Sādhanā (2019) 44:166
https://doi.org/10.1007/s12046-019-1150-9

© Indian Academy of Sciences

Distributed cat modeling based agile framework for software development

B PRAKASH* and V VISWANATHAN

Vellore Institute of Technology, Chennai 600 127, India
e-mail: prakash.bala@vit.ac.in; viswanathan.v@vit.ac.in

MS received 1 August 2018; revised 21 December 2018; accepted 1 May 2019; published online 24 June 2019

Abstract. Software development is a challenging process that requires in-depth understanding and an effective model such that the developed software inherits good quality and reliability, and attains customer satisfaction towards achieving the goals successfully. The effectiveness of the software is enabled by modifying the operating modules of the software through a model, like agility. In this paper, the catastrophic and distributed computing models are integrated into the software development process. The proposed model is termed as a distributed cat model that is developed with the aim to handle the risk factors engaged in various developing stages of the agile model. The risk factors that affect the communication, planning, release, design, coding and testing modules of the agile modules are deeply learned and executed such that the risk factors are tackled by various modules present in the proposed distributed cat model. The effectiveness of the proposed model is analysed based on the performance metrics such as Index of Integration (IoI) and Usability Goals Achievement Metric (UGAM), for which five products, including the hotel management system, Customer Relationship Management system (CRM), rainfall prediction system, temperature monitoring system and meta-search system, are employed. The analysis is performed using the parameters like mean difference, variance, standard deviation and correlation coefficient. The result proves that the proposed model offers a great positive deviation contributing to high degree of performance in software development.

Keywords. Software development; catastrophic model; distributed computing; agile model; software quality.

1. Introduction

The growth in scientific research and development in the field of software development seems to be increasingly reported [1]. Agile Software Development (ASD) [2] is a significant methodology that was developed in the late 1990s to meet the uncertainty issues related to the requirements of the customer, evolution of technology and inconsistent business environments. The methods used in ASD have neglected the highly formalized thought process followed in the present, which remains dynamic, user-centric, continuous releases and so on [3]. There are a large number of unique characteristics in ASD that have an impact on the decision-making process such as fast-paced incremental and iterative nature, and one relies on self-motivation and management [4]. Moreover, to differentiate the features for effective decision-making, ASD gains remarkable significance [5]. The basic assumptions made in the agile development pose a massive challenge while applying it practically in large-scale projects. Large-scale programmes possess a huge number of stakeholders and customers of the product, and their requirements are transformed to the developers, which in turn intensify the problems due to large customer involvement [6].

Distributed Agile Development (DAD) aims at the speed and quality of agile process. Also, it exhibits the cost benefits involved in Distributed Software Development (DSD), but it injects remarkable demerits as a result of the deviation in their respective key tenets. Agile methods ensure communication between face-to-face frequently and ensure trust. The distance factor in DSD highlights various organizational standards, cultures and policies, causing the minimization in the team cohesion [7–9]. Risks are distinctive to the projects and experience in the field of software development with respect to the traditional development approach [10]. There are five categories of the risks – ‘Project Management,’ ‘Software Development Life Cycle (SDLC),’ ‘External Stakeholder Collaboration,’ ‘Group Awareness’ and ‘Technology Set-up,’ – which are mapped to the Leavitt model for betterment of organizational changes. The idea behind mapping is that it facilitates us to identify the organizational aspects that manage the risks buried in DAD projects [8, 11, 12]. In contrast, ASD makes use of an iterative approach for constructing the software and intends to minimize the time taken for development, prioritizing value. At the same time, it

*For correspondence

ensures the quality of the software and thereby minimizes risks [12].

The traditional, plan-driven product and project management models do not suit the organizations of agile development, in which the scoping decisions are frequently made, and implementation is carried together with the requirements engineering process [13]. The efficient progress of the organizational growth towards achieving their objective cannot be achieved if the management process does not abide by the agile development [14]. The intra-team knowledge sharing in ASD gains significance in accomplishing tasks on a specific project and provides an opportunity for exploring the creativity to ensure the competitiveness of the organization. The agile culture assures a natural environment for intra-team knowledge sharing through agile practices and minimal documentation [15]. Therefore, agile methods fall on the same line of the strategies employed in KM personalization [16]. It is known that there is no common process suitable for all projects, emphasizing the fact that all the methods in agile and non-agile approaches are required to be tailored and combined to assist multiple projects [17]. The fact mentioned earlier seeks attention regarding the software development capability that integrates the agile and traditional components to develop a hybrid software development method to overcome the challenges and to apply agile approaches for large projects [18]. Hence, using agility on a large scale needs the merging of both the agile and the non-agile components that architect the context-aware hybrid adaptive methodologies [19].

This paper proposes a modified agile model for which the distributed concept and the catastrophic model are fused. The main objective of the modified agile model is to design and develop software with less complexity and time. This study presents the architecture of the agile model and deliberates the risk factors involved in developing agile projects. The risk factors at different stages of the agile model are based upon various factors, and the proposed model aims at capturing all the risk factors listed in the model.

The major contribution of this study is developing an agile framework based on distributed cat modelling. The proposed model is the integration of catastrophic and distributed computing model that aims at modelling and designing a better software development process. The proposed model exhibits the qualities such as high reliability and consumes less time and cost for developing software.

The paper is organized as follows. Section 1 gives the background of the software development process, section 2 deliberates the motivation of the work, section 3 explains the agile framework and section 4 illustrates the risk factors involved in the agile development process. The proposed methodology is discussed in section 5. The analysis and discussion are provided in section 6, and finally, section 7 concludes the paper.

