ASEAN Engineering Journal

Full Paper

AN EMPIRICAL STUDY TO IDENTIFY AND WEIGH THE RISK ATTRIBUTES THROUGH THE APPLICATION OF FMEA IN THE SOFTWARE INDUSTRY WITH FOCUS ON MSME AND SME IT FIRMS IN BENGALURU REGION

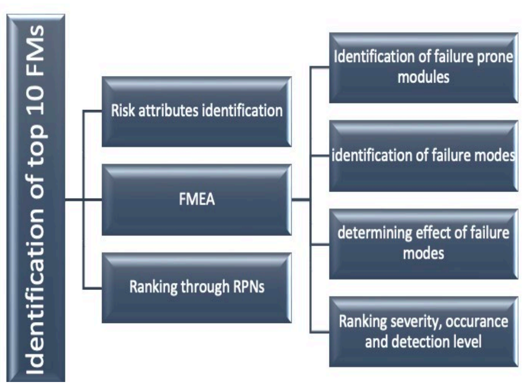
Article history
Received
11 June 2023
Received in revised form
05 September 2023
Accepted
12 September 2023
Published online
29 February 2024

Sandhya Soman[a]*, Piyush Kumar Pareek[b]

[a]GITAM School of Sciences, Dept of Computer Science, Bengaluru, Karnataka, India.
[b]Department of Artificial Intelligence & Machine Learning, NITTE Meenakshi Institute of Technology, Bengaluru, Karnataka, India

*Corresponding author
ssoman@gitam.edu

Graphical abstract



Abstract

Uncertain events can occur anytime during the lifecycle of software. An organization's readiness to deal with these events determines its ability to mitigate the potential losses. Risk management is an essential component of the lifecycle process and is given due diligence in a typical multi-tiered environment of the technological giants. On the contrary, small IT firms, especially the ones falling under the belt of SME and MSME sectors, often follow a reactive approach to risk management. Consequently, the probable losses incurred can be huge and sometimes even challenge the firm's existence in the market. Considering the above issue, we have put forward this paper after performing an empirical study targeting only the Bengaluru region's MSME/SME IT firms. The study was primarily conducted with two objectives (1) to identify the risk attributes and (2) to determine the effectiveness of FMEA techniques and weigh these attributes through the FMEA process. 43 failure modes were identified, which were ranked by a panel of 12 QA experts based on their past project experiences. The top ten failure modes identified by the process will be considered for future analysis and for determining the mitigation steps as a part of the risk management process, particularly for start-ups.

Keywords: Software quality, risk attribute, FMEA, MSME-IT firms, software failure modes, Risk Management in MSME/SME IT

1.0 INTRODUCTION

Risk Management is often underrated in the software lifecycle. This statement is particularly true for start-ups that usually follow a hybrid lifecycle mode. QA activities are cut short or replaced with other activities to meet stringent deadlines. This omission affects the quality of the final product. A risk management technique that has recently sought the attention of software professionals is FMEA, although it has yet to receive widespread popularity. The literature has shreds of evidence of the application of this technique by well-established QA teams. This technique can mitigate risk and add to the existing quality assurance process.

FMEA is a quality assurance process often used as a risk management technique. Firms often use it as a quantitative tool to determine potentially problematic areas. The steps followed in software FMEA are custom-tailored to suit a given organization. We have, however, tried to generalize the standard steps.

1. The stakeholders identify the modules/components that could fail in the production environment.
2. The failure modes are identified and numbered.
3. The effect of each of these modes is identified.
4. Every mode is assigned severity, occurrence, and detection ranking. The ranking is on a range from 1-10.

5. The risk priority number, which indicates the risk by an entity, is calculated by taking the product of severity ranking, detection ranking, and occurrence ranking.
6. Once the RPN is calculated, it is further utilized to deal with the failure modes in the order of its weight. The higher the value of RPN, the more damage could be caused by the failure mode.

One of the primary advantages of FMEA is that it can serve as an excellent tool for analyzing failures and reducing their adverse impact. They also enhance the reliability and quality of the underlying process and product.

The number of papers stressing the use of quality measures to improve the software process in SMEs is quite limited. In this section, we have listed a notable few.

