Received November 13, 2017, accepted December 14, 2017, date of publication December 29, 2017, date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2787981

Improving the Quality of Software Development Process by Introducing a New Methodology–AZ-Model

MUHAMMAD AZEEM AKBAR[1,2], JUN SANG[1,2], ARIF ALI KHAN[3], (Member, IEEE), FAZAL-E-AMIN[4], NASRULLAH[1,2], MUHAMMAD SHAFIQ[5], SHAHID HUSSAIN[3], HAIBO HU[1,2], MANZOOR ELAHI[3], AND HONG XIANG[1,2]

[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society of Ministry of Education, Chongqing University, Chongqing 400044, China
[2]School of Software Engineering, Chongqing University, Chongqing 401331, China
[3]COMSATS Institute of Information Technology, Islamabad 54700, Pakistan
[4]Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11362, Saudi Arabia
[5]Chongqing University of posts and telecommunications, Chongqing 400065, China

Corresponding author: Jun Sang (jsang@cqu.edu.cn)

This work was supported by Chongqing Research Program of Basic Science and Frontier Technology under Grant cstc2017jcyjB0305.

ABSTRACT Quality is the most important factor for software development as it mainly defines customer satisfaction that is directly related to the success of a software project. The software process model is used to ensure software quality, represent a variety of task settings, manage project duration, improve the process and range to execute the process understanding, and to appropriate implicit conjecture for all task settings. Several software processes models exist in software albeit with limited scope. Given this viewpoint, this paper presents a new software development life cycle model, "AZ-Model," for software development by introducing new activities during software development life cycle. It overcomes the limitations of traditional models and significantly impacts the production of a quality product in a time-box. This paper also presents a comprehensive comparative study and statistical analyses to examine the significance of AZ–Model for software development.

INDEX TERMS Software development model (SDM), project management (PM), AZ-Model, software development life cycle (SDLC), six pointed star model, project management factors.

I. INTRODUCTION

Software Engineering is a growing and emerging field in the world since software makes life more comfortable. Figure 1 illustrates software engineering conceptions. The importance of software is undeniable. Specifically, in the present time frame, the fact remains that computers are indispensable in today's world due to their widespread use in almost every field of life, especially in commerce, industry, medicine, education, engineering, and agriculture.

A bird's eye view on the history of software engineering indicates that the era prior to and including the 1960s was considered as the functional era, the 1970s was the schedule era, 1980s was the cost era, while 1990s and beyond were considered as the efficiency and quality era. Given the excellent results of software development methodologies, software organizations are increasingly dependent on it, and thus, software is increasingly attractive. Hence, the importance of software engineering is rapidly promoted to control in-house production and for global business purposes. It is necessary for organizational business to grow and to meet institutional complexities due to the involvement of computer systems, and therefore, software quality has also become the keen user interest.

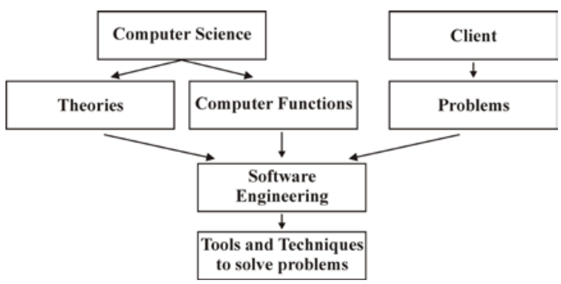

FIGURE 1. Explanation of software engineering conception [1].

The quality of the software mostly depends on the software development life cycle (SDLC). The SDLC is a route used by software development industry to design, develop, and test high-quality software. The aim of SDLC involves producing high-quality software that meets or exceeds customer expectations, reaches completion within time, and cost estimation, and is directly related to the customer as well as organizational satisfaction. It is a priority for every organization to adopt a low-cost software development model. If the low-cost model can effectively produce high-quality software, then it should be adopted to enjoy long-term benefits [2]. It is necessary for every organization to search for high-quality and low-cost software development models. Hence, it is considered that a good SDLC captures, verifies, and implements user requirements within the time-box and bought [3].

Many formal software development models are used although the specification is formal, and this is refined through several phases of design implementation [4].

A. EXISTING WELL-KNOWN MODELS

The waterfall model is the first, most influential, and most commonly used process model [5]. This model was recommended by Royce and includes a linear or sequential execution of stages in a way such that the previous phase provides feedback to the subsequent phase, and this typically follows the system design corresponding to the most significant process model [6]–[8].

In order to overcome the key limitations of the waterfall model, an iterative model of software development was introduced [9]. In this approach, requirements are collected, and the project is developed and delivered to the customer through iterations. Every delivered iteration is in addition to the already delivered iterations [10].

A Rational Unified Process model (RUP) was introduced with a parallel working style in which the new iteration commences prior to releasing the current iteration, and this is extremely time effective [11].

A spiral model is another example of the iterative model in development and from the delivery viewpoint. In the spiral model, prototyping and design elements are combined in a stage [12]. Four major phases are involved in this model as follows: i) objective, ii) risk, iii) development and validation, and iv) planning [13].

A very popular SDLC model that was termed as the V-Model was developed in 1980 [14]. This model involved an increased focus on testing to ensure the quality of the software, and even each phase of V-Model is associated with testing [13].

The above-stated methodologies are plan-driven and strong document-oriented methodologies. Hence, highly rigged methodologies should be considered. Owing to the reaction, a new concept of software development methodologies came into existence along with agile methodologies in 2001 [15]. These methodologies are very flexible and also known as lightweight methodologies [16]. Many lightweight methodologies exist, and a few of these are discussed in the present study.

Extreme programming (XP) is the most commonly used method in agile methodology and involves the advanced form of the problems encountered in long development cycles of traditional development models [17], [18]. The XP procedure is described by short development cycles, incremental arranging, constant feedback, dependence on communication, and a transformative outline [19].

The scrum methodology is an iterative and incremental process model to produce or manage any project. Essentially, the scrum is a term that originates from strategy in rugby [20]. It does not require or provide any specific software development method or practice that should be used by the scrum. The scrum only requires certain management practices and tools in different phases of scrum to avoid potential confusion due to unpredictability and complexity [21].

Various other approaches are introduced for project management and include Rapid Application Development (RAD) in 1991 by James Martin, Scrum (1995), Dynamic system development method (1995), crystal clear (mid-1990s), Feature-driven development (1997), Prince 2 project management approach in 1996, and Agile manifesto published in 2001. These are generic methods of project management adopted by different organizations based on projects requirements.

B. MOTIVATIONS

Based on the literature review, software development methodologies are broadly divided into two categories, namely heavyweight and lightweight methodologies. Both the methodologies are not yet satisfied because heavyweight methodologies are process-oriented, predictable, and less accepting of changes, while lightweight methodologies are people-oriented, adoptable, and easily accept changes in requirements. However, both methodologies are important for software development organizations.