2. Motivation

2.1 *Related work*

Kulkarni and Padmanabham [1] proposed artificial intelligence activities, in which the individual characteristics and correlation behaviour among the products were identified. This method suffers from a great issue as it possesses a low value of correlation for the Index of Integration (IoI) metrics. Gill *et al* [19] proposed a qualitative constructive, empirical research approach that can be used as a checklist or vision-guiding reference architecture. The demerit of the method is that analysis, coding and labelling of context-aware hybrid adaptive methodology reference architecture (HAMRA) elements are subjected to human error and mistakes, which may lead to inconsistencies. Heikkilä *et al* [14] designed a model for Ericsson telecommunications, in which the agile development methods were successfully employed in organizations, where the higher level planning processes were not agile. The challenge is that the process of combining agile methods with a flexible feature development process can bring many benefits, but large-scale software development seems to require specialist roles and significant co-ordination effort. Shrivastava and Rathod [10] proposed risk management framework intended for DAD and presents frequently used methods to reduce their impact. The demerit of the method is that it does not support multiple vendor involvements in distributed agile projects. Santos *et al* [16] introduced a method termed as the fostering effective inter-team knowledge sharing. The advantage is that the effectiveness of the method is related to applying purposeful practices, along with organizational conditions and stimuli. The drawback of the method is that it lacks an organizational structure. Shrivastava and Rathod [8] addressed the specific risks in DAD environment. It possesses the advantage that when the risk factors and their causes relevant to scientific and formal models are explored, it addresses the risks in DAD projects. However, DAD projects offered substantial risks due to contradiction between distributed development and agile practices. Dingsoyr *et al* [6] proposed an agile method adaptation that adopted to scale, and the context of development programmes influences the scaling approach. However, large size data influence the identification of risks. Satapathy and Rath [20] proposed a story point approach that quantified the size of the user story, but the number of records available for testing purpose is very less. Hence, obtaining optimal results by considering this small dataset is very difficult to be ensured.

2.2 *Challenges*

The various challenges involved in the agile framework strategy are discussed here.

- The concentration of the knowledge sharing is on the tacit knowledge available in the agile methods, and this remains a drastic challenge in case of the presence of a large number of participants in the project. Large-scale projects possess a number of customers and stakeholders of the product, and the requirements of the product are to be informed to the developers. This again raises the challenges of customer involvement [6].

- The unique characteristics belonging to the requirements of the software tend to increase the dimensionality of the relevant datasets, make them sparse and more often lead to ambiguous expressions. The afore-explained strategy increases the challenges on the standard processing techniques [21].
- ASD is linked to improved decision-making, and the existing theories concentrate on the precise strategies of decision-making in such environments. The clarity regarding the establishment and evaluation of the myriad of decisions from software feature inception to product delivery and refinement is lacking [5].
- The standard or common method of risk management process and tool are not available in software development situations [12].

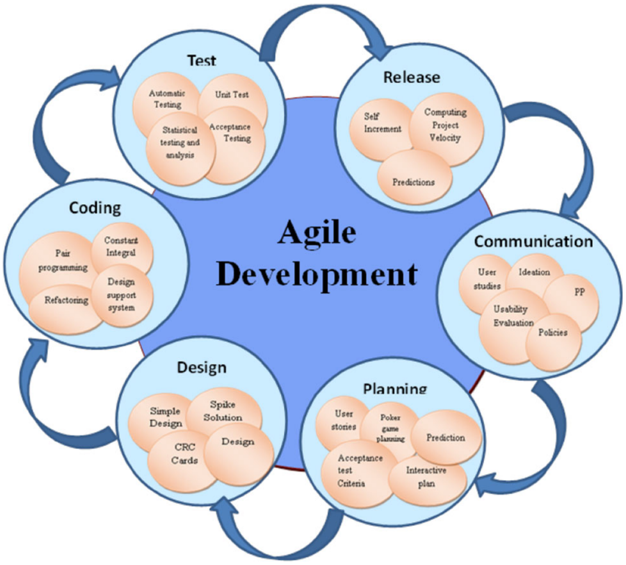

Figure 1. Architecture of agile development process.

3. Agile framework

The main aim of developing the software relies on the software qualities, including the reliability, availability, maintainability, cost, time and so on, which in turn depends on software requirements, design, development and testing. The agile model [1] plays a major role in modelling and documenting the software systems depending on their best practices that are based on certain principles and ideas. Figure 1 shows the architecture of the agile model, which comprises two modules: communication and iterative modules. The iterative module is formulated as the following steps that include the release, planning, design, test and coding.

Communication: Communication [1] is the first step, which comprises steps like user studies, ideation, usability evaluation and policies. The communication phase possesses the policies that shape the products and their evaluation. The steps involved in communication aim at defining the product based on the policies. Thus, policies play a major role in decision-making.

Iterative phase: The iterative phase [1] consists of five major steps that are repeated for a defined period until the required output is derived. The five major steps in the iterative phase are planning, design, coding, testing and release.

4. Risk factors in software development stage using agile framework

Figure 2 shows the major risk factors involved at various steps of the software development process. The factors that impact the communication are based on the channel selection, number of persons, hierarchy-based factors, data confidentiality and distance-based factors.

Communication: The communication phase deliberates the communication parameters for achieving the best outcome. Agile teams require communicating among themselves and with their customers, frequently, and hence, language seems to be an open problem during direct communications. Moreover, the socio-cultural differences cause bad effects on the agile development processes.