Elsalam et al. [3] performed a study in Sudan to identify the contributary factors for software quality. They identified the need for more process awareness improvement, unavailability of skilled staff, and lack of communication among team members as the key contributory factors. A similar attempt was made by Mishra et al. [7] to understand why the start-ups struggled and failed so soon. The authors attributed the lack of a flexible model as a primary reason since it could not accommodate [20] the dynamicity of the market. Also, the need for a metric/matrix to track the accountability of the team members in the various tasks hindered getting an accurate picture of the project's progress. Another viewpoint to this was given by Poth et al. [9], who concluded that if an organization would have opted for a leagile setup from the onset, then it could bring down the cost involved in project completion and can also assure conformance to timelines and this fact, would be particularly beneficial for the start-ups.

The impact of Requirements Engineering [3] on project success was studied by Aguilar et al. [1]. The study revealed RE to be an often-neglected phase in the lifecycle [19] of projects in SME firms. Giving proper attention could be a contributory factor in improving the project lifecycle. [21] highlights the most overlooked failure modes through CDE- an enumerated list containing the FMs for software systems since 1960.[22] discusses the application of FMEA for intelligent systems for assessing the risks. The authors have proposed an extension of FMEA to provide a more structured approach to identify hazards and scenarios that could lead to losses in systems and their associated sub-systems.

In an attempt to understand the challenges faced by SMEs, a focus group study was conducted by Farsi [4]. The challenges identified included a need for more training, no emphasis on continuous learning, limited funds, and restrictions. In his research report, Kesab Das [2] discussed the possibilities of encouraging the SME segment of India through measures like generating healthy competition among peers and support for funds so that they can focus on innovation. According to the author, the most critical area is to enhance the skills/upskill the human resources. Mukherjee et al. [8] cited the lack of current technology as a significant reason for the failure of budding IT firms, as this results in products of low quality and more susceptibility to failures. The above study clearly shows the need for more empirical evidence on start-up process improvements. Most of the measures used are based on a build-and-fix approach with little or no flexibility in the process being followed [5]. There needs to be more quality assurance and risk management activities followed. Either there is no proper QA process, or the activities are skipped due to lack of time.

However, little or no evidence has been found for the MSME/SME IT firms [11]. One motivation for applying this technique could be to improvise the overall testing process. FMEA can enable firms to identify those areas to be prioritized for test case writing, which module to test, and which ones have less risk if skipped. This study was conducted to understand the risk attributes of a budding IT enterprise [6] and understand its impact on the other modules. This paper enlists the findings of the application of FMEA to identify and unveil the risk attributes, particularly for small and budding enterprises. The study is conducted in the Bengaluru region. The organization of the paper is as follows. The next section discusses a brief overview of the FMEA process. The third section discusses the methodology adopted. The subsequent section discusses the application of FMEA and its results, and the last section highlights the key findings of our study.

2.0 METHODOLOGY

The quality assurance process must become more stringent to ameliorate the quality of the software. The bigger IT Firms [10][13] have several quality assurance methods, which cannot be effectively applied to the smaller firms due to their limited bandwidth, resources, and the constrained environment in which they operate. Since our present study aims to understand the software process followed in such firms and uncover ways to improve the process to enhance the software quality, we decided to perform Software FMEA to determine the possible causes of failure and risk factors. This enabled us to identify the failure modes and their effects on the software. There has been sufficient evidence available in the literature for applying FMEA to larger firms. Still, to our knowledge, such evidence has yet to be available for the smaller IT Firms.

Data Collection: Our study is focused on the MSME IT Firms in the Bengaluru Area. We identified the MSME IT firms in and around Bengaluru's Rural and Urban areas and requested them to participate in our study. Out of 30+ firms approached, we got a positive response from 12. Due to their request for anonymity, their names are not listed in this paper. We adopted the following inclusion criteria: firms in the software industry for >2 years have delivered>5 projects and are under the MSME sector. Currently, the paper focuses on the insights gained in the context of Bengaluru-based firms. Still, as a continuation of the current work, we intend to extend the same to other areas to generalize the results in the next phase.

We conducted structured interviews [14] with the professionals from these firms and tried to document the common failure modes they had [12] experienced.

Application of FMEA: Through discussion with the quality experts, we identified 43 failure modes from PF1 through PF43. Excluding the common ones and merging the semantically similar ones, 18 effects related to the Failure Modes were charted out [18]. The next step was the identification of the possible effects. Some of the failure effects were common. As the next part of the process, we assigned severity and occurrence degree of detection to each failure mode. The list was passed to a panel of 12 Quality Experts (focused group for the current study). The professionals were asked to rank these modes for severity, occurrence, and detection subjectively and anonymously. The rankings' mean was considered the final ranking for each category. Based on these rankings, the RPN was calculated, and the modes were arranged in descending order. The higher the RPN number, the greater the risk associated with the failure mode. The top-ten failure modes were extracted and further validated through the Quality Assurance Experts from two software firms. In the subsequent sections, we also propose measures that can be adapted to deal with these failure modes.