Presently, global business is the key interest of every organization. Therefore, software development organizations currently conduct business all over the word for economic feasibility. Specifically, in developing countries, software development cost is extremely low and is almost one-third lower than that in developed countries [22], [23]. There are several other causes of outsourcing by software organizations such as availability of skilled human resources and reduced work load [24], [25]. Nevertheless, huge risks are involved in outsourcing and include progress incompatibility, coordination problem, cultural differences, and slightly hidden costs [23], [26], [27]. However, several solutions and causes exist with respect to outsourcing software projects [2], [28]. In this situation, heavyweight methodologies are most effective because there is no need to involve customers throughout the SDLC in heavyweight methodologies. Requirements are collected in the parent country and are handed over to the contractors of developing countries after analysis to implement the next required phases of the SDLC. In agile methodologies, informal communication is considered more valuable than formal communication, and thus, there is limited opportunity for agile methodologies [29], [30].

In order to improve the existing software development methodologies, several initiatives were recently adopted in different organizations, although they were not very successful. Extant studies indicated that various organizations do not use formal software development methodology and instead only use self-created methodology [31], [32]. The most important causes for the non-adoption of formal software development methodology include the unavailability of suitable, adjustable, and flexible methodology according to the organizations and project requirements [33]–[36]. Furthermore, a review of previous studies described the reasons for lack of adoptability of development methodologies as pertaining to the non-suitability for specific social and cultural characteristics with respect to development teams and organizations [34], [37]–[40].

- Thus, with respect to the adoptability of software methodology, it is necessary to introduce an innovative methodology that is flexible and adjustable based on the needs of both a development team and an organization that possesses both technical suitability for a project and social suitability for a development team.
- Hence, based on the requirements and to eliminate the limitations of heavyweight and lightweight methodologies, an innovative intermediate methodology termed as the "AZ-Model" was introduced with the capacity to overcome the existent gap and limitations of lightweight and heavyweight methodologies [41].

II. PROPOSED RESEARCH WORK

- The proposed study involves an improved version of AZ-Model after obtaining the opinion of experts and obtaining expertise after proper implementation for various sizes of projects in organizations with different sizes.
- Furthermore, statistical analyses were performed to examine the significance of the proposed AZ-Model.

The proposed AZ-Model of software development is broadly divided into three phases. The first phase is design and communication (customer involvement phase), in which requirements are collected and the design is developed after the analysis. With respect to collecting the quality requirements, a prototyping tool is introduced to deal with innocent customers, nonfunctional requirements, and usability requirements. With respect to developing the design, unit testing and risk is analyzed along with usability testing [42]. During this phase, a customer is involved till the design is finalized. Customer satisfaction is extremely important for passing the design phase to proceed to the next phase.

The second phase corresponds to the development phase which begins with coding. The provided design is transformed into a programming notation by a programmer. Unit testing and risk analysis are simultaneously performed, and testing is conducted to ensure that the developed code provides results based on the software requirement specification (SRS) along with usability. After obtaining satisfactory results from the testing team, the product is deployed, and a deployment test is conducted with respect to the customer's environment.

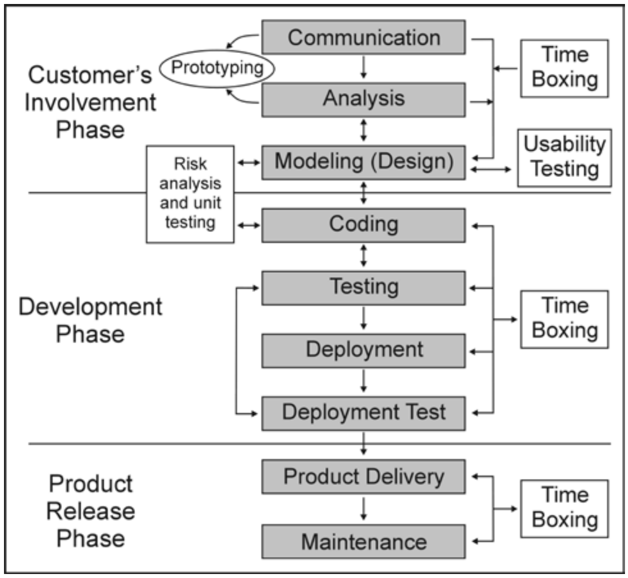

FIGURE 2. AZ-Model of software development.

If the result from the deployment test is satisfactory, then the product enters into third phase of AZ-Model in which it is released in the market based on the nature and ownership of the product. The project manager releases the phase deal based on the nature of the stakeholder. Time-boxing and strong project management are always involved very actively in conjunction with all processing phases of AZ-Model. The hierarchical chart of AZ-Model is listed in Figure 2.

III. CAPABILITY STUDY OF AZ-MODEL

The proposed model includes the following capabilities representing the powers of AZ-Model that prioritize it from the other models.

A. LIMITING WORK IN PROCESS (WIP)

The proposed model includes highly calculated tasks. The developing team works within the prescribed errands defined by the project management team and follows all the instructions or standards. With respect to requirement gathering, the engineers simply collect the specified requirements. The design is developed under the required boundaries, and thus, all the phases include extremely calculated tasks.

B. PROJECT MANAGEMENT (PM)

Project management plays a vital role in the success of a project. Model development and project management are equally important for the successful completion of a project. They blend together to form a complete methodology to deliver a high-quality product to the customer. Throughout system development, the PM and software development

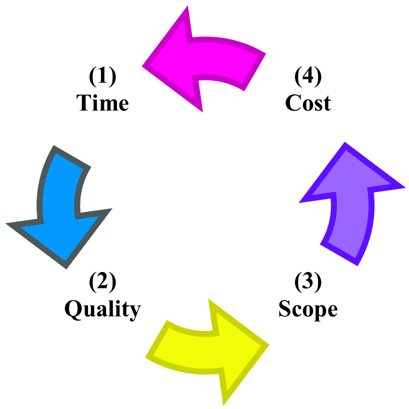

FIGURE 3. Project diamond.