Channel selection: The risk factors influencing the channel selection include direct communication and indirect communication. In the direct communication the contents addressed to the developers are direct, yet they suffer from the language issues and sometimes they are vague. Meanwhile, in the indirect communication, communication occurs either through any of the communication models like phones, e-mails and so on. The problem is that the content communicated to the team may not reach in a proper way, affecting the developing cycle of an agile model.

Data confidentiality: The data regarding the organization, person, policies and so on, should be secured to ensure the successful implementation of the software. The policies and the principles of the development team vary among projects and hence, assuring the confidentiality of the data is a major issue. Providing protection should be activated not only to the data of the project but also to all the documents of the organization.

Distance: Distance in communication is a major issue that may be affected due to the global dialects, cultural distances and eventual conflicts. These issues may disrupt the discussion session, leading to misunderstandings regarding the project.

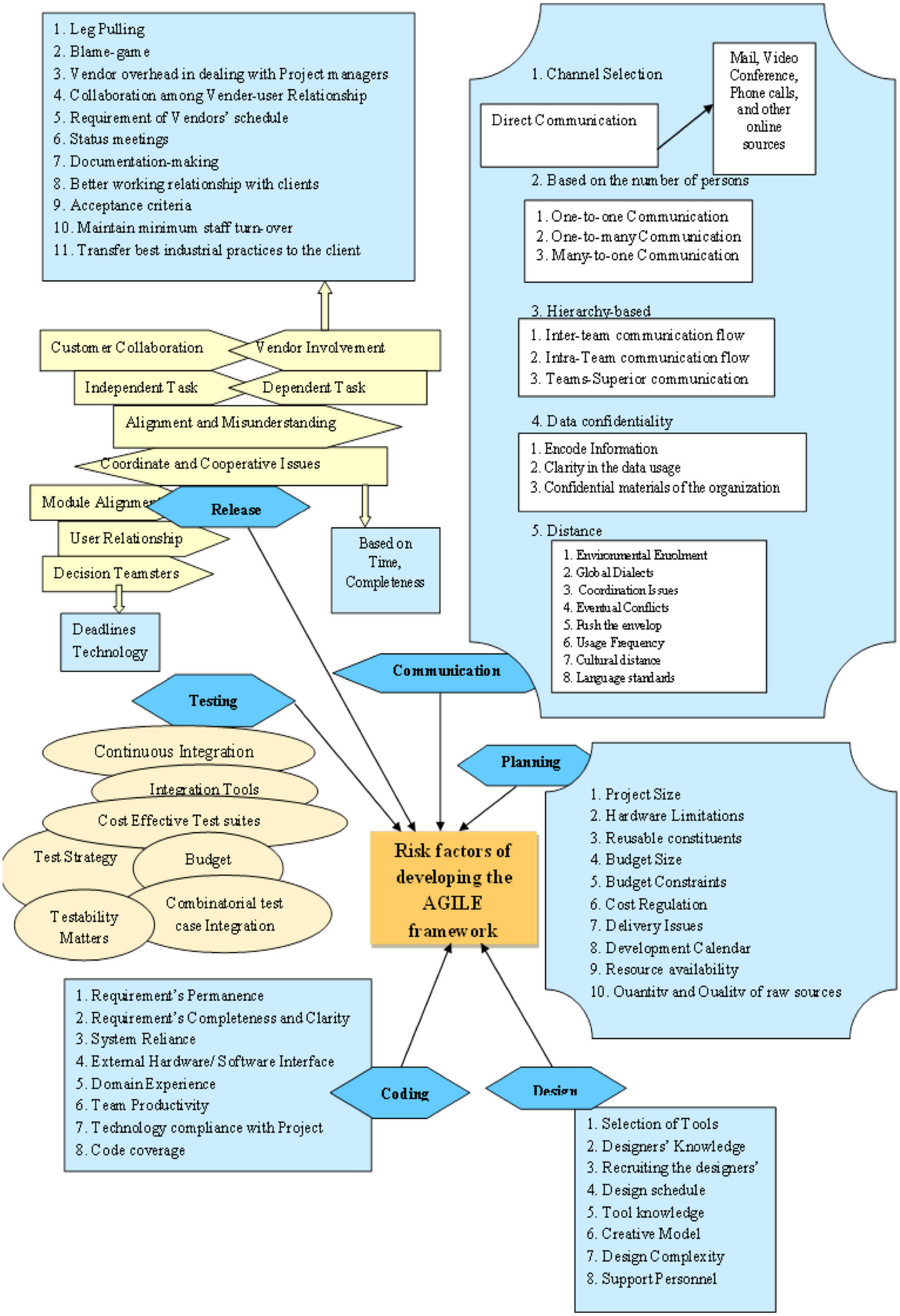

Figure 2. Risk factors in the development process of the software.

Hierarchy-based communication: Communication occurs among the members of the development team or among the developing teams or between the teams and the superior persons, such as project managers, team leads and so on.

Communication based on the number of persons: Communication regarding the project may include one or more number of persons attending the discussion. In the discussion regarding the product, the details of the individuals are communicated to the individual persons in a flow. For instance, the project manager communicates the information to the team lead and the information is passed down to his team members, and this information is communicated to the lower level to achieve the task. This flow of communication involves a single individual to a number of persons involved in the team.

Planning: The planning [1] section comprises the team's plans to achieve the goals of the project. The plans can be based on the long-term goals and short-term goals. The long-term goals are based on the mission, goals, policies and objectives of the organization, whereas the short-term goals concentrate on the rules, procedures, programs and training that focuses on achieving the long-term goals.