3.0 RESULTS AND DISCUSSION

In this section, we describe the FMEA process followed by us to gain insight into the failure modes of the software.

Step 1:

For this step, the data was gathered through an in-person interview/Google Meet with the Quality Experts and documenting their views. [17] We listed the modes cited by them and assigned each of them an ID so that we could use them for further analysis. Also, the semantically similar modes were combined as a "single mode," and all such combined modes were validated by at least three software experts based on their availability. The modes identified are enlisted in Table 1.

Table 1 Potential Failure Modes

PF MODE	PF DESCRIPTION
_PFM_1	Lack of clarity in requirements until late in the process
_PFM_2	Fluctuations in priorities amid sprints
_PFM_3	Non uniformity in user story points estimation
_PFM_4	Technical Debts
_PFM_5	Lack of documented support
_PFM_6	Unvalidated requirements
_PFM_7	Random allocation of tasks (push rather than pull system)
_PFM_8	Poor infrastructural support
_PFM_9	No change management
_PFM_10	QA practices skipped
_PFM_11	Siloed work environment
_PFM_12	No clear definition of acceptance criteria
_PFM_13	Random team composition
_PFM_14	Unaccounted schedule slippage
_PFM_15	Planning considered time consuming/erroneous/unnecessary
_PFM_16	Resources working on multiple projects
_PFM_17	Unavailability of customers
_PFM_18	Improper budget planning

Step 2: Effect of Failure Modes (EFMs)

In the next step, the effects of each failure mode were determined. The effects were listed as _EF_1: End Product Affected, _EF_2: Acceptance test failed, _EF_3: Unsatisfied customer, _EF_4: Compromised quality, _EF_5: Under/improper utilization of resources, _EF_6: Increase in budget

Step 3: Mapping of FMs with EFMs

Table 2 reflects the mapping used to determine the effect of each failure mode with the corresponding effects.

Table 2 Mapping of FMs with EMs

	_EFM_1	_EFM_2	_EFM_3	_EFM_4	_EFM_5	_EFM_6
_PFM_1	X		X	X		X
_PFM_2	X			X		
_PFM_3		X	X			
_PFM_4	X					
_PFM_5					X	
_PFM_6	X					X
_PFM_7				X	X	
_PFM_8	X			X		
_PFM_9	X		X	X		
_PFM_10	X		X	X		
_PFM_11					X	
_PFM_12		X	X			
_PFM_13					X	
_PFM_14			X			X
_PFM_15						
_PFM_16	X		X	X		
_PFM_17	X					
_PFM_18				X		X

Step 4: Assigning of Severity, Occurrence, Degree of detection for each failure modes.

We formed a panel of 12 software experts from different organizations to assign each failure mode's degree of severity, occurrence, and detection. [16] The average of the ratings was considered for analysis. RPN measure for each failure mode was calculated through the product of the degree of severity with the degree of occurrence and ease of detection. The standard scales used for rating were from [1-10], one (1) being the least and ten (10) being the highest. The mean score rankings assigned by the Quality Panel as depicted in Table 3

Table 3 Severity, Occurrence, and Detection Ranking

	Severity	Occurrence	Detection	RPN
_PFM_1	2.45	7.00	8.67	148.75
_PFM_2	6.63	4.28	3.93	111.40
_PFM_3	9.07	8.27	5.34	400.55
_PFM_4	8.24	5.31	3.38	147.87
_PFM_5	3.97	6.70	3.67	97.42
_PFM_6	3.92	9.05	9.89	350.86
_PFM_7	5.07	3.30	3.67	61.47
_PFM_8	1.44	8.21	9.96	117.54
_PFM_9	6.79	3.43	9.82	228.52

_PFM_10	5.34	2.42	3.45	44.48
_PFM_11	1.09	3.65	4.44	17.71
_PFM_12	4.82	7.77	8.50	318.59
_PFM_13	4.36	6.69	8.51	248.55
_PFM_14	4.94	3.54	1.07	18.67
_PFM_15	7.39	7.06	6.38	332.58
_PFM_16	3.96	9.95	5.25	206.86
_PFM_17	9.09	3.80	2.43	83.86
_PF_18	4.75	2.18	8.67	89.79