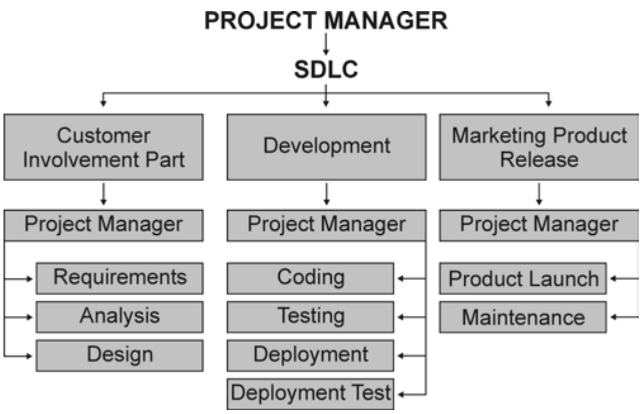

FIGURE 4. Workflow diagram of AZ-Model.

model (SDM) collectively function to achieve milestones. The overall flow of the project is defined beginning with planning, and this is followed by dividing the entire project in to small and manageable segments. Good project management requires specialization of a project manager in relevant areas. This is because a skilled project manager can efficiently manage any required changes during monitoring, given the complexity of the phase for achieving milestone [43]. The four major variables in a project include time, cost, quality, and scope [44]. The project diamond is shown in Figure 3, in which only scope is free in time-boxing.

Hence, AZ-Model is divided into three broad parts that are assigned to the project manager based on the specialization or related work experience. The first phase of AZ-Model is the customer involvement phase, and thus, it is important for a hired project manager to possess the expertise to determine the knowledge of the customer and deal and communicate with a customer, especially when stakeholders belong to different geographical areas [45]. The second phase is the development phase, and the hired project manager for this phase must manage the technicality of work and team members. The last phase is the product-releasing phase, and the choice of the project manager in this phase depends on the nature of the product (developed for customized customer or for general purpose customers). If the product is a general-purpose product, such as Microsoft office, then the hired project manager must be specialized in determining when and where to release product and should market the product from a good business prospective. The effort of estimation significantly impacts project management. Effort depends on the project size. Therefore, effort is calculated as $E = n(size)^m$, where E represents the required effort, n and m represent the factors that are empirically calculated, and size represents the project size that is calculated by using suitable matrix such as function point, line of code, object point, use case point, and number of screens [46], [47]. Figure 4 shows the work flow of a project.

C. MAKING DEVELOPING PROCESS VISIBLE

The proposed methodology is highly visible to present the tasks that should be followed by the developing team members. The methodology also possesses the capacity to control the overloading and enables in adjusting the workflow gap. Visibility is also helpful in determining the time and cost of the project.

D. EFFECTIVE CUSTOMER INVOLVEMENT

Customer involvement significantly impacts SDLC. While comparing both methodologies, a customer is involved in heavyweight methodology only until requirement gathering, while customer is involved throughout the SDLC in lightweight methodology. Often both the fore-mentioned methodologies exhibit negative impacts. Therefore, according to the proposed model, a customer is involved until completion of a satisfactory design.

E. INCREASE THROUGHPUT

Owing to specialized project management in the proposed model, each phase and even each task is well managed. Hence, maximum utilization of available resources and strong time-boxing aids in increasing the throughput, and this is extremely effective for an organization from the business prospective.

F. TIME-BOXING

Time-boxing significantly impacts proper workflow of an organization similar to the manner in which the quality of the product significantly impacts customer satisfaction as well as the goodwill of the organization. Consequently, a process utilized in the software project enables the execution of engineering tasks to achieve goals and in-time delivery to the destination. Additional manpower can be hired for in-time completion because the time-box should be standardized and cannot be expanded [48]. As in other methodologies, time-boxing is used for the timely delivery of products. Therefore, in the proposed model, strong time-boxing is introduced to aid in completing and delivering the project according to the approved time frame. Therefore, the total length of time is divided based on the complexity or nature of the phases of development model. The time provided to each phase is known as a time-box that corresponds to the fixed duration to complete all portions of the assigned phase. We consider a time-box with a time duration T and containing n-phases P1, P2, P3. . . . . ..Pn. As discussed above, each phase Pi is executed by a committed team, and V represents the time required for risky or unforeseen circumstances. Each V denotes the time saved at each stage of the project. It is assumed that the size of the dedicated team for the phase Pi is Si (representing the number of assigned resources to this phase). Each phase has Tp = (T/n–V) available time to complete the task for a time-box, and thus, T/n is the time-box for each stage. When the team in a phase Pi completes the task for the phase of time-box k; it then passes the output of the time-box to the executing team in phase Pi+1, and subsequently commences the execution of its phase for the following time-box k+1. The team size is calculated as if the team size for executing a phase Pi corresponds to Si, and thus, the effort for the phase Pi is E(Pi)=Si(T/n). It should be noted that the duration of T/n for each phase is approximately same. Therefore, the total effort consumed for a phase within a time-box is also calculated as $E(TB) = \sum_{i=1}^{n} E(Si)$, and the project team size is calculated as $\sum_{i=1}^{n} Ri$. Hence, the time-boxing approach enables the utilization of additional manpower to manage the delivery time.

G. EMBEDDING QUALITY

Quality describes the customer satisfaction as well as development organization. While determining the customer satisfaction, a triangle that consists of the time, budget, and customer expectation requirements is considered. Thus, from an organizational viewpoint, another triangle of workflow, goodwill, and business is considered. Hence, the proposed methodology can produce quality products for both customers and organizations.

IV. RESEARCH METHODOLOGY

This section describes the collection and analysis of the data. During the selection of the SDLC method, only experience-based selection criteria should be followed as opposed to any other numerical formula.

In order to determine the validity and efficiency of AZ-Model, individual opinions are collected. Furthermore, the data is collected from organizations after applying the proposed model. The popular six-pointed star model of project management is used to describe the effect of proposed model to the project, as shown in Figure 5.

Traditionally, the success effecting factors of projects include time, cost, and scope [49]. The project management body of knowledge (PMBOK 4.0) introduced an advanced model of triple constraint that is based on six factors, namely schedule, scope, budget, risk, resources, and quality, which are extremely important for the success of a project [50], [51].

The six-pointed star model's factors are divided into two triangles. The scope, schedule, and budget factors on a triangle (Fig. 6) are termed as input or output project factors. However, the risk, resources, and quality factors on the other triangle are termed as process factors (Fig. 7). The schedule

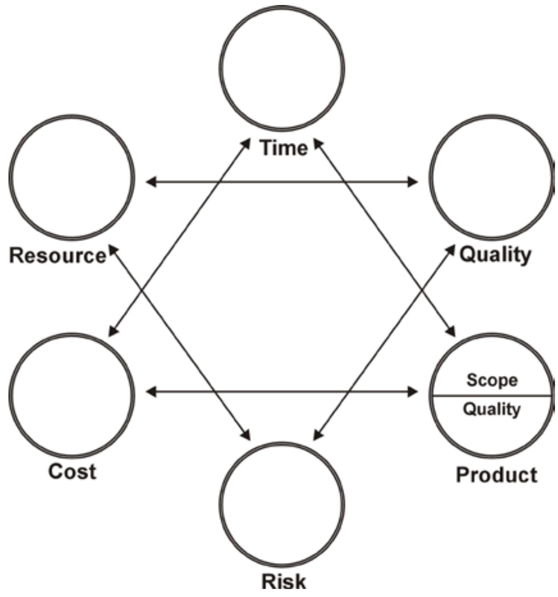

FIGURE 5. Six-pointed star model.