Design: The project design is based on the tools employed for the project, and the operator's knowledge in the domain and regarding the tool. Moreover, the designer should be creative in modelling the project without complexity in dealing with it.

Coding: Coding is the major criterion for developing the project, and the factors that affect the coding process include the system reliance, experience of the person in a particular domain and their expertise.

Testing: Testing is the major area that deliberates the success degree of the software, and it enables us to finalize the quality of the software. In testing, the testing strategy follows standard testing matters as it should tally with the time and the cost. In testing, strategies affect the progress of the software in achieving the goal. Moreover, the software development team suffers a lot in deciding the optimal and effective test suites that ensure effective testing of the software.

Release: The release [1] step pictures the risk factors at the time of delivering the software product. The success of the product is based on the relationship between the client, and mainly on the active vendor involvement. The relationship between the vendor and the client is the major risk factor, and the turn-off of the team staff can cause a major issue on the delivering side of the software product. Moreover, if there is no customer collaboration, the developed software tends to fail, and it occurs only when the customer demands are not completely nullified. Also, the technology concerned should be an effective means that should coincide with the objective of the software and the software developed should assure completeness to the client.

5. Proposed framework of agile process using a distributed cat model

The proposed framework of the agile model employed the catastrophic model [22–24] and distributed computing in the agile framework. The process of using the catastrophic model along with the distributed computing is to tackle the risk factors, enhance the performance of the software product and minimize the losses. The risk factors of the agile model for designing the software are presented in figure 2, and the risk factors are tackled using the catastrophic model that is highlighted in figure 3.

The catastrophic model promotes disaster management and the modules in the catastrophic model are spatial event module, vulnerability module, accounting risk model, Adaptive Monte Carlo Optimization Model and Stochastic Optimization Model. The distributed computing factors include the MapReduce framework, parallel scheduling and cluster architecture that manage the onshore projects. Thus, the proposed agile framework based on distributed cat model is used by software companies in developing the software effectively. Upon developing the model, the company acquires feedback from the company personnel and from the user side to prove the effectiveness of the model, and hence to improve the features.

5.1 *Catastrophic model*

The modules used in the catastrophic model [22] for handling the risk factors are depicted in figure 4, and the discussion is deliberated here.

Adaptive Monte Carlo Optimization Model: The model ensures effective decision-making by overcoming destructive plans and decisions. It offers an effective solution in combining the various modules of the product effectively without any loss in the budget.

Vulnerability module: The vulnerability module finds and solves the spatial patterns of the released factors that contribute to the economic losses of the company. This module estimates the direct losses and adds all the possible cascading effects that affect the failure of the developed software product.

Accounting risk module: The accounting risk module tackles the risk factors affecting the product budget such that it brings perfection to the company and the team. It ensures the budget of the product to be complete over time, and it ensures high quality and quantity for bringing a perfect product.

Spatial event module: The spatial event module tackles the problems associated with software development in terms of the distance in teams, persons, technologies and so on. The lack of the tool knowledge employed for performing the software development process and missing the deadlines causes a huge problem on the customer side. As the lack of sufficient knowledge consumes more time,

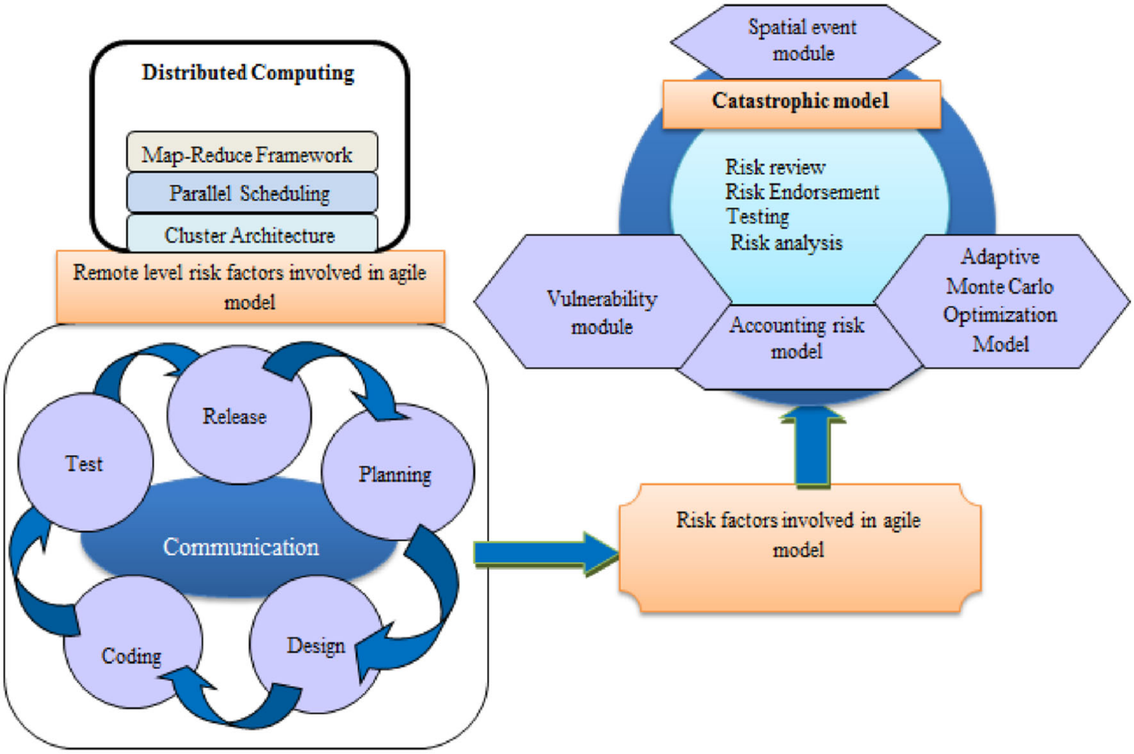

Figure 3. The proposed distributed Cat modeling-based agile framework for software development.

