Risk Management Building Block for the Open Agile Software Development Life Cycle (OASDLC)

Anand Kumar Rai, Shalini Agrawal, Mazhar Khaliq, Abhishek Kumar

Abstract*: *The idea of OASDLC was originated specially for "Brihaspati" project and was framed keeping in mind the breaches and boundaries modelled by present Agile Software Development Life Cycle (ASDLC) models. OASDLC was further tested for attaining lower costs and efforts involved in agile project. We analysed this model and found that this model lack of risk management process. This paper proposes a unique OASDLC model incorporated with risk management building block so as to achieve a best practice for open agile software project. Proposed model is tested and validated through MATLAB fuzzy logic simulator and IBM Watson online tool.

Index Terms*: *Risk Management, Agile Software, SDLC

I. INTRODUCTION

Open Source Software (OSS) development is not a software development methodology, but ordinary development practices may be select by analysis within the OSS community. These practices frequently behave to embody the collective nature of OSS projects. A group of people is most important to the OSS. When a vigorous group of people does not available to develop an OSS project, its success possibility become less. Iterative and incremental development is the other OSS common feature. In these practices, it is not unexpected that agile methods, or mechanisms of the agile procedure, are generally used in OSS development.

Open source projects generally obtain the group effort of many geographically distant people who do not share any organizational structure. OSS project indicates that these projects are not candidates to the use of agile methods since some basic values seem to be destroyed. The distance and diversity deteriorate developer's communication, which are the most important values within agile methods. Though, most open source projects share principles with the Agile Manifesto. Always be ready for changes, deliver real features, work with and resolve continuous feedback, respect collaborators and users and to face challenges are expected attitudes of an agile developers obviously initiate in free and open source software communities.

Revised Manuscript Received on May 06, 2019

Mr. Anand Kumar Rai, M.Sc.(Computer Science) Gurughasi Das University, Bilaspur, CG. from Alagappa University, Karaikudi, Tamilnadu. India

Dr. Shalini Agrawal, Department of Computer Science and Engineering, Technology, University of Allahabad, Uttar Pradesh, India

Dr. Mazhar Khaliq, M.C.A. from Aligarh Muslim University and got Ph.D.(Computer Science) from Integral University, Lucknow Uttar Pradesh, India

Dr. Abhishek Kumar, is Assistant Professor, Computer Science at Banaras Hindu University Uttar Pradesh, India

There have been adequately of hypothetical methods offered to describe the Open Source Software Development Life Cycle (OSSDLC). Furthermore there is no definite OSSDLC model as such. There are combined efforts included of developers and users community for OSS (Scacchi et al., 2006). The OSSDLC model has been introduced by Woods D in 2005 (Woods and Guliani, 2005). The model includes comprehensively of seven stages beginning with the knowledge of the software project. The low standard step is to perform formal examination and prove the validity of the requirements as such a public prototype is created. A forum is created by the mailing list, blogs etc. after accomplishing the prototype. For the gradual development of the software it is needed to remove the impurities or unwanted elements from the prototype. This development is organized through a release phase leading to either inactivity or further refining. The prototypical exposes all facets of OSSDLC and has all components for a widely applicable model for mainstream of OSSD projects.

Formulation and hypothesis of Open Agile Software Development Life Cycle model (OASDLC) has been done keeping in mind the limitations drawn by existing Agile Software development Life Cycle (ASDLC) models, further it has been tested for achieving lower costs and efforts. These tests have been further proved by means of hypothesis validation with the help of a survey based research (Subhas C. Misra and Virender Singh, 2013).

In this paper Section 2, the brief history and evolution of open agile software development has been described. In Section 3, proposed model has been introduced. In the existing OASDLC model risk management building block has been incorporated and each phase of the model has been described properly. In section 4, fuzzy rule base has been formed with the help of questionnaire and literature survey, and these rules are applied in Matlab fuzzy inference system simulator. The numerical value of the risk indicators has been collected from the simulator. The data analysis has been made with the help of IBM Watson online learning tool and risk indicators sensitivity has been evaluated. Section 5, describes the concluding remark and future work.

II.BACKGROUND

The open source software (OSS) development has transformed the method software is being developed, maintained and restructured. Typically OSS are dispersed as freeware with or without any charge (Schryen, G., 2011).

OSS has been used to develop Firefox and Apache, Linux, Android etc. It is misconception for open source software development system has poor quality but, Linux open source operating system adaptation has been increased most due to its high quality. Source codes availability means the public could give their contribution in completion of the software in this way making software open to group of developers who could contribute their information and skills aquired through experience and they could customize the software to suit their needs (Scacchi, 2007).

