a *weight between 1 to 5 (1 is minimum and 5 is maximum)* to each attribute for the implementation of the each security policy.

These checklists along with the review form were sent to the thirty experts from the varied fields' viz. academia, industry, scientific organizations, educational institutions, research bodies, government organizations. After a comprehensive exercise, we were able to have duly filled feedback forms from the twenty experts only. After collecting these forms, feedback was compiled in two ways. At first level, based on the comments cited in the review forms, we made some revisions in the checklists/attributes and then again a fresh ranking was taken. At the second level, we designed a format in an excel sheet, in which all the data from the experts' comments were filled. Since, we received the feedback from twenty experts only; an average rank value of each attribute was calculated. Based on the average value of each attribute, we finalized the weight of the attributes of each security policy, which are discussed in one of our earlier papers in detail [9].

3 Risk Assessment: In the absence of any framework, fully dedicated to the risk assessment in the requirements phase of SDLC, the framework RAF is proposed. After determining the weight of the attributes of the security policies, we propose the two tiers risk assessment, which can be performed by using the formulas. The formulation is done by using the concept of averages which is a suitable statistical tool that may be used in these conditions. Here, two terms have been introduced: Policy Compliance Factor (PCF) and Risk Factor (RF). PCF refers to the overall compliance/adherence to policy checkpoints. RF refers to the quantified estimation of occurrence of the risk.

For the Tier -I risk assessment:

Policy [Attributes]

$PCF = \sum W_i X_i / n$

where X_i = { 1 or 0

and i = 1, 2, 3, ……n

Here, W_i is the weight of the attribute, and X_i is the value of the compliance of the checkpoint i.e. if a checkpoint is compliance, the value will be 1, and if not, its value will be 0.

For the Tier -II risk assessment:

Risk [Policy]

$RF = \sum W_i X_i / n$

where X_i = { 1 or 0

and i = 1, 2, 3, ……n

Here also, W_i is the weight (value) of the security policy which is calculated at the Tier-I, and X_i is the value of its occurrence i.e. if a security policy is applicable for a project, the value will be 1, and if not, its value will be 0. Although, we strongly recommend that all these policies

are applicable for building secure software. But, still, requirement engineers may decide it depending upon the need of the project.

4 Risk Tolerance: Based on the above calculated risk value, its tolerance limit may be decided. We propose the following limits, as given in the Fig. 2:

- *Low Risk:* SRS is at low risk if the value of the final risk value is ≥ 3.5.
- *Medium Risk:* SRS is at medium risk if the value of the risk lies between 2.5 to 3.5.
- *High Risk:* SRS is at high risk if the risk value is ≤ 2.5.

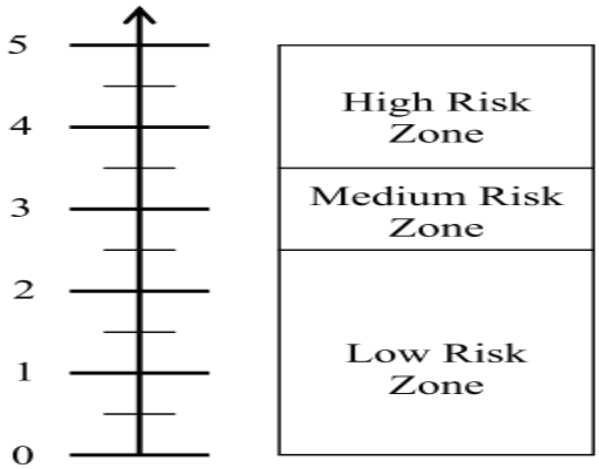

Fig. 2. Risk Zones

III. IMPLEMENTATION MECHANISM

Proposal of any framework is only useful if it is easy/handy to implement in the real life projects. We tried to make RAF user friendly from the point of implementation. Following are the guidelines/ steps for implementation of the RAF:

- SRS will be taken as input in the RAF.
- The next step as per the RAF will be structured walkthrough by checklist filtering of the SRS, in which several checklists of security policies are provided for verification of the SRS.
- If any checkpoint is not pertinent to the project, it may be identified as 'not applicable'. This will not be taken into consideration.
- For all the applicable checkpoints, requirement engineers may assess the compliance/non-compliance checkpoints.
- Accordingly, the weight of every attribute is taken for each security policy.
- Then, the two levels of the assessment may be followed, as specified in the RAF for the calculation of the risk.
- If the risk is tolerable, then the teams should handover the final SRS to the designers. Final SRS will be the output of the requirement phase of the SDLC.
- If the risk is not tolerable, then the teams should modify the SRS and repeat the steps from beginning, iteratively.