Step 5: Determining top ten failure modes

Based on the RPN ratings, the modes were arranged in descending order of RPN, as shown in the Table 4 (results have been rounded off).[15] The top 10 modes have been highlighted in red. The results were validated by the QA experts from two firms, who chose to keep their identities confidential. They agreed upon the empirical results being calculated

Table 4 Top Ten Failure Modes

PF No.	Mode	RPN
_PFM_3	Unvalidated requirements	400
_PFM_6	Non uniformity in user story points estimation	351
_PFM_15	Planning considered time consuming/erroneous/unnecessary	333
_PFM_12	No clear definition of acceptance criteria	319
_PFM_13	Random team composition	249
_PFM_9	No change management	229
_PFM_16	Resources working on multiple projects	207
_PFM_1	Lack of clarity in requirements until late in the process	149
_PFM_4	Technical Debts	148
_PFM_8	Poor infrastructural support	118
_PFM_2		111
_PFM_5		98
_PFM_18		90
_PFM_17		88
_PFM_7		61
_PFM_10		44
_PFM_14		19
_PFM_11		18

4.0 CONCLUSION

Determining the failure modes for small sector IT Firms is challenging, as the software process needs to be better defined. Today, the SDLC cycle is embracing automation. It has also opened its arms to incorporating Machine Learning to understand the patterns and trends in the lifecycle process so that meaningful insights can be drawn from them. It can be utilized to ensure that the software quality is upheld. The uncertainties associated with quantifying the effort involved in different types of projects are evident in the literature. Another fact that requires attention is that in the post-COVID era, many start-ups have blossomed in the market. It is crucial to understand the existing process followed by them and streamline it so that we can reduce the number of failures encountered by these firms and promote MSME in a country like ours. To understand this, we considered the companies that have been in the market for over a year and have delivered more than three projects. We applied Software-FMEA based on the data collected from the Quality Experts and uncovered “ten” significant failure modes. The failure causes analysed can be enlisted as non-uniformity in user story point estimation, unvalidated requirements, non-significant attitude towards planning, unclear/ambiguous definition of acceptance criteria, Random team composition, no change management process, Resources working on multiple projects, Lack of clarity in requirements until late in the process, Technical Debts, Poor infrastructural support.

As a part of an extension of the current work, we would be considering the applicability of these failure modes for various types of projects based on their genres, like ML projects, DS projects, and Agile projects, and see whether the results resonate with them. Also, the risk assessment can be automated into software tools once a substantial list of standard failure modes, agreed upon by the experts, is charted out. The ultimate aim of this study is to come forward with a process model or framework custom-tailored to the environment and opportunities in a smaller IT Firm and can fit their lifecycle process models.

Acknowledgement

Sincere gratitude to the SME/MSME firms (IT) of Bengaluru for providing valuable insights and support for the conduction of this study. providing valuable insights and support for the conduction of this study.

References

[1] Aguilar, J., Zaldívar Colado, A., Tripp-Barba, C., Espinosa, R., Misra, S., and Zurita, C. E. 2018. A Survey About the Impact of Requirements Engineering Practice in Small-Sized Software Factories in Sinaloa, Mexico. Computational Science and its Applications – ICCSA 2018. *Lecture Notes in Computer Science. 10963, Springer, Cham.* DOI: https://doi.org/10.1007/978-3-319-95171-3_26

[2] Carbone, T. A., and Tippett, D. D. 2004. Project Risk Management Using the Project Risk FMEA. *Engineering Management Journal*, 16(4): 28-35. DOI: https://doi.org/10.1080/10429247.2004.11415263

[3] Solyman, A., Ibrahim, O. A., and Elhag, A. A. M. 2015. Project management and software quality control method for small and medium enterprises, *2015 International Conference on Computing,*

*Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)*123-128. DOI: https://doi.org/10.1109/iccneee.2015.7381442

[4] Farsi, J. Y., and Toghraee, M. R. 2014. Identification the main challenges of small and medium sized enterprises in exploiting of innovative opportunities (Case study: Iran SMEs). *Journal of Global Entrepreneurship Research.* 2(1): 4. DOI: https://doi.org/10.1186/2251-7316-2-4

[5] Pai, A., S, H S., Soman, S., and Pareek, P. 2019. Analysis of Causes and Effects of Longer Lead Time in Software Process Using FMEA. *Proceedings of the Second International Conference on Emerging Trends in Science & Technologies for Engineering Systems (ICETSE - 2019)* DOI: https://doi.org/10.2139/ssrn.3508574