Open source software has been designed to impress positively in distinct fields such as health , education, business, telecommunication, security etc. with or without any payment. In OSS development process communication hurdles occures frequently because the developers communicate through the communication media due to distributed location. (Wang, J., 2012). In addition, OSS projects contributors are not under any kind of responsibility they act independently, for this reason when they loss motivation they make less their voluntary participation for the project, and this causes negative impact on the OSS project (Khanjani & Sulaiman, 2011). The submission of the Public developer's code must go through a formal assessment process where standard, quality and operation are tested before integrating into the project (Pedro Serrador, 2015). The disadvantages of the open source software has been listed like increase in versioning, many imprecise licenses, lack of central developer and problem solver, etc. (Silic, M., & Back, A., 2013). Open source software is generally shared without restriction or interference among developers and researchers that are ready to contribute to raise it to a higher standard, besides this , things became advance as computer became widespread then the non-open source software manufacturer turns into the best business model for various companies to earn profit. Contrarily, companies have started to recognize the value or significance of free and open source software which are highly lucrative. (Wang, J., 2012).

Nowadays numerous web applications practice the open source group contribution to prepare advance documentation, good design and having perception for their software project. For the open source software projects numerous free web hosting facilities are available that deals free services such as mailing list, project management, forum support, bug tracking and many more (Wang, J., 2012).

In this study we referred here OSSDLC model proposed by Woods D in 2005 and OASDLC model has been introduced for Brihaspati software. This model consist of six phases and start with the requirement of the software. Second step accomplishes the experiments and validation of the requirement is the second step after which a public prototype is shaped. Once the prototype is completed a group is made by means of Mailing list, Blogs, and Forums. The prototype is open and for review and further contributions, permits in for an incremental development of the software. This addition is evaluated through a release phase leading to either remove impurities or further stagnant (Subhas c. and Virendra Singh, 2015).

III. PROPOSED RISK MANAGEMENT OASDLC MODEL

Open source planning comprises with many development tools and phases. Figure 1, shows six phases in Open Source Software Development (OSSD). It includes repeatedly software release, minimum planning and system design is required against the traditional development model. OSSD software project empower each individuals to take part to increase coordination and software standardisation and so on (Sen, R. et al., 2012; Khomh, F. et al., 2012a))

The Open Agile software development is used to greatly reduce the project's cost. OASDLC model was developed for Brihaspati software, which has been used in Agile Software Development Life Cycle, but this model lack the risk management block. In this study, we have incorporated risk management building block with OASDLC.

3.1Initial concept and agile method selection: Initial concept is created on the basis of the requirement which is the framework for the concept to work on. For example, if someone want to make a web based application for online sales that becomes a need and the initial concept, for this work company developer (Cdvp) will play a very important role to play in initializing a concept at this point an appropriate agile method must be selected.

3.2Proofing and risk indicator identification: In this step we need survey to find the ideas or concepts are available in the market or not. If it is available we join the available ideas and if not we can go forward to create our own. We need to develop our own forum or communication media in which the public developers (Pdvp) can discuss the ideas or concept put across their contentions at this stage project risk indicators development environment risk, process issue risk and their risk elements must be ascertained.

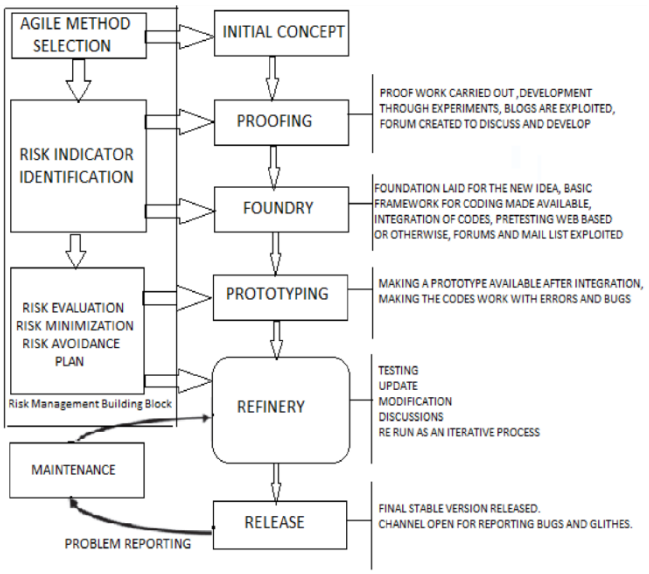

Figure 1. Proposed Risk Management model for OASDLC

3.3 Foundry and risk indicator identification: In this step foundation of the project is formed. The underlying structure for the project is laid out here in which the Cdvp and the management team determine the methods for the source code update and different platforms for contributions such as Mailer lists, open forums, and bug reporting and fixing are allowing.