IV. EXPERIMENTAL VALIDATION AND RESULTS

The empirical study has been accomplished by the implementation of RAF on SRS of five live projects. In subsequent sections, a brief introduction of the project is given followed by the two-tier risk assessment along with evaluation of risk tolerance and discussion.

1. Tryout-I: SRS of this project provides an outline of a software product to handle the student course information process. Individuals responsible for reviewing all proposals for this software are the intended audience for this document. This may include students, faculty, administrators and any other individual who may be responsible for maintaining and upgrading the current computer system, and purchasing new systems. The software product proposed by this SRS is the Student Course Information System (SCIS).

Table I. Tryout Data (Project-I) for Risk Factor

S. N.	Policy	PCF	Compliance Status	Weighted PCF
1.	*Backup and Recovery*	0	1	0
2.	*Encryption*	1.78	1	1.78
3.	*Data Classification*	2.11	1	2.11
4.	*Non-Repudiation*	0.51	1	0.51
5.	*'Confidentiality of Data'*	1.29	1	1.29
6.	*Virus Protection*	1.87	1	1.87
7.	*Event Log and Audit Trail*	0.56	1	0.56
8.	*Access Controls and Rights*	2.10	1	2.10
9.	*Authentication*	2.23	1	2.23
				∑ WPCF = 12.45
				RF == (12.45) / 9 = 1.38

Now, the value of the risk factor (RF) is compared with the threshold values, as specified in the RAF. It can also

be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of the software. Here, the value of the RF is 1.38, which is at the high risk zone as specified in RAF; this value is not tolerable. Hence, requirement engineers should revise the SRS by incorporating the security related points. Backup and recovery related attributes must be taken into consideration because there is not any single compliance for this policy. For other policies, the attributes which are not complied and the value '0' is assigned, must be taken into consideration. The countermeasures may also be used for the same. For example, in backup policy, following countermeasure may be used: a tool for checking and reporting about daily backup, a directive for backup safety, a mechanism for periodic backup logs and review and a process for checking the readability of the backup media. Similarly, for the remaining policies, the countermeasures may be adapted.

2 Tryout-II: The SRS of the second project provides concrete details concerning the software goals as identified by Work Packages 1, 2 and 3 of the Cohort Oriented Virtual Campus for Effective Language Learning (COVCELL) Project. The intended audience of the SRS is primarily the partners of the COVCELL Project, but it also addresses all other parties that might have an interest in the software under development (e.g. the wider Moodle Open Source community).

Table II. Tryout Data (Project-II) for Risk Factor

S.N.	Policy	PCF	Compliance Status	Weighted PCF
1.	*Backup and Recovery*	0.79	1	0.79
2.	*Encryption*	1.61	1	1.61
3.	*Data Classification*	3.15	1	3.15
4.	*Non-Repudiation*	1.52	1	1.52
5.	*'Confidentiality of Data'*	1.64	1	1.64
6.	*Virus Protection*	2.19	1	2.19
7.	*Event Log and Audit Trail*	1.55	1	1.55
8.	*Access Controls and Rights*	2.43	1	2.43
9.	*Authentication*	2.25	1	2.25
				∑ WPCF = 17.24
				RF == (17.24) / 9 = 1.92

The overall objective of the COVCELL Project is to address the need, established in current work on online language learning, for a virtual environment in which language learners can meet and interact in the process of language study – a 'virtual campus' for language learning. This virtual campus will be attained by developing software modules for the Open Source CMS (Course Management System) Moodle. These modules will focus on supporting the teaching of languages, but will offer functionality that may be applicable to a broader group of users, beyond the language teaching community. A summary of the estimated values of different policies and finally calculated risk factor based on the aforementioned techniques is described in the Table II, as follows:

Now, the value of the risk factor (RF) is compared with the threshold values, as specified in the RAF. It can also be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of the software. Here, the value of the RF is 1.92, which lies at the high risk zone as specified in RAF. This value is also not tolerable. Hence, requirement engineers should revise the SRS by incorporating the security related points then only they should move to the Design Phase of SDLC. Again, backup and recovery related attributes must be taken into consideration because there is only one compliance for this policy. For other policies, the attributes which are not complied and the value '0' is assigned, must be taken into consideration. The countermeasures may also be used for the same. For example, in backup policy, following countermeasure may be used: a tool for checking and reporting about daily backup, a mechanism for periodic backup logs and review and a process for checking the readability of the backup media. Similarly, for the remaining policies, the countermeasures may be adapted.