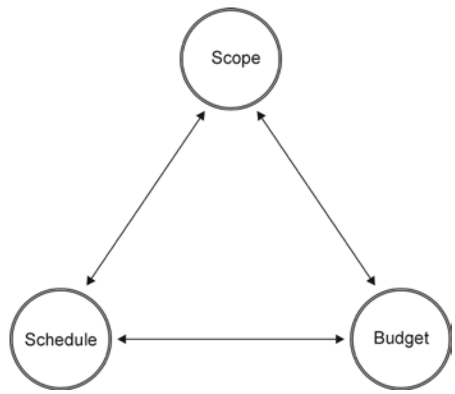

FIGURE 6. First triangle of the six-pointed star.

factor handles in-time completion of a project. Project milestone and SRS implementation is controlled by the scope factor, and the budget factor is used to satisfy the budget, requirements, and return on investment. Risk is analyzed and managed by the risk factor, the availability of resources is managed by the resource factor, and finally, the overall project success and satisfaction is maintained by the quality factor. Hence, it is considered that the aim of all the factors involves checking the validity and efficiency of the proposed model [52].

A. DATA COLLECTION

A survey was conducted to collect responses from employees in organizations that are involved in the implementation of AZ-Model based on their working areas. The survey was conducted from March to December 2016. The questionnaire was designed in two parts, in which the first part contained general information about the respondent and the second part included the factors of the six-pointed star model shown

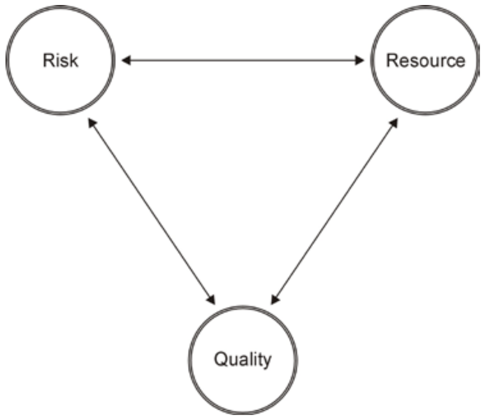

FIGURE 7. Second triangle of the six-pointed star.

TABLE 1. Factor-wise survey questions in the proposed model.

Factor	Survey question	Q. No.
Schedul-ing	By using this model project team are aware of the project status.	2.1
	Project teams get the satisfactory requirements form the customer.	2.2
	Project is delivered on time according to schedule.	2.3
Scope	Project usually has well defined scope	3.1
	Project management methodology is effective to make the scope clear	3.2
Budget	Project completed within budget	4.1
	The project provides good Return on Investment.	4.2
Risk	Project risk and opportunities are managed	5.1
	Business objectives are meet	5.2
Resource s	Human and material resources are mostly available	6.1
	Team can work well together to achieve expected results	6.2
	Maximum utilization of available resources	6.3
Quality	Quality requirements are met	7.1
	Client satisfaction is met	7.2
	The project is successful overall	7.3

TABLE 2. Numerical Response corresponding to the Likert scale.

Response of Likert scale	Response Score
Strongly disagree	1
Disagree	2
Neutral	3
Agree	4
Strongly agree	5

TABLE 3. Factor-wise survey questions in the proposed model.

NO. of projects	Frequency	Percent
1-2 Project	6	27.3
3-8 Project	10	45.5
9-20 Projects	5	22.7
More than 20 Projects	1	4.5
Total	22	100.0

TABLE 4. The model that is already used by the respondent.

Software development models	Frequency	Percent
Waterfall	3	13.6
V-shaped	9	40.9
Spiral	4	18.2
RUP	6	27.3
Total	22	100.0

TABLE 5. Confidence of the developing team.

Response	Frequency	Percent
Disagree	2	9.1
Neutral	4	18.2
Agree	8	36.4
strongly agree	8	36.4
Total	22	100.0

TABLE 6. Size of respondent's organization.

Number of employees	Frequency	Percent
10 to 49	8	36.4%
50 to 250	7	31.8%
Above 250	7	31.8%
Total	22	100.0%

in Table 1. In order to elicit respondent opinions, the Likert scale was used, as shown in Table 2. Each response is associated with a numerical score that generates the numerical response of the collected data. Thus, the proposed model is implemented in 24 software development organizations, and responses are obtained from 22 of the fore-mentioned organizations (91.67%) that nearly completed the projects using AZ-Model.

Table 3 describes the size of a respondent organization. Table 4 represents the model that is used by the respondent organization before or along with the proposed model. While developing the project, (i) the confidence of team members and the confidence for moving the project into the next phase significantly impacts the project quality. Therefore, in order to produce a quality product, team confidence between the team members and between the different phases is extremely important. Table 5 shows the confidence of team members while developing the project at each step while developing a specific phase or moving to the next phase by using the proposed model.

Survey respondents were related to each field of development or included specialist members such as project managers, requirements engineers, analysts, designers, coders, testers, and marketer. Experience of the respondent is also important with respect to project expertise. Thus, based on the general survey information, 41.5% respondents possess less than 2 years of work experience, and 33.0% respondents possess 3 years to 5 years of work experience, and 25.5% of the respondents possess more than 5 years of work

TABLE 7. Summery of the survey response count

Factor	Survey question	S.D	D	N	A	S.A	R.C
Scheduling	By using this model project team are aware of the project status. (2.1)	0.00%	0.00%	23.00%	36.00%	41.00 %	22
		0	0	5	8	9	
	Project teams get the satisfactory requirements form the customer. (2.2)	0.00%	9.00%	32.00%	36.00%	23.00%	22
		0	2	7	8	5	
	Project is delivered on time according to schedule.(2.3)	0.00%	9.00%	23.00%	36.00%	32.00%	22
		0	2	5	8	7	
Scope	Project usually has well defined scope. (3.1)	0.00%	9.00%	27.00%	32.00%	32.00%	22
		0	2	6	7	7	
	Project management methodology is effective to make the scope clear. (3.2)	5.00%	18.00%	18.00%	32.00%	27.00%	22
		1	4	4	7	6	
Budget	Project completed within budget. (4.1)	9.00%	5.00%	32.00%	23.00%	32.00%	22
		2	1	7	5	7	
	The project provides good Return on Investment. (4.2)	0.00%	14.00%	27.00%	23.00%	36.00%	22
		0	3	6	5	8	
Risk	Project risk and opportunities are managed. (5.1)	9.00%	23.00%	23.00%	32.00%	14.00%	22
		2	5	5	7	3	
	Business objectives are meet. (5.2)	0.00%	23.00%	32.00%	36.00%	9.00%	22
		0	5	7	8	2	
Resources	Human and material resources are mostly available. (6.1)	0.00%	5.00%	27.00%	36.00%	32.00%	22
		0	1	6	8	7	
	Team can work well together to achieve expected results. (6.2)	0.00%	9.00%	27.00%	36.00%	27.00%	22
		0	2	6	8	6	
	Maximum utilization of available resources. (6.3)	0.00%	14.00%	23.00%	36.00%	27.00%	22
		0	3	5	8	6	
Quality	Quality requirements are met. (7.1)	0.00%	9.00%	27.00%	41.00%	23.00%	22
		0	2	6	9	5	
	Client satisfaction is met. (7.2)	5.00%	9.00%	32.00%	32.00%	23.00%	22
		1	2	7	7	5	
	The project is successful overall. (7.3)	0.00%	14.00%	18.00%	32.00%	36.00%	22
		0	3	4	7	8	