At this stage rest of the risk indicators must be identified which could not be identified earlier.

3.4 Prototype and risk evaluation, risk minimization: In this phase primary prototype is created with help of both the Cdvp and the Pdvp. this prototype is made available to all for testing, usage and bug reporting. At this phase the software is not entirely prepared yet and is just a prototype and would have large number of issues. The identified risk indicators are evaluated at this stage.

3.5 Refinery and risk evaluation, risk minimization: At this stage the prepared prototype need to be tested and upgraded to make the software better further testing is needed to carry out at industrial standards. A remarkable point is here while making changes it would be difficult for Cdvp to manage such a project single headedly; consequently the problems can be planned at the same time to the mailer list so that multiple solutions can be formulated. The evaluated risk indicators value used to incorporate for project risk minimization.

IV. DISCUSSION AND RESULT

Agile software project risk indicators have been identified (reference) and rule base have been created with the help of online forum discussion with agile experts and open agile software development case studies. The list of risk indicators are given below (Gamalielsson, J. et al., 2011; Kim Dikert et al., 2016; Richard Kemp,2010).

4.1 Development Environment Risk (DER) - An agile environment is defined as an environment that creates and supports a culture that encourages a team of people to work toward a common goal. This is done by incorporating the importance and value of individuals and their interactions - especially in terms of working to achieve quality, collaboration, and acceptance of frequent change in the company culture (Dingsoyr T. et al. , (2013)),. If there is something wrong in the said factors then the probability of agile development risk increases.

4.2 Process Issue Risk (PIR) Agile development process includes agile documentation, software engineering standards . A mechanism to ensuring the software engineering standards, Procedure is needed for tracking and reviewing the performance of subcontractors etc. (Kim Dikert et al, 2016). If there is something wrong in the aforesaid factors then the probability of process issue risk increases.

4.3 Staff Size and Experience Risk (SSE) in agile software development the project manager have to choose the expert people for self-organizing team, Team members should have the right skill set, and staff should be committed for entire duration of the project and necessary agile training must be given to the staff. If there is something wrong in the aforesaid factors then the probability of staff size and experience risk increases.

4.4 Technical Issue Risks (TIR) Application detailed description techniques are introduced for good communication between the customer and developer for the agile software. Specific methods are used for the analysis of agile software. Agile software project quality assessment standard system needed. If there is something wrong in the aforesaid factors then the probability of technical issue risk increases.

4.5 Technology Risk (TR) Innovative Technology to be built to your company needed the use of new analysis, design, and innovative testing methods.it demands the use of innovative or unconventional software development methods. Requirements put more than necessary performance limitation or restriction on the product which need to managed. If there is something wrong in the aforesaid factors then the probability of technology risk increases.

4.6 Schedule Risk (SR) this is need to be ascertain that project Schedule, required resources, and product definition have all been determined by the customer or upper management, The Schedule is hopefulness about the future, "best case," rather than realistic, "expected case". There should not be the cascading delays in dependent tasks. If there is something wrong in the aforesaid factors then the probability schedule risk increases.

Table 1, Shows 12 rules with qualitative values of risk indicators low (L), medium (M) and high (H). These qualitative values have been applied in Fuzzy Inference System (FIS)(Leary Tomlin et al.,2016)

Matlab simulator to generate the quantitative values of project risk indicators and project risk. FIS.

Table 1. Rule base for Open Agile Software Project

S.N.	DER	PIR	SSE	TIR	TR	SR	PR
1	H	H	L	H	M	L	H
2	L	M	M	M	L	M	M
3	M	H	L	H	M	M	H
4	L	L	L	L	L	L	L
5	H	L	L	H	L	L	M
6	H	H	M	M	M	M	H
7	L	L	M	L	M	L	L
8	H	H	H	L	L	L	H
9	H	M	L	H	L	L	H
10	M	M	M	L	L	L	M
11	L	L	H	H	H	H	H
12	H	H	M	M	M	M	H
13	H	H	H	H	H	H	H
14	L	L	M	H	H	M	H

converts qualitative values into quantitative form with the help of fuzzification and the de-fuzzification process. Table 2, shows the quantitative values of project risk and project risk indicators quantitative values which have been collected from FIS. Table 2, shows the qualitative value (low, medium, high) of triangular membership grades on its value scale.