4.3 Tryout-III: The project, Community Influencer Service Framework (CISF) is a next generation globalized and scalable collaboration platform to achieve business vision.

Table III. Tryout Data (Project-III) for Risk Factor

S. N.	Policy	PCF	Compliance Status	Weighted PCF
1.	*Backup and Recovery*	2.38	1	2.38
2.	*Encryption*	1.98	1	1.98
3.	*Data Classification*	2.05	1	2.05
4.	*Non-Repudiation*	1.48	1	1.48
5.	*'Confidentiality of Data'*	1.57	1	1.57
6.	*Virus Protection*	2.75	1	2.75
7.	*Event Log and Audit Trail*	1.30	1	1.30
8.	*Access Controls and Rights*	2.10	1	2.10
9.	*Authentication*	2.58	1	2.58
				∑WPCF = 18.19
				RF == (18.19) / 9 = 2.02

It highlights the value of Community Influencers, enables MS groups to recognize and engage these influencers in a systematic and predictable manner, promotes the confidence in Community, and drives affinity toward Microsoft. CISF requirements have been captured by Microsoft team in form of reusable engines and functional modules in a series of documents. A summary of the estimated values of different policies and finally calculated risk factor is described in the Table III.

Now, the value of the risk factor (RF) is compared with the threshold values, as specified in the RAF. It can also be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of the software. Here, the value of the RF is 2.02, which is at the high risk as specified in RAF. This value is not tolerable. Hence, requirement engineers should revise the SRS by incorporating the security related features. Here, event log and audit trails related attributes must be taken into consideration on high priority because there are maximum non-compliances for this policy. For other policies also, the attributes which are not complied and the value '0' is assigned, must be taken into consideration. The countermeasures may also be used for the same. For example, in event log and audit trail policy, following countermeasure may be used: by framing a directive for employees accountability, a tool for security breaches, periodic review of internet connection, system monitoring and compliance monitoring, a mechanism for the periodic review of security environment and the practice monitoring of IT users. Similarly, countermeasures may also be adapted for the remaining policies.

4.4 Tryout-IV: The purpose of this SRS is to analyze and define the high-level design needs and features of the Accident and Incident Reporting (AIR) – an Intranet based web application for Compass Group of users. It focuses on the architecture used, the various classes that will be needed and the actions that the user can perform. This document covers the design of the AIR Application. This design includes the User Interface Layer, Middleware Business Logic Layer and Data Access Layer. It also includes the Business Entities (Entities and Transfer Objects), which will be used across the three layers for data transfer. A summary of the estimated values of different policies and finally calculated risk factor is described in the Table IV, as follows:

Now, the value of the risk factor (RF) is compared with the threshold values, as specified in the RAF. It can also be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of the software. Here, the value of RF 1.75 which is at the high risk as specified in RAF; this value is not tolerable. Hence, requirement engineers should revise the SRS by incorporating the security related points. Here, again event log and audit trails related attributes must be taken into consideration on high priority because there are maximum non-compliances for this policy. For other policies also, the attributes which are not complied and the value '0' is assigned, must be taken into consideration. The countermeasures may also be used for the same. For example, in event log and audit trail policy, several tools/countermeasures may be used including framing a directive for employees accountability, a tool for security breaches, periodic review of internet connection, system monitoring and compliance monitoring, a mechanism for the periodic review of security environment and the practice monitoring of IT users. Similarly, countermeasures may also be adapted for the remaining policies for obtaining the maximum compliance in each security policy.

