IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2017, accepted September 29, 2017, date of publication October 26, 2017,
date of current version November 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2762729

Build Software or Buy: A Study on Developing Large Scale Software

BASIT SHAHZAD[1], ABDULLATIF M. ABDULLATIF[2], NAVEED IKRAM[3], (Senior Member, IEEE), AND ATIF MASHKOOR[4]

[1]Department of Engineering, Faculty of Engineering and Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan
[2]College of Computer Science, King Saud University, Riyadh 11543, Saudi Arabia
[3]Faculty of Computing, Riphah International University, Islamabad 44000, Pakistan
[4]Software Competence Center Hagenberg GmbH, Hagenberg 4232, Austria

Corresponding author: Basit Shahzad (basit.shahzad@gmail.com)

The work of A. Mashkoor was supported in part by the Austrian Ministry for Transport, Innovation and Technology, in part by the Federal Ministry of Science, Research and Economy, and in part by the Province of Upper Austria in the frame of the COMET center SCCH.

ABSTRACT Software development is a set of activities which time, budget, and effort of the human resource. Over the years, the software development process has matured to enable the adaptation and integration of commercially available components. The availability of commercially-off-the-shelf and modifiable-off-the-shelf components has transferred the complexity from development and design phases to the integration phase and we can observe large- application development by integrating the available components. Often, due to the limitation of technological and other resources, developing a software application in-house may be less beneficial. In such circumstances, development firms opt to either buy software or outsource the development. In this paper, we identify the factors that govern the decision of making software applications in-house, outsourcing them, or buying them from the market. Since the concept of component integration is more common in large scale applications, in this paper we consider this case.

INDEX TERMS Software make vs buy decision, software outsourcing, domestic outsourcing.

I. INTRODUCTION

The software development activity is a set of procedures that yields software as an outcome. The procedure and standards in the software development have improved over time and practices like CMM (Capability Maturity Model), now CMMI (Capability Maturity Model Integration), is often considered as an industry standard in software development [1]–[3]. Presence of software in computing devices, mobiles, wearable devices and phablets has made