S.D= strong disagree, D= Disagree, N= Neutral, A=Agree, S. A=Strong agree, R.C=Response Count

TABLE 8. Average score and percentage of the six-pointed star.

	Likert scale									
Factors	1		2		3		4		5	
	AVG	%age	AVG	%age	AVG	%age	AVG	%age	AVG	%age
Schedule	0.00	0	1.33	6.06	5.67	25.76	8.00	36.36	7.00	31.82
Scope	0.50	2.27	3.00	13.64	5.00	22.73	7.00	31.82	6.50	29.55
Budget	1.00	4.55	2.00	9.09	6.50	29.55	5.00	22.73	7.50	34.09
Risk	1.00	4.55	5.00	22.73	6.00	27.27	7.50	34.09	2.50	11.36
Resource	0.00	0	2.00	9.09	5.67	25.76	8.00	36.36	6.33	28.79
Quality	0.33	1.52	2.33	10.61	5.67	25.76	7.67	34.85	6.00	27.27

TABLE 9. Correlation between success of the project and project management factors.

Factors	Persons correlation with quality factors
Schedule	0.82
Scope	0.63
Budget	0.76
Risk	0.79
Resource	0.67

experience. Hence, Table 6 shows the size of the respondent organizations. The organizations with 10–49 employees have a response rate of 36.4%, organizations with 50–250 employees have a response rate of 31.8%, and the organizations with more than 250 employees have a response rate of 31.8%.

TABLE 10. Average and standard deviation values for each factor.

Factor	Mean	Std. Dev.
Schedule	5.5	3.162
Scope	4.4	2.71
Budget	4.4	2.76
Risk	4.4	2.59
Resource	5.5	2.62
Quality	4.4	2.89

B. RESULT AND DISCUSSION

This section discusses statistical analyses based on numerical results collected from the survey respondents that determine how the management factors are related to each other and how AZ-Model affects each factor of the six-pointed star model. The collected results from the respondent are discussed in a statistical form.

The survey response of the six-pointed star model is summarized in Table 7, which presents all the achieved frequencies from the respondents. The frequencies and percentage of computed frequencies are declared in the form of a Likert scale. Each part of the Likert scale significantly impacts the analysis of the significance of the proposed model. Hence,

TABLE 11. Brief comparison between traditional and proposed model

Traditional Model	Drawback	Solution By Propose Model
Waterfall	• Customer involvement only in first phase. • Requirements documents are freeze in first phase. • High amounts of risk and uncertainty. • Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model. • It is difficult to measure progress within stages. • Prototyping is not used to getting the requirements.	• Customers are involved until the design is finalized. • Requirements documents are not freeze until the design is finalized. • Risk and uncertainty is very low due to analyze the risk in customer involvement phase. • Suitable for the project where requirements are at moderate to high risk of changing because requirements can be changeable until the design is finalized. • It's able to messieurs the progress with in the stages due to good project time boxing chart at each stage. • Prototyping is used to get the true and very clear requirements and maximum customer involvement especially for innocent customer.
Iterative	• System architecture or design issues may arise because all requirements are not gathered in the beginning of the entire life cycle. • Much customer involvement disturb the time boxing and bought. • Not very good method for proper working flow of an organization. • Management is more. Complex. • Completion of the project is not known early. • Project's progress is highly dependent upon the risk analysis phase.	• All the requirements are collected before starting the development phase. • Customer should be involve only in customer involvement phase through a strong time-boxing. • Through strong time boxing all the projects are finished in time so it's an effective model for proper workflow of an organization. • Project is managed through a proper working flow diagram. • Project completion is known from the starting of the project and time boxing is managed according to the available time. • Risk is analyzed before starting to development phase and in between development though unit testing.
RAD	• Dependency on technically strong team members for identifying business requirements. • It deal the project that can be modularized. • Requires highly skilled developers/designers. • High dependency on modeling skills. • Inapplicable to cheaper projects. • Management complexity is more. • Requires user involvement throughout the life cycle. • Rapid application development produce low quality product as it not consume so much time on requirement gathering, Analysis and designing rather than it's focuses on Development phases.	• All the requirements are gathered through the proper work flow as usual. • It will deal every types of project. • All the work done through proper channel so high skilled developer or designers are not prerequisites. • Not so much dependency on design. • It can be applicable for cheaper and smaller types of project. • Management is not complex, proper project management is introduced in every phase. • User just involve in first phase which is customer involvement phase. • The proposed model greatly focuses on the Requirement, analysis and design.
RUP	• Rational Unified process for development only focuses on functional requirement.	• This propose model deal with functional and non-functional requirement as well.
Spiral	• Management is more complex. • End of project may not be known early. • Not suitable for small or low risk projects. • Large number of intermediate stages requires excessive documentation. • Spiral methodology has not time limit to get a mile stone and require very experience project manager and team.	• Have good project management. • Through strong time boxing deadline in known early. • It's suitable for small and low risk projects. • Have limited intermediate stages no excessive documents are required. • This methodology depends on dead line according to every process and not much experienced team and project manager required.
Agile	• Not suitable for handling complex dependencies. • More risk of sustainability, maintainability and extensibility. • Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction. • There is very high individual dependency, since there is minimum documentation generated. • Transfer of technology to new team members may be quite challenging due to lack of documentation.	• Suitable for handling complex dependencies. • Have proper mechanism for sustainability, maintainability and extensibility. • Not much depends on customer have technique to understand the customer. • Have good and motivated team work as well project manager and effective documentation is done. • Transfer technology is not a big challenge due to sufficient documentation.
V-Model	• High risk and uncertainty. • Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects. • Not suitable for the projects where requirements are at a moderate to high risk of changing.	• Every task is clear and risk should be analyzed at early stage. • Good for complex and object oriented projects. • It's suitable where requirements because customer should be involved until the design is finalized.

the table shows that the values of strongly disagree and disagree are lower than the values of agree and strongly agree that represent the positive response of the respondents toward

FIGURE 8. **Frequency analyses of six pointed star model.**

the proposed model. Furthermore frequency analyses of agree +strong agree is presented in figure 8.