Table IV. Tryout Data (Project-IV) for Risk Factor

S. N.	Policy	PCF	Compliance Status	Weighted PCF
1.	*Backup and Recovery*	1.55	1	1.55
2.	*Encryption*	1.80	1	1.80
3.	*Data Classification*	1.08	1	1.08
4.	*Non-Repudiation*	2.47	1	2.47
5.	*'Confidentiality of Data'*	1.91	1	1.91
6.	*Virus Protection*	2.21	1	2.21
7.	*Event Log and Audit Trail*	0.64	1	0.64
8.	*Access Controls and Rights*	1.13	1	1.13
9.	*Authentication*	3.00	1	3.00
				∑WPCF =15.79
				RF == (15.79) / 9 = 1.75

4.5 Tryout-V: The purpose of this SRS is to analyze and define the high-level design needs and feature s of the Sales Outlook and Forecasting Tool (SOFT) Application. It focuses on the architecture used, the various classes that will be needed and the actions that the user can perform. This document covers the design of the SOFT Application. This design includes the User Interface Layer, Middleware Business Logic Layer and Data Access Layer. It also includes the Business Entities which will be used across the three layers for data transfer. A summary of the estimated values of different policies and finally calculated risk factor is described in the Table V.

Now, the value of the risk factor (RF) is compared with the threshold values, as specified in the RAF. It can also

be decided by the requirement engineers according to the security needs; the threshold value may vary according to the security requirements of the software. Here, the value of RF is 2.04, which is at the high risk as specified in RAF. This value is not tolerable at any cost. Hence, requirement engineers should revise the SRS by incorporating the security related points. Here, non-repudiation related attributes must be taken into consideration on high priority because there are maximum non-compliances for this policy. For other policies also, the attributes which are not complied and the value '0' is assigned, must be taken into consideration. The countermeasures may also be used for the same. For example, in non-repudiation policy, following countermeasure may be used: by framing a directive for authentication by digital signature and its uniqueness and compliance with IT (Amended) Act, 2008, a mechanism for the accuracy of the information and metadata of the document etc. Similarly, countermeasures may also be adapted for the remaining policies.

Table V. Tryout Data (Project-V) for Risk Factor

S. N.	Policy	PCF	Compliance Status	Weighted PCF
1.	*Backup and Recovery*	3.17	1	3.17
2.	*Encryption*	1.44	1	1.44
3.	*Data Classification*	2.13	1	2.13
4.	*Non-Repudiation*	0.98	1	0.98
5.	*'Confidentiality of Data'*	1.64	1	1.64
6.	*Virus Protection*	2.69	1	2.69
7.	*Event Log and Audit Trail*	2.6	1	2.6
8.	*Access Controls and Rights*	1.54	1	1.54
9.	*Authentication*	2.21	1	2.21
				∑ WPCF =18.4
				RF == (18.4) / 9 = 2.04

V. COMPARISION OF RESULTS

Summary of the tryout results of all the five SRSs is given in the Table VI. For the comparison of results, we wanted a rating from the SRS providers. But in industry, this is highly informal. Hence, they were unable to provide any quantified value for the Risk Factor. They could only provide a general opinion with respect to security as saying that 'we rate them highly insecure'. Although, we do correlation analysis in such type of studies but due to the limitations, as discussed above, it could not be feasible. Since, their revelation about these SRSs is that they are deficient in security, it confirms our results. From these evidences, the utility of RAF is automatically ascertained to some extent. However, it may not be enough/sufficient to conclude so strongly about the effectiveness of the assessment mechanism by us but certainly up to some extent. Therefore, we recommend our proposal i.e. RAF for the risk assessment through SRS of a project, particularly meant for the requirements phase.

Table VI. Summary of Tryout Results

S. N.	Name of the Project	Risk Factor	Risk Level
1.	Student Course Information System (SCIS)	1.38	High
2.	Cohort Oriented Virtual Campus for Effective Language Learning (COVCELL) Project	1.92	High
3.	Community Influencer Service Framework (CISF)	2.02	High
4.	Accident and Incident Reporting (AIR)	1.75	High
5.	Sales Outlook Forecasting Tool (SOFT)	2.04	High

VI. CONCLUSION AND FUTURE WORK

The risk assessment of SRSs of all the five projects is elaborated and discussed, which reveals that the project SRSs lack severely on security aspects and need major modifications/revisions. Requirement engineers should revise the SRS documents and they should add the security flavor before proceeding to the design phase for building secure software, which is the thrust area for the customers as well as industry. Overall industry results also verify and support our scrupulous recommendations. Since RAF is developed by us; hence, the personal Hawthrone-effect on the findings cannot be ruled out. But, the assessment appears to be quite significant at the rating as well as giving incites with regard to correction, incorporation, modification, and enhancement of SRSs to frame better compliance on security features.