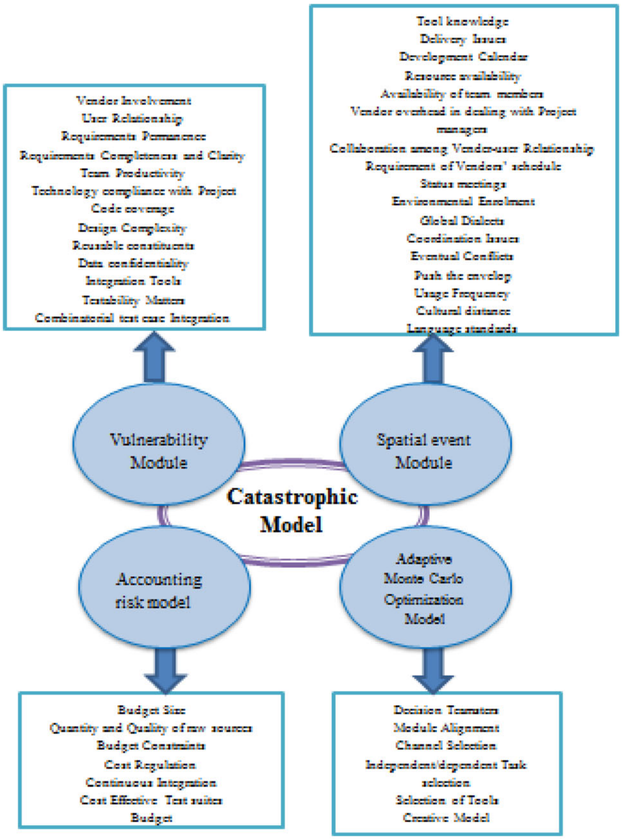

Figure 4. Risk factor handling using the proposed cat-model-based agile framework.

sufficient training should be offered to the layman engaged in the development process.

5.2 *Distributed model*

The distributed model with the MapReduce framework, parallel scheduling and cluster architecture ensures effective planning and a platform to tackle the problems associated with the agile model. The distributed modules employed in the organizations assure a greater degree of flexibility and offer a better solution to the users based on their business requirements. The modules in the distributed scenario assure strong social interaction and compensate for the delays in presenting the requested application.

MapReduce Framework: The MapReduce framework is the best solution that coordinates the individuals at various geographical locations engaged in the developing stage of the software.

Parallel scheduling: Parallel scheduling is activated when multiple users across the globe use the same product or the software for multiple purposes simultaneously. The teams at various geographical locations are coordinated simultaneously and enhance the knowledge sharing among them.

Cluster architecture: The clustering phenomenon is highly significant as it tends to highlight the operating status of a team or personnel in a particular domain with high degree of information sharing among them such that the effective development stage is evoked.

Table 1. Attributes of the product.

Products	Lines of code (LOC)	Effort (no. of persons)	Duration (months)
Hotel management system	80,000	8	7
CRM system	94,650	15	10
Rainfall prediction system	97,325	14	11
Temperature monitoring system	83,528	10	8
Meta-search system	98,580	16	12

The risk factors of the agile model are managed effectively using the distributed cat model and it offers a new production environment for the software.

6. Analysis and discussion of the proposed agile model

This section presents the results and discussion of the proposed model of the software development.

6.1 *Scale of the product*

The performance of the proposed model is evaluated using five products: hotel management system, Customer Relationship Management (CRM) system, rainfall prediction system, temperature monitoring system and meta-search system. Table 1 depicts the product attribute such as lines of code, number of persons (effort) and months of each product.

6.2 *Measuring the effectiveness of integration*

The proposed model is evaluated using the five products that include the hotel management system, CRM system, rainfall prediction system, temperature monitoring system and meta-search system. The hotel management system monitors the workers and the entire operating environment of the hotel. The CRM system product is based on the various services of the customer and manages the customer and vendor management. The rainfall prediction system predicts the rainfall of the particular area or locality in the future or current time to plan the schedules, like agriculture, picnic and so on. A temperature monitoring system monitors the temperature of the system and alarms in the presence of vulnerabilities. Finally, the meta-search system is the optimizing web searching software that combines the multiple search engines. The proposed model is applied to the five products from the first step of communication to the deliverystage. For the evaluation of the performance of the proposed model using IoI, 25 programmers with a masters degree in Computer Science are employed, and the evaluation of the Usability Goals Achievement Metric (UGAM) score is done using 25 students from various disciples. Among the 25 students employed for evaluating the performance of UGAM, 10 students are from the Engineering fields, ten students are from Arts and the remaining five are from other fields.