In Table 8, the average and percentage of each Likert scale for all the six-pointed star factors are provided to analyze the significance of the proposed model based on the respondents. The average score and percentage is calculated as related to 2–3 questions for each factor. The Likert scale is shown in Table 2. For example, the Schedule factor corresponds to question numbers 2.1, 2.2, and 2.3, and thus, the accumulative average and percentage of all the questions is computed. The schedule factor represents that 25.76% respondents are neutral while 36.36% agreed and 31.82% respondents strongly agreed with respect to the scheduling of the proposed model. Thus, the average and percentage of all the factors indicate positive responses from the respondents with respect to the proposed model.

C. CORRELATION BETWEEN THE SUCCESS OF THE PROJECT AND PROJECT MANAGEMENT FACTORS

The quality factor is directly related to the inclusive success of the project. The quality factor is included in the survey questionnaire to examine a stakeholder's satisfaction, check the requirements for quality projects, and determine project success. A sanity check is performed to ensure the positive correlation of the quality factors with respect to the other five factors of project management. Hence, the positive correlation among all other factors with the quality factor indicates that the proposed model with each individual factor affects the overall quality of the project.

The Pearson's correlation values are computed for average scores of all participants for the success factor with respect to the average score of the other factors of the six-pointed star model. If the computed value is 1, then it indicates perfect correlation and 0 indicates no correlation. Table 9 shows the correlation values. It should be noted that values <0.05 represent the significance correlation at the 95.00% confidence level.

Therefore, the results of the correlation indicate that the quality factor is positively correlated with other factors of the six-pointed star model that represents project success. The range of correlation values from 0.63 to 0.82 reveals that the factors are highly correlated. Hence, the positivity of the correlation describes the effect of the proposed model on the five factors of the six-pointed star model for the quality factor.

Table 10 analyzes the effect of individual factors of the proposed model. For this purpose, the mean and standard deviation of each factor of the six-pointed star model is computed. There are 22 respondents and 2–3 questions for each factor. Only 15 questions are related to the six-pointed star model. Therefore, the collected data is used to compute the average and standard deviation values. Mean and standard deviation shows the factor-wise levels of satisfaction. Therefore, the results of the schedule factor show the extent of satisfaction with a factor relative to the other factors.

D. BRIEF COMPARISON OF THE TRADITIONAL AND PROPOSED MODEL

See Table 11.

V. CONCLUSION

Software development methodology is extremely important to enable each software development organization to develop a quality project within a given time period and budget. A review of extant literature reveals that it is necessary to determine the limitations and gap between different software development methodologies. The state of the art indicates that the main contradiction between different methodologies should include adoptability and predictability, should be people-oriented and process-oriented, requirements collection and requirement change management. In order to

TABLE 12. Appendix (Questionnaire survey)

Section- A1 (Respondent Information)		
Full Name (optional)		Job Title Position
Have you ever been participated in an software development methodology	Yes ☐	NO ☐
Working Experience (Years) in Software development organization		
What is the scope of your company?	Client ☐	Vendor ☐
	Not sure ☐	Other
Email Address		
Current address of your organization including country		
How many years of industry/academia experience do you have in your field?		
Have you ever participated in Software development life cycle Improvement Project?	yes ☐	NO ☐
Section- A2 (Organization Detail)		
Name of Organization (Optional)		
What is the primary business function of your organization? (You may tick more than one)	Collocated Software development ☐	Global/offshore Software development ☐
	Research ☐	Other
Please specify the size of your organization.	Small ☐	Medium ☐
	Large ☐	Not sure ☐
Please specify the number of employees in your	Less than 50 ☐	51-100 ☐
	101-150 ☐	Greater than 150 ☐
Please specify the type of your organization	National ☐	Multinational ☐
	Not sure ☐	Other
Does your organization adopted Software development life cycle Process Improvement standards or (CMMI/ISO)	CMMILevel-1 (Initial) ☐	CMMILevel-2 (Managed) ☐
	CMMILevel-3 (Defined) ☐	CMMILevel-4(Quantitatively Managed) ☐
	CMMILevel-5 (Optimizing) ☐	ISO ☐
	Other	Not sure ☐
Which software development methodology your organization adopted		
Which factor is most important for you as a professional when adopting a method	Low cost ☐	Easy to handle ☐
	Great productivity ☐	Great reliability ☐
	Other	Not sure ☐

*

fill the gap and eliminate the limitations of methodologies, a comprehensive software development methodology "AZ-Model" is introduced. The proposed model is broadly divided into three phases, namely the customer involvement phase, development phase, and releasing phase. According to the proposed model, the customer is involved in SDLC until the completion of satisfactory design of the project. The effective involvement of the customer minimizes the occurrence of risks due to the changing requirements. Furthermore, prior to commencing the development phase, unit testing is

TABLE 12. ***Continued.***

Section: B –Questions related to six pointed star model. The aim of this section is to collect factor wise responses from the participants which make it possible to analyze the proposed model according to six pointed star model provided by the project management Body of knowledge (PMBOK 4.0). The factor wise collection of data make us enable to check that the proposed model is efficient for what factor to what extent. Please rank each question according to your own understanding and experience about AZ-Model of software development.						
SDA= strong disagree, DA= Disagree, N= Neutral, A=Agree, S A=Strong agree						
Factors	Questions	SDA	DA	N	A	SA
Scheduling	By using this model project team are aware of the project status					
	Project teams get the satisfactory requirements form the customer.					
	Project is delivered on time according to schedule.					
Scope	Project usually has well defined scope					
	Project management methodology is effective to make the scope clear					
Budget	Project completed within budget					
	The project provides good Return on Investment.					
Risk	Project risk and opportunities are managed					
	Business objectives are meet					
Resources	Human and material resources are mostly available					
	Team can work well together to achieve expected results					
	Maximum utilization of available resources					
Quality	Quality requirements are met					
	Client satisfaction is met					
	The project is successful overall					
Section: C-Add your comment if any						

conducted in the design phase to analyze and minimize the risk in conjunction with usability testing. Strong time-boxing and specialized project management ensures in-time delivery for proper workflow of the project as well as organization. Specialized project management ensures that each phase and even each task of the project is well managed and available resources are maximally utilized. Hence, effective time-boxing and good project management improves throughput of the organization.