Although RAF is validated on five live projects; however, to generalize the results, further study on a larger sample of SRS is needed. A software tool may also be developed for the automation of this complete process. In future, depending upon the need of the project and advancement in technology, some more policies may also be added. This work may also be extended for the further phases of SDLC by developing various checklists as per requirement and chaining with requirement phase policies. The work may provide guidance to the researchers and industry persons for developing more secure software.

VII. REFERENCES

[1] Pandey S. K., Mustafa K., Ahson S. I. (2007, December). A checklist based approach for the mitigation of stack overflow attacks. In the Proceedings of the Third IEEE International Conference on Wireless Communications and Sensor Networks, WCSN 2007, (pp. 174-176). Allahabad, India

[2] Mazumdar Chandan, Barik Mridul Sankar, Sengupta Anirban. (2007, September). Enterprise information security risk analysis: A quantitative methodology. In the Proceedings of the National Workshop on Software Security. (pp. 1-12). N. Delhi, India.

[3] Hirsch Corey & Ezingeard Jean- Noel. (2008, January). Perceptual and cultural aspects of risk management alignment: a case study. Journal of Information Systems Security, JISSec, 4(1), 3-20.

[4] Stoneburner Gary, Goguen Alice, Feringa Alexis. (2002, July). Risk management guide for information technology systems. NIST Special Publication, (800-30). Retrieved February 14, 2008, from http://csrc.nist.gov/publications/nis tpubs/800-30/sp800-30.pdf

[5] Abdullah Tahir, Mateen Ahmed, Sattar Ahsan Raza, Mustafa Tasleem. (2010). Risk analysis of various phases of software development models, European Journal of Scientific Research, ISSN 1450, 140(3), 369-376.

[6] Allen Julia, H. Barnum, Sean Ellison, Robert J., McGraw Gary, Mead Nancy R. (2008). Software security engineering: A guide for project managers. (pp. 6-8). Addison Wesley Professional.

[7] Ashbaugh Douglas A. (2008, October, 23). Security software development, assessing and managing security risk. CRC Press.

[8] Mustafa K. & Pandey S. K. (2010, January). A comparative study of risk assessment methodologies. International Journal of Computer Science and Information Security. (Accepted)

[9] Pandey S. K. & Mustafa K. (2010, Sep-Oct). Risk Assessment Framework (RAF). International Journal of Advanced Research in Computer Science. 1(3), 423-432.

[10] Mustafa K., Pandey S. K., Rehman S. (2008, September). Security assurance by efficient access control and rights. CSI Communication, 32(6), 29-33.

[11] Mustafa K., Rehman S., Pandey S.K. (2009, March): Confidentiality related security assessments. IEEE International Advance Computing Conference. Patiala.

[12] Pandey S. K., Mustafa K. (2010, July). Recent Advances in SRE Research. International Journal of Computer Science and Engineering, 2(4), 1079-1085.

[13] Pandey S. K., Mustafa K. (2010, Aug). Security Assurance: An Authentication Initiative by Checklist. International Journal of Advanced Research in Computer Science, 1(2), 110-113.

[14] S. K. Pandey, K. Mustafa: Security Assurance through Efficient Event Log and Audit Trails, Journal of Global Research in Computer Science, USA, Illinois, Vol. 3, No. 1, Jan, 2012, pp. 27-30.

AUTHOR

Dr. Santosh K. Pandey is presently working as Scientist 'C' with the Department of Electronics & Information Technology (DeitY), Ministry of Communications & IT, Government of India, New Delhi. Before joining DeitY, he was a Faculty of Information Technology with Board of Studies, the Institute of Chartered Accountants of India (Set up by an Act of Parliament) New Delhi. Prior to this, he worked with the Department of Computer Science, Jamia Millia Islamia (A Central University) New Delhi and Directorate of Education, Govt. of NCT of Delhi. He has a rich Academics & Research experience in various areas of Computer Science. His research interest includes: Software Security, Requirements Engineering, Security Policies and Standards, Formal Methods, Cloud Computing, Security Metrics, Vulnerability Assessment etc. He has published around 46 high quality research papers and articles in various acclaimed International/National Journals (including IEEE, ACM, CSI) and Proceedings of the reputed International/ National Conferences (including Springer). Out of these publications, most of them have good citation records. He has been nominated in the board of editors/reviewers of various peer-reviewed and refereed Journals. In addition, he has also served as a Program Committee Member of several reputed conferences in India as well as abroad. He has also been designated in various academic/research committees by the government organizations as well as software companies as a subject expert.