6.3 *Performance metrics employed*

The metrics used for the analysis include the IoI and UGAM [25]. UGAM is a metric that measures the quality of the software based on the experience of the user. UGAM is a product metric with a scale range of 0–100, in which 100 specifies the best user experience possible and 0 indicates the worst experience of the user. UGAM contains the goal parameters, such as goals, goal parameters, score and weight. The UGAM parameter is given as

$$UGAM = \frac{\sum_{\ell=1}^{g} W_\ell * A_\ell}{\sum_{\ell=1}^{g} W_\ell} \tag{1}$$

where A_ℓ symbolizes the goal parameter, W_ℓ denotes the weight of the goal parameter and $g$denotes the total number of the goal parameters. The scoring parameter takes values between 0 and 100, where 100 indicates the best possible user experience and 0 signifies the worst possible user experience, and the weight takes values between 0 and 5. The attributes belonging to UGAM is explained briefly in [1]. The IoI metric comprises three parameters: weight, score and activity evaluation guidelines. IoI metric is given as

$$IoI = \sum_{\ell=1}^{g} \left(\frac{W_\ell * A_\ell}{\sum_{\ell=1}^{g} W_\ell} \right). \tag{2}$$

The weights are assigned by the experts of the organization and by the group of the reviewing team. The attributes of IoI are deliberated in [1].

6.4 *Sample values of the UGAM and IoI score distribution*

The evaluation of the proposed model is carried out using the IoI and UGAM model; the IoI metric is associated with

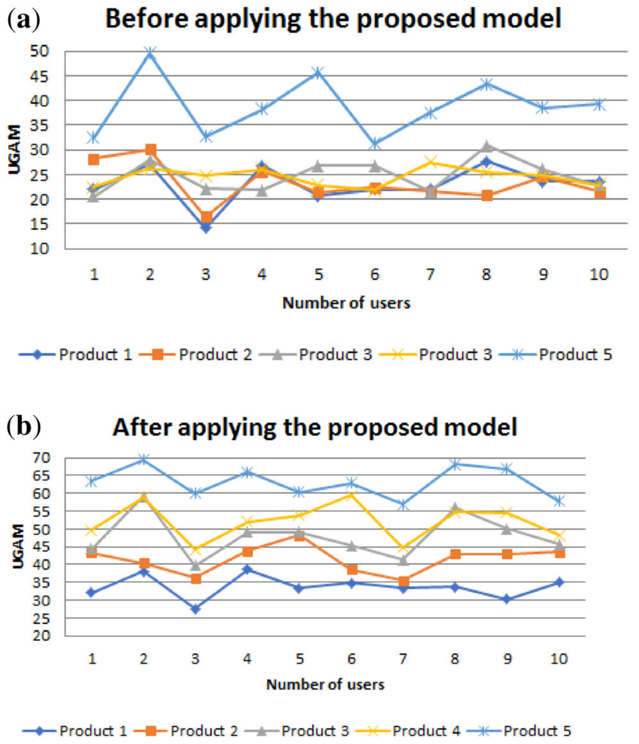

Figure 5. (**a**) UGAM before applying proposed model. Figure 5. (**b**) UGAM after applying proposed model.

the software development, and hence the metric is allocated to the team of the developers that generate the attribute values. The attributes relevant to IoI and UGAM are collected from 25 developers and students, and the sample values are shown in figures 5 and 6, respectively. Figure 5(a) and (b) shows the attribute values of UGAM before applying the proposed model and after applying the proposed model. Similarly, figure 6(a) and (b) shows the attribute values of IoI before and after applying the proposed model.

6.5 *Correlation analysis of user and employee experience*

Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two variables (user and the employee). Covariance analysis is used to find the variation in the intention of the user to buy a particular brand forvarious levels of price and the user's attitude towards that product. It is also used to find how a change in the price level of a product will affect the consumption of that product by the users. Hence, these measures are used to evaluate the effectiveness of the proposed model.

Table 2 shows the correlation analysis based on the experience of the user and the employee of the releasing organization. The correlation and covariance of the user and the employee are better upon the utilization of the proposed model when compared with the existing model. The review is conducted for five products, and the correlation and the covariance of the product 1 before the application of the proposed model are0.4889 and 16.3116, whereas the correlation and covariance of the product 1 after applying the proposed model is 0.7074 and 18.5385 respectively. Product 2 has a correlation of 0.2548 and covariance of 8.3382 before applying the proposed model, whereas UGAM acquires a value of 0.1144 and 2.2766 after applying the proposed model. The product 4 acquires the negative correlation before applying the proposed model and positive correlation after applying the proposed model.

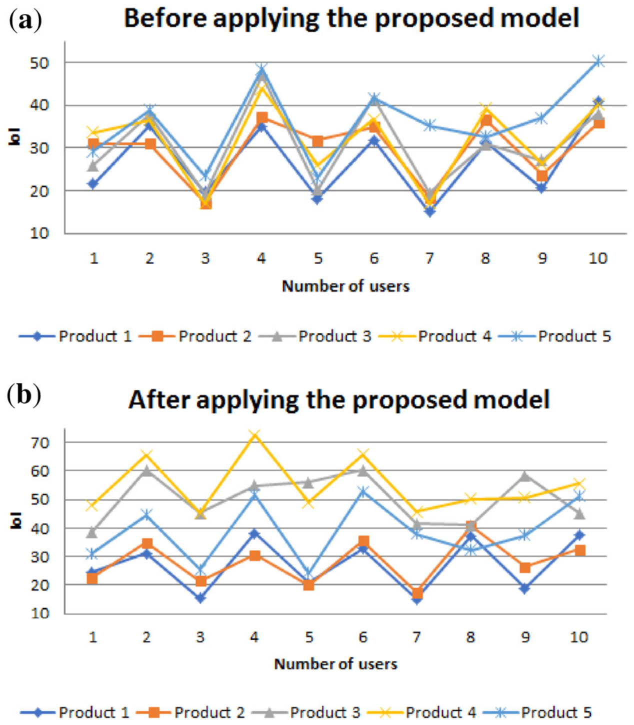

Figure 6. (**a**) IoI before applying proposed model. Figure 6. (**b**) IoI after applying proposed model.