Figure 2, shows the scale of qualitative value on the basis of fuzzy membership function, further these qualitative values sub-divided into its sub categories as per the membership grades. For example Low (very low, low), Medium (medium low, medium, medium high), High (high, very high). In given Figure 2, low triangle at the scale value ranges from 0 to 0.4, on the scale value 0 membership grade of low triangle is highest, hence we can say that 0 represent the very low value and as we move on the scale value, the membership grade of the low triangle decreases and at the scale value 0.4 goes down to 0. Medium triangle ranges from 0.1 to 0.9. On the scale value 0.1 medium membership grade is 0 and at 0.5 it has highest membership grade 1, further at 0.9 medium triangle has minimum membership grade 0 and high triangle ranges from 0.6 to 1, on the scale value 0.6 high triangle membership grade is 0; further we move on the scale value the membership grade increases, on scale value 1 the high triangle membership grade reaches its highest point 1.

Table 2. Qualitative value determination on the basis of fuzzy membership degree

S.N.	Input Scale	Low MF Value	Medium MF value	High MF value	Qualitative value
1	0.0	1.0	0.0	0.0	Very low
2	0.1	0.75	0.0	0.0	Low
3	0.2	0.5	0.25	0.0	Low
4	0.3	0.25	0.5	0.0	Medium low
5	0.4	0.0	0.75	0.0	Medium
6	0.5	0.0	1.0	0.0	Medium
7	0.6	0.0	0.75	0.0	Medium
8	0.7	0.0	0.5	0.25	Medium high
9	0.8	0.0	0.25	0.5	high
10	0.9	0.0	0.0	0.75	high
11	1.0	0.0	0.0	1.0	Very high

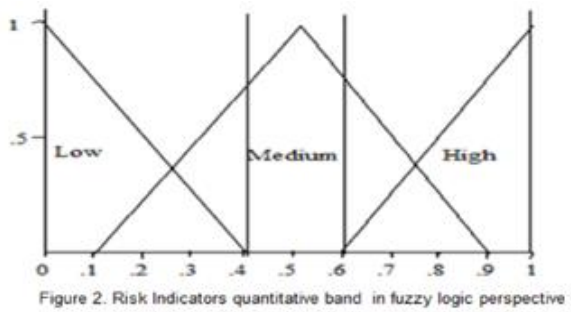

Figure 2. Risk Indicators quantitative band in fuzzy logic perspective

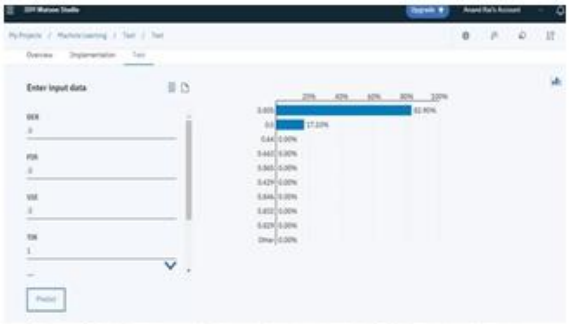
Figure 3. Risk Indicators sensitivity analysis through online IBM Watson learning tool

Table 3. Risk Indicators Sensitivity Analysis

S.N.	DER	PIR	SSE	TIR	TR	SR	PR
1	1	0	0	0	0	0	0.5
2	0	1	0	0	0	0	0.56
3	0	0	1	0	0	0	0.5
4	0	0	0	1	0	0	0.5
5	0	0	0	0	1	0	0.16
6	0	0	0	0	0	1	0.56
7	1	0.5	0.5	0.5	0.5	0.5	0.5
8	0.5	1	0.5	0.5	0.5	0.5	0.58
9	0.5	0.5	1	0.5	0.5	0.5	0.5
10	0.5	0.5	0.5	1	0.5	0.5	0.81
11	0.5	0.5	0.5	0.5	1	0.5	0.86
12	0.5	0.5	0.5	0.5	0.5	1	0.87

Agile software risk indicators sensitivity analysis has been done through IBM Watson learning tool. It can be seen in table 3, we checked the sensitivity of the risk indicators under two condition. In first condition we taken first risk indicator (DER) value very high (1) and rest risk indicators value very low (0) then we found project risk is moderate. We repeated this test for respective indicators in given table 3, we have taken risk indicators value very high which sensitivity to be tested hence we found no one indicators are sensitive in this condition. In second condition we fixed rest five risk indicators value moderate (0.5) and taken one risk indicators value very high (1) which sensitivity to be tested. We repeated this test for given six risk indicators. Finally we found, out of six risk indicators three risk indicators TIR(Technical Issue Risk), TR(Technology Risk) and SR(Schedule Risk) are highly sensitive and DER(Development Environment Risk), PIR(Process Issue Risk) and SSE(Staff Size and Experience risk) are moderate sensitive.