In order to prove the effectiveness of the proposed model, a survey was conducted to collect the opinions of the proposed model from the users, and statistical tools are applied to analyze the collected data. The designed questionnaire contains different queries regarding general information on the respondent organization, and six-pointed star models were used to collect the technical feedback about the proposed model. Hence, the proposed model is implemented in 24 well-known organizations although responses were obtained from 22 organizations that almost completed the project by using AZ-Model. Statistical tools were applied to calculate the mean, standard deviation, and correlation at 95% confidence level between the factors of the six-pointed star model to present the effectiveness of AZ-Model. Statistical results listed in Table 8 show the cumulative mean and percentage for each factor. Hence, the percentage of agree and strongly agree significantly exceed other elements of the Likert scale, and this indicates the positive impact of AZ-Model on the six-pointed star model. Table 9 presents the correlation of quality factor with respect to other factors of the six-pointed star model, and Table 10 shows the most positive effective factor of the proposed model. The statistical results reveal that AZ-Model is extremely effective for software organizations to produce a quality product within a given time and budget.

A future study will involve examining the effect of different software development methodologies on the six-pointed star model and a brief comparison of AZ-Model with other software development methodologies in statistical form.

APPENDIX
QUESTIONNAIRE SURVEY

See Table 12.

REFERENCES

[1] N. M. A. Munassar and A. Govardhan, "A comparison between five models of software engineering," *Int. J. Comput. Sci. Issues*, vol. 7, no. 5, pp. 95–101, 2010.

[2] S. U. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: A systematic literature review," in *Proc. 4th IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Limerick, Ireland, Jul. 2009, pp. 207–216.

[3] J. E. Purcell, "Comparison of software development lifecycle methodologies," in *Proc. Web Appl. Secur. Workshop*, Feb. 2007, pp. 1–9.

[4] I. Sommerville, *Software Engineering*, 7th ed. Reading, MA, USA: Addison-Wesley, 2004.

[5] H. D. Benington, "Production of large computer programs," in *Proc. Navy Symp. Adv. Programm. Methods Digit. Comput. Office Naval Res.*, Washington, DC, USA, 1956, pp. 15–28.

[6] W. W. Royce, "Managing the development of large software systems," in *Proc. IEEE Wescon*, Aug. 1970, pp. 328–338.

[7] B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.

[8] B. W. Boehm, *Software Engineering Economics*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1981.

[9] V. R. Basili and A. J. Turner, "Iterative enhancement: A practical technique for software development," *IEEE Trans. Softw. Eng.*, vol. SE-1, no. 4, pp. 390–396, Dec. 1975.

[10] C. Larman and V. R. Basili, "Iterative and incremental developments. A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003, doi: 10.1109/MC.2003.1204375.

[11] P. Kruchten, *The Rational Unified Process—An Introduction*. Reading, MA, USA: Addison-Wesley, 2000.

[12] B. W. Boehm, "A spiral model of software development and enhancement," *IEEE Comput.*, vol. 21, no. 5, pp. 61–72, May 1998.

[13] R. Llewellyn. (2005). *Software Development Life Cycle Models*. [Online]. Available: http://codebeter.com

[14] S. Mathur and S. Malik, "Advancements in the V-model," *Int. J. Comput. Appl.*, vol. 1, no. 12, pg. 29-34, 2010, doi: 10.5120/266-425.

[15] M. James. (2009). *Scrum Methodology*. [Online]. Available: http://scrummethodology.com

[16] S. W. Ambler, "The agile scaling model (ASM): Adapting agile methods for complex environments," *Environments*, Jan. 2009.

[17] K. Beck, *Extreme Programming Explained: Embrace Change*. Reading, MA, USA: Addison-Wesley, 2000.

[18] P. Abrahamsson, O. Salo, J. Warsta, and J. Ronkainen. (Sep. 2017). *Agile Software Development Methods: Review and Analysis*. [Online]. Available: http://www.inf.vtt.fi/pdf/

[19] P. Abrahamsson, "Extreme programming: First results from a controlled case study," in *Proc. Eur. Conf.*, Sep. 2003, pp. 259–266.

[20] K. Schwaber and M. Beedle, *Agile Software Development With SCRUM*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, Oct. 2001.

[21] L. Rising and N. S. Janoff, "The Scrum software development process for small teams," *IEEE Softw.*, vol. 17, no. 4, pp. 26–32, Aug. 2000.

[22] S. U. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: An empirical study," in *Proc. 11th Int. Conf. Product Focused Softw. Develop. Process Improvement (PROFES)*, Limerick, Ireland, 2010, pp. 146–159.

[23] A. Tariq and A. A. Khan, "Framework supporting team and project activities in global software development (GSD)," in *Proc. Int. Conf. Emerg. Technol. (ICET)*, 2012, pp. 1–6.

[24] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2617–2634, 2005.

[25] S. Pokharel, "Stakeholders roles Virtual project environment: A case study," *J. Eng. Technol. Manag.*, vol. 28, no. 3, pp. 201–214, 2011.

[26] C.-Y. Chen, P.-C. Chen, and Y.-E. Lu, "The coordination processes and dynamics within the inter-organizational context of contract-based outsourced engineering projects," *J. Eng. Technol. Manag.*, vol. 30, no. 2, pp. 113–135, 2013.

[27] Q. Yang, S. Kherbachi, Y. S. Hong, and C. Shan, "Identifying and managing coordination complexity in global product development project," *Int. J. Project Manag.*, vol. 33, no. 7, pp. 1464–1475, 2015.

[28] S. Schneidera, R. Torkarb, and T. Gorschek, "Solutions in global software engineering: A systematic literature review," *Int. J. Inf. Manag.*, vol. 33, no. 1, pp. 119–132, 2013.

[29] L. Layman, L. Williams, D. Damian, and H. Bures, "Essential communication practices for extreme programming in a global software development team," *Inf. Softw. Technol.*, vol. 48, no. 9, pp. 781–794, 2006.

[30] T. Nguyen, T. Wolf, and D. Damian, "Global software development and delay: Does distance still matter?" in *Proc. IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Aug. 2008, pp. 45–54.

[31] D. Avison and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*, 3rd ed. New York, NY, USA: McGraw-Hill, 2003.

[32] M. Huisman and J. Iivari, "Systems development methodology use in South Africa," in *Proc. 9th Amer. Confon Inf. Syst.*, 2003, pp. 1040–1052.

[33] B. Fitzgerald, "An empirical investigation into the adoption of systems development methodologies," *Inf. Manage.*, vol. 34, no. 6, p. 317328, 1998.

[34] A. Cockburn, *Agile Software Development*. Reading, MA, USA: Addison-Wesley, 2002.

[35] S. B. Henderson, "Method engineering for 00 systems development," *Commun. ACM*, vol. 46, no. 10, pp. 73–78, 2003.

[36] J. Ralyté, R. Deneckère, and C. Rolland, "Towards a generic model for situational method engineering," in *Proc. 15th ICAISE*, Jun. 2003, p. 1029.

[37] M. Huisman and J. Iivari, "The individual deployment of systems development methodologies," in *Proc. 14th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, Toronto, ON, Canada, May 2002, pp. 134–150.

[38] J. Highsmith, *Adaptive SW Development*. London, U.K.: Dorset House, 2000.

[39] M. J. Gallivan, "Organizational adoption and assimilation of complex technological innovations: Development and application of a new framework," *ACM SIGMIS Database*, vol. 32, no. 3, pp. 51–85, 2001.

[40] G. C. Green, R. W. Collins, and A. R. Hevner, "Perceived control and the diffusion of software process innovations," *J. High Technol. Manage. Res.*, vol. 15, no. 1, pp. 123–144, 2004.

[41] A. Mateen, M. Azeem, and M. Shafiq. (2016). "AZ model for software development." [Online]. Available: https://arxiv.org/abs/1612.08811

[42] S. Velmourougan *et al.*, "Software development life cycle model to build software applications with usability," in *Proc. Int. Conf. IEEE Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2014, pp. 271–276.

[43] P. Jalote, *CMM in Practice–Processes for Executing Software Projects at Infosys* (SEI Series on Software Engineering). Reading, MA, USA: Addison-Wesley, 2000.

[44] K. Beck, *Extreme Programming Explained*. Reading, MA, USA: Addison-Wesley, 2000.

[45] M. Niazi *et al.*, "Challenges of project management in global software development: A client-vendor analysis," *Inf. Softw. Technol.*, vol. 80, pp. 1–19, Dec. 2016.

[46] D. D. Galorath and M. W. Evans, *Software Sizing, Estimation, and Risk Management*. Boca Raton, FL, USA: Auerbach Publications, 2006.

[47] L. M. Laird and M. C. Brennan, *Software Measurement and Estimation: A Practical Approach*. Hoboken, NJ, USA: Wiley, 2006.

[48] F. P. Brooks, *The Mythical Man Month*. Reading, MA, USA: Addison-Wesley, 1975.

[49] C. Chatfield and T. Johnson. (2007). *Microsoft Office Project 2007 Step by Step*. [Online]. Available: http:office.microsoft.com/en-us/project-help/a-short-course-in-project-management-HA010235482.aspx#BMtime

[50] W. R. Duncan, *Project Management Processes, A Guide to Project Management Body of Knowledge*. Charles, MD, USA: Automated Graphic Systems, 1996, pp. 27–39.

[51] K. Schwaber, *Agile Project Management With Scrum*. Redmond, WA, USA: Microsoft Press, 2004.

[52] H. Lei *et al.*, "A statistical analysis of the effects of scrum and Kanban on software development projects," *Robot. Comput.-Integr. Manuf.*, vol. 43, pp. 59–67, Feb. 2017.

MUHAMMAD AZEEM AKBAR received the M.Sc. in computer science and M.S. in computer science degrees from the University of Agriculture Faisalabad, Faisalabad, Pakistan, and he is currently pursuing the Ph.D. degree with the Department of Software Engineering, Chongqing University, China. His research interests are global software development life cycle, requirements engineering, empirical studies, global software requirements change management, software defect prediction, and software risk management. He has Outstanding Academic carrier.

JUN SANG received the B.Sc. degree from Shanghai Jiaotong University, China, in 1990, and the M.E. degree and the Ph.D. degree in computer science from Chongqing University, China, in 1993 and 2005, respectively. He is currently a Professor with the School of Software Engineering, Chongqing University. His research interests include software engineering, digital image watermarking, information security, and image processing.

ARIF ALI KHAN (M'17) received the B.S. degree in software engineering from the University of Science and Technology, Bannu, Pakistan, in 2010, the M.Sc. degree in information technology from Universiti Teknologi PETRONAS, Malaysia, and the Ph.D. degree in software engineering from the Department of computer science, City University of Hong Kong. He is currently an Active Researcher in the field of empirical software engineering. He is interested in software process improvement, 3C's (communication, coordination, and control), global software development, and evidence-based software engineering. He has participated in and managed several software engineering related research projects. He is a Student Member of ACM.

FAZAL-E-AMIN received the B.S. degree in computer science from Hamdard University in 2003, the master's degree in information technology from Quid-i-Azam University in 2005, the master's degree in software engineering from International Islamic University in 2008, and the Ph.D. degree from the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, in 2012. He is currently an Assistant Professor with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University. His research interest includes open source software, software usability, software quality, and global software development.

NASRULLAH received the M.S. degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2009. He is currently pursuing the Ph.D. degree in software engineering from the Chongqing University of China. His research interests are in image compression, image encryption, data hiding, machine learning, and deep learning.

MUHAMMAD SHAFIQ received the master's degree in computer science from the University of Agriculture, Faisalabad, Pakistan, in 2015. He is currently pursuing the Ph.D. degree in software engineering from the Chongqing University of Posts and Telecommunications, China. His research interests are global software development life cycle, requirements engineering, empirical studies, global software requirements change management, software defect prediction, and software risk management.

SHAHID HUSSAIN received the M.Sc. in computer science and the M.S. in software engineering degrees from Gomal University, Dera Ismail Khan, and the City University of Science and Information Technology, Peshawar, Pakistan, and the Ph.D. in software engineering from the Department of Computer Science, City University of Hong Kong. His research interests include the software design patterns and metrics, text mining, empirical studies, data science, and software defect prediction. He was a recipient of the Outstanding Academic Performance Award from City University of Hong Kong.

HAIBO HU received the master's degree in software engineering and the Ph.D. degree in computer science from Chongqing University, China, in 2004 and 2012, respectively. He is currently an Associate Professor with the School of Software Engineering, Chongqing University, China. He is also a Research Associate with the Key Laboratory for Dependable Service Computing in Cyber Physical Society of MoE, China. His current research interests include software engineering, pattern recognition, semantic web, and visual analytics.

MANZOOR ELAHI received the Ph.D. degree from Graduate School of the Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with COMSATS Institute of Information Technology, Islamabad. His Ph.D. thesis was title Detecting Outliers for Data Stream Under Limited Resources, under the supervision of Prof. H. Wang, Intelligence Engineering Laboratory, Institute of Software, Chinese Academy of Sciences. His major areas of interest are mining data streams, ubiquitous data mining, intrusion detection, distributed data mining, machine learning and data mining, real-time system, publish/subscribe system, wireless sensor networks, and active databases ECA rules.

HONG XIANG received the B.Sc. degree in mathematics from Sichuan Normal University, China, in 1984, the M.Sc. degree in mathematics and computer science from Lakehead University, Canada, in 1992, and the Ph.D. degree in mathematics from the University of Alberta, Canada, in 1998. He is currently a Professor with the School of Software Engineering, Chongqing University. His research interests include software engineering, information security, and cryptography.