6.6 *Correlation analysis of products improvement*

Table 3 shows the correlation analysis among the products before and after applying the proposed model. The product 1, 2, 3, 4 and 5 acquired the correlation value of 0.7215, 0.5182, 0.2150, 0.4391 and 0.5501, respectively, before applying the proposed agile model, whereas they acquired a correlation value of 0.5236, 0.1275, 0.3734, −0.2174 and 0.1933, respectively, after applying the proposed model in the development process. The product 1, 2, 3, 4 and 5 acquired the covariance value of 61.6737, 31.1278, 30.1968, 56.7620 and 82.9098, respectively, before applying the proposed agile model, whereas they acquired a correlation value of 7.1728, 2.7141, 11.4243, −1.1376 and

Table 2. Correlation analysis of user and employee experience.

	Metrics	Product 1	Product 2	Product 3	Product 4	Product 5
Before applying the proposed model to IoI and UGAM	Correlation analysis	0.4889	0.2548	0.1283	−0.1285	0.0048
	Covariance analysis	16.3116	8.3382	2.5024	−1.2211	0.2419
After applying the proposed model to IoI and UGAM	Correlation analysis	0.7074	0.1144	0.2048	0.3972	0.1185
	Covariance analysis	18.5385	2.2766	17.9701	24.6060	5.0157

Table 3. Correlation analysis among products.

	Metrics	Product 1	Product 2	Product 3	Product 4	Product 5
Before and after applying the proposed model to IoI	Correlation analysis	0.7215	0.5182	0.2150	0.4391	0.6501
	Covariance analysis	61.6737	31.1278	30.1968	56.7620	82.9098
Before and after applying the proposed model to UGAM	Correlation analysis	0.5236	0.1275	0.3734	−0.2174	0.1933
	Covariance analysis	7.1728	2.7141	11.4243	−1.1376	6.1290

6.1290, respectively, after applying the proposed model in the development process.

6.7 *Performance deviation analysis*

The performance deviation analysis is dispalyed in table 4, which shows the deviation parameters based on the metric before and after applying the proposed model. The product 1, 2, 3, 4 and 5 acquired the mean difference of −3.0357, 8.2697, −12.7217, −14.6552 and −5.7142, respectively, with respect to IoI before and after applying the proposed agile model; meanwhile, they acquired a mean difference value of −10.9828, −10.431, −23.7931, −27.3276 and −30.9483, respectively, with respect to UGAM before and after applying the proposed model in the development process. The product 1, 2, 3, 4 and 5 acquired the variance of 78.2904, 55.044, 182.2056, 230.4038 and 97.966, with respect to IoI before and after applying the proposed agile model, whereas they acquired a variance of 42.8041, 102.8344, 164.7798, 214.3561 and 181.6505, respectively, with UGAM before and after applying the proposed model in the development process. Similarly, the product 1, 2, 3, 4 and 5 acquired the standard deviation of 8.8481, 7.4192, 13.4983, 15.1790 and 9.8978, respectively, with IoI before and after applying the proposed agile model, whereas they acquired a standard deviation of 6.5424, 10.1470, 12.8366, 14.6409 and 13.4777, respectively, using UGAM before and after applying the proposed model in the development process.

Table 4. Performance deviation analysis.

	Metrics	Product 1	Product 2	Product 3	Product 4	Product 5
Before and after applying the proposed model to IoI	Mean difference	−3.03571	8.2697	−12.7217	−14.6552	−5.7142
	Variance	78.2904	55.0447	182.2056	230.4038	97.9666
	Standard deviation	8.8481	7.4192	13.4983	15.1790	9.8978
Before and After applying the proposed model to UGAM	Mean difference	−10.9828	−10.431	−23.7931	−27.3276	−30.9483
	Variance	42.8041	102.8344	164.7798	214.3561	181.6505
	Standard deviation	6.5424	10.1470	12.8366	14.6409	13.4777

Table 5. Performance forecasting using the metrics.

Metrics		P1	P2	P3	P4	P5	Forecasted
Before applying proposed model	UGAM	20.1969	21.8448	22.1069	24.5691	39.97414	45.5188
	IoI	24.0703	25.1732	28.1675	30.6237	34.3552	40.1002
After applying proposed model	UGAM	30.0172	39.0345	41.4414	47.1344	59.1639	74.3553
	IoI	24.6564	27.6869	40.8115	49.9963	54.9963	59.0674

6.8 *Performance forecasting*

The performance forecasting is illustrated in table 5, which provides the analysis based on the performance metrics. The forecast values before and after applying the proposed model are found to be 45.5188, 40.1002, 74.3553 and 59.0674, respectively, which are computedwith respect to the past values. Initially, an organization releases five products, and at the release of product 1 before applying the proposed model, the UGAM and IoI values are 20.1969 and 24.0703, respectively. After applying the proposed model for the software development process, the UGAM and IoI values of product 1 are 30.0172 and 25.6564, respectively. The application of the proposed model to the software development step increases the performance as is shown in table 5. Similarly, the IoI values of the products after applying the proposed model are increased when compared with the IoI value before applying the proposed model in the software development stage.