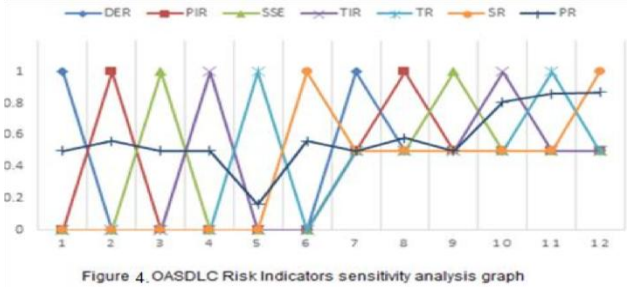

Figure 4. OASDLC Risk Indicators sensitivity analysis graph

V. CONCLUSION

The OASDLC model has been built specifically for the small scale and medium scale open source agile project. I realized that for the large scale agile project development, risk management is very necessary component. It has been seen that large scale open source agile software development project has plenty amount of bugs reporting (Sen, R. et al., 2012). To reduce the amount of bugs and maximize the quality of software's, we needed slight modification in OASDLC model. In this work risk building block has been incorporated in existing OASDLC model which had lacking the risk management. This model has been validated on 14 fuzzy rule base using the Matlab simulators. The fuzzy rule base is created with help of literature review, open agile case studies, discussion with agile professionals through online forum. In this study, three risk indicators (TIR, TR, and SR) have been found very sensitive while other three risk indicators (DER, PIR, SSE) found moderate sensitive.

REFERENCES

1. Conlon, M.P., (2007), An examination of initiation, organization, participation, leadership,and control of successful Open Source software development projects.Information Systems Education Journal 5 (38), 1– 13.
2. Del Bianco, V., Lavazza, L., Morasca, S., & Taibi, D. (2011). A survey on open source software trustworthiness. Software, IEEE, 28(5), Page 67-75.
3. Dingsoyr T, Moe NB.(2013), Research challenges in large-scale agile software development. ACM SIGSOFT Software Engineering Notes. 2013; 38(5):38–9.
4. Feitelson, D.G.(2012), Perpetual development: a model of the Linux kernel life cycle,Journal of Systems and Software 85 (4), 859–875.
5. Gamalielsson, J., Lundell, B., Mattsson, A., (2011). Open Source software for model driven development:
6. a case study. In: Hissam, S. (Ed.), Open Source Systems:Grounding Research. IFIP Advances in Information and Communication Technology,vol. 365. Springer, Heidelberg, pp. 348–367.
7. Khanjani A., & Sulaiman, R. (2011), The Aspects of Choosing Open Source versus Closed Source. IEEE Symposium on Computers & Informatics, page 644- 649.
8. Khomh, F., Dhaliwal, T., Ying, Z., & Adams, B. (2012a). Do faster releases improve software quality? An empirical case study of Mozilla Firefox. Department of Elec. & Comp. Eng., Queen's Univ., Kingston, ON, Canada. IEEE Working Conference Publication, page 179 -188.
9. Kim Dikert, Maria Paasivaara, Casper Lassenius(2016), Challenges and success factors for large-scale agile transformations: A systematic literature review, The Journal of Systems and Software, Elsevier ,Vol. 119(c),pp. 87–108.
10. Leary Tomlin, Derek T. Anderson, Christian Wagner, Timothy C. Havens, James M. Keller (2016),Fuzzy Integral for Rule Aggregation in Fuzzy Inference Systems, Information Processing and anagement of Uncertainty in Knowledge-Based Systems, Springer, Vol. 610,pp 78-90.
11. Mahdi A., Hamed M. and Abushama H. (2013), Popular Agile Approaches in software Development: review and analysis. Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference (IEEE).
12. Mukker A., Singh L. and Mishra A.K. (2014), Systematic Review of Metrics in Software Agile Projects. COMPUSOFT, An international journal of advanced computer technology, 3 (2), February-2014 Volume-III, Issue-II.
13. Pedro Serrador(2015), Does Agile work? - A quantitative analysis of agile project success, International Journal of Project Management, Elsevier, Vol33 (5), pp. 1040–1051.
14. Kumar, A., Vengatesan, K., Rajesh, M., Singhal, A. 57205678507; 55611316200;56606891300; 24765540900; Teaching literacy through animation & multimedia (2019) International Journal of Innovative Technology and Exploring Engineering, 8 (5), pp. 73-76.
15. S. Ontan˜on, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss: "RTS AI: Problems and Techniques".
16. OpenAI Five: June 28,2018 (https://blog.openai.com/openai-five/).
17. Abhishek Kumar, Achintya Singhal, Jitendra Sheetlani "ESSENTIAL-REPLICA FOR FACE DETECTION IN THE LARGE APPEARANCE VARIATIONS", International Journal of Pure and Applied Mathematics, volume 118, Issue 20, Pages: 2665-2674
18. Deep Reinforcement Learning with OpenAI Gym: April 27,2016 (https://ai.intel.com/openai).
19. Selvaraj Kesavan, E. Saravana Kumar, Abhishek Kumar & K. Vengatesan (2019) An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services, International Journal of Computers and Applications, DOI:10.1080/1206212X.2019.1575034
20. Richard Kemp(2010), Open source software (OSS) governance in the organisation Review, Elsevier, Volume, May 2010, Pages 309-316.
21. Schryen, G. (2011). Is open source security a myth?, Communications of the ACM, 54(5), 130-140.
22. Sen, R., Singh, S.S., Borle, S., 2012. Open Source software success: measures and analysis. Decision Support Systems, Elsevier, 52 (2), 364–372.
23. Silic, M., & Back, A. (2013). Information security and open source dual use security software: trust paradox Open Source Software: Quality Verification, Springer, page 194-206.
24. Subhas C. Misra and Virender Singh,(2013), Conceptualizing open agile software development life cycle (OASDLC) model, International Journal of Quality & Reliability Management, ©Emerald Group Publishing Limited, Vol. 32 No. 3, 2015 pp. 214-235.
25. Wang, J.,(2012), Survival factors for Free Open Source Software projects: a multi-stage perspective, European Management Journal, Elsevier ,30 (4), 352–371.

Authors Profile

Mr. Anand Kumar Rai completed his M.Sc.(Computer Science)degree in 1999 from Gurughasi Das University, Bilaspur,CG.He has Completed his M.Phil(Computer Science)in 2008 from Alagappa University,Karaikudi, Tamilnadu. He has been serving as an Assistant Professor in Mumtaz PG College since 2000. He has more than 10 publications in reputed national and international journals, his subject specialization in software engineering and artificial intelligence.Now he is doing his research work on the topic" Risk minimization plan for agile software using fuzzy logic" from Shri Ramswaroop memorial University, Deva Road ,Barabanki.

Dr. Shalini Agrawal completed BSc(Mathematics) in 1998 from University of Allahabad and DOEACC A Level Certification in 2001. She also completed MSc(Computer Science) in 2005 from J.K Institute of Applied Physics and Technology, University of Allahabad and M.Tech(Computer Science & Engineering) in 2010 from Amity University. She is a PhD from Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow in 2015. Her area of specialization is Cloud Computing, High Performance Computing and Software Engineering. She joined Sri Ramswaroop Memorial University as HoD of Department of Computer Science and Engineering, in the year 2015. She has also guided two M.Tech Projects and presently guiding two PhD students in their research work. She is also the chairperson of Digital Monitoring Cell of the University where she is taking initiatives to develop MOOC courses and streamline other University processes in accordance with Smart Campus directives of MHRD. She is a professional member of IEEE and CSI.

Dr. Mazhar Khaliq did M.C.A. from Aligarh Muslim University and got Ph.D.(Computer Science) from Integral University, Lucknow. He has more than 20 years of teaching experiences. Now he is an Assistant Professor & Subject In-charge Department of Computer Science, Khwaja Moinuddin Chishti Urdu, Arabi-Farsi University. He has more than 15 national and international publications.

Dr. Abhishek Kumaris Assistant Professor, Computer Science at Banaras Hindu University.He is Apple Certified Associate, Autodesk Certified & Adobe Certified Educator. His research interests is Stereoscopy, 3D Animation, Image Processing, VR & AR, Multimedia, Game Technology, MOOCs, Graphics & Visual Effects.