[6] Kupiainen, E., Mantyla, M. V., and Itkonen, J. 2015. Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies. *Information & Software Technology*. 62: 143–163 DOI: https://doi.org/10.1016/j.infsof.2015.02.005

[7] Mishra, D., and Mishra, A. 2009. Software process improvement in SMEs: A comparative view. *Computer Science and Information Systems*. 6(1): 111–140. DOI: https://doi.org/10.2298/csis0901111m

[8] Mukherjee, S. 2018. Challenges to Indian micro small scale and medium enterprises in the era of globalization. *Journal of Global Entrepreneurship Research*, 8, Article: e28. DOI: https://doi.org/10.1186/s40497-018-0115-5

[9] Poth, A., Sasabe, S., Mas, A., and Mesquida, A. L. 2019. Lean and agile software process improvement in traditional and agile environments. *Journal of Software,* 31 Article: e1986. DOI: https://doi.org/10.1002/smr.1986

[10] Soman, S., and Pareek, P. 2020. An exploratory analysis on challenges prevailing in small and medium IT firms. *Journal of Physics*, *Conference Series*. DOI: https://doi.org/10.1088/1742-6596/1427/1/012010

[11] Soman, S., and Pareek, P. 2022. An empirical study to understand the effect of the awareness of Agile and Lean Software Development Life Cycle on factors to reduce the time taken for product development. *International Journal of Health Sciences (IJHS).* 6338–6344. DOI: https://doi.org/10.53730/ijhs.v6ns3.7405

[12] Soman, S., Aditya Pai, H., Pareek, P., and Ramesh, B. N. 2019. Study on the factors that increase the lead times in small and medium level enterprises in IT sector. *Journal of Advanced Research in Dynamical and Control Systems,* 11(4 Special Issue): 1689-1698

[13] Soman, S., Pareek, P., Dixit, S., Chethana, R.M., and Kotagi, V. 2021. Exploration Study to Study the Relationships Between Variables of Secure Development Lifecycle (SDL). *Emerging Technologies in Data Mining and Information Security. Lecture Notes in Networks and Systems*, 164. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-15-9774-9_59

[14] H, A. P., S, S. H., Soman, S., and Pareek, P. 2019. ROC Structure Analysis of Lean Software Development in SME's Using Mathematical CHAID Model. *Social Science Research Network*. DOI: https://doi.org/10.2139/ssrn.3508577

[15] Gandomani, T. J., and Nafchi, M.N. 2015. An empirically-developed framework for agile transition and adoption: A grounded theory approach. *Journal of Systems and Software*. 107: 204-219 DOI: https://doi.org/10.1016/j.jss.2015.06.006

[16] Shi, H., Wang, X., Li, G., and Zhang, H. 2011. FMEA-Based Control Mechanism for Embedded Control Software. *011 International Conference of Information Technology, Computer Engineering and Management Sciences. 110-112*. DOI: https://doi.org/10.1109/icm.2011.244

[17] Pandey, V., Kadekodi, P., Bhatarkar, G. S., and Manohar, R. 2021. A Proposed Systematic Software Robustness Verification Framework (SRVF) for Enhancing Critical Software Module Robustness. In *SAE Technical Paper Series.* DOI: https://doi.org/10.4271/2021-26-0481

[18] Velmourougan, S., Dhavachelvan, P., Baskaran, R., and Ravikumar, B. 2014. Software development Life cycle model to build software applications with usability. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 271-276. DOI: https://doi.org/10.1109/icacci.2014.6968610

[19] Dong, W., Wang, J., Zhao, C., Zhang, X., & Tian, J. 2008. Automating Software FMEA via Formal Analysis of Dependence Relations. 32nd Annual IEEE International Computer Software and Applications Conference. 490-491. DOI: https://doi.org/10.1109/compsac.2008.71

[20] Ozarin, N. W., & Siracusa, M. (2003). A process for failure modes and effects analysis of computer software. *Annual Reliability and Maintainability Symposium. 365-370.* DOI: https://doi.org/10.1109/rams.2003.1182016

[21] Neufelder, A. M. (2023). Software FMEA and the common Defect Enumeration. 2023 *Annual Reliability and Maintainability Symposium. 1-6*. DOI: 10.1109/RAMS51473.2023.10088208

[22] Hanna, A., Bengstsson, K., Larsson, S., and Gotvall, P-L (2023). Risk assessment for Intelligent and Collaborative Automation System by Combining FMEA and STPA. DOI: http://dx.doi.org/10.2139/ssrn.4530824