7. Conclusion

The modified agile model using the catastrophic model and distributed computing facilitates the effective modelling of software in terms of cost and application. The proposed model enhances the operating condition of the software as per the application and meets the customer demands. Initially, to develop the proposed agile model, the architecture of the standard agile model is studied, the risk factors are explored, and the causes and the effects of the risk factors are listed briefly. The listed risk factors are analysed to understand the bad effects such that the proposed distributed cat modelling aims at tackling the adverse effects of the risk factors affecting the development stages of the software. The proposed model is tuned particularly to tackle the listed threats, and the frameworks working at managing the risk factors include the MapReduce framework, parallel computing and the cluster architecture in the distributed computing framework and the several modules in the catastrophic model. The proposed model is analysed to prove the effectiveness of the model using IoI and UGAM metrics; the calculation in terms of correlation proves that the proposed model results in an improvement in products. The proposed model is effective in improving the product quality because it takes the advantages of both the catastrophic and distributed computing models. The catastrophic model lowers the chances of financial losses and minimizes the agency costs. The distributed computing model enhances the robustness and reliability of the overall system, reaching near-optimal or optimal control. The analysis further proves that both the practitioners and the users attained better satisfaction level with the proposed model as compared with the existing agile model.

References

[1] Kulkarni R H and Padmanabham P 2017 Integration of artificial intelligence activities in software development processes and measuring the effectiveness of integration. *IET Softw.* 11(1): 18–26

[2] Parsons D, Susnjak T and Lange M 2014 Influences on regression testing strategies in agile software development environments. *Softw. Qual. J.* 22(4): 717–739

[3] Austin R D and Devin L 2009 Research commentary—weighing the benefits and costs of flexibility in making software: toward a contingency theory of the determinants of development process design. *Inf. Syst. Res.* 20(3): 462–477

[4] Highsmith J 2017 *Agile Project Management*. Boston, MA: Addison-Wesley

[5] Drury-Grogan M L, Conboy K and Acton T 2017 Examining decision characteristics & challenges for agile software development. *J. Syst. Softw.* 131: 248–265

[6] Dingsøyr T, Moe N B, Fægri T E and Seim E A 2018 Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empir. Softw. Eng.* 23(1): 490–520

[7] Jalali S and Wohlin C 2010 Agile practices in global software engineering – a systematic map. In: *Proceedings of the 5th IEEE International Conference on Global Software Engineering*, Princeton, NJ, USA, pp. 45–54

[8] Shrivastava S V and Rathod U 2017 A risk management framework for distributed agile projects. *Inf. Softw. Technol.* 85: 1–15

[9] Almeida E S, Alvaro A, Lucredio D, Prado A F and Trevelin L C 2004 Distributed component-based software development: an incremental approach. In: *Proceedings of the 28th Annual International Computer Software and Applications Conference*, China

[10] Shrivastava S V and Rathod U 2015 Categorization of risk factors for distributed agile projects. *Inf. Softw. Technol.* 58: 373–387

[11] Cicotti G 2017 An evidence-based risk-oriented V-model methodology to develop ambient intelligent medical software. *J. Reliab. Intell. Environ.* 3(1): 41–53

[12] Odzaly E E, Greer D and Stewart D 2018 Agile risk management using software agents. *J. Ambient Intell. Humaniz. Comput.* 9(3): 1–19

[13] Jantunen S, Lehtola L, Gause D C, Dumdum U R and Barnes R J 2011 The challenge of release planning. In: *Proceedings of the Fifth International Workshop on Software Product Management* (IWSPM), Trento, pp. 36–45

[14] Heikkilä V T, Paasivaara M, Lasssenius C, Damian D and Engblom C 2017 Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. *Empir. Softw. Eng.* 22(6): 2892–2936

[15] Boden A, Avram G, Bannon L and Wulf V 2009 Knowledge management in distributed software development teams – does culture matter. In: *Proceedings of the Fourth IEEE International Conference on Global Software Engineering*, Limerick, pp. 18–27

[16] Santos V, Goldman A and Souza CRD 2015 Fostering effective inter-team knowledge sharing in agile software development. *Empir. Softw. Eng.* 20(4): 1006–1051

[17] Mahanti A 2006 Challenges in enterprise adoption of the agile methods – a survey. *J. Comput. Inf. Technol.* 14(3): 197–206

[18] Boehm B and Turner R 2003 *Balancing Agility and Discipline: A Guide for the Perplexed, Portable Documents*. Addison-Wesley Professional, Hoboken, NJ

[19] Gill A Q, Henderson-Sellers B and Niazi M 2016 Scaling for agility: a reference model for hybrid traditional-agile software development methodologies. *Inf. Syst. Front.* 20(2): 315–341

[20] Satapathy S M and Rath S K 2017 Empirical assessment of machine learning models for agile software development effort estimation using story points. *Innov. Syst. Softw. Eng.* 13(2–3): 191–200

[21] Belsis P, Koutoumanos A and Sgouropoulou C 2014 PBURC: a patterns-based, unsupervised requirements clustering framework for distributed agile software development. *Requir. Eng.* 19(2): 213–225

[22] Ermolieva T, Ermoliev Y, Fischer G and Galambos I 2003 The role of financial instruments in integrated catastrophic flood management. *Multinatl. Finance J.* 7(3–4): 207–230

[23] Yang Y, Lopez D, Tian H, Pouyanfar S, Fleites F C, Chen S and Hamid S 2015 Integrated execution framework for catastrophe modeling. In: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, Anaheim, CA, pp. 201–207

[24] Transform catastrophe risk management to improve product performance and reduce losses. From http://www.genpact.com/downloadable-content/insight/transform-catastrophe-risk-management-to-improve-product-performance-and-reduce-losses.pdf, accessed in October 2017

[25] Joshi A, Sarda N L and Tripathi S 2010 Measuring effectiveness of HCI integration in software development processes. *J. Syst. Softw.* 83(11): 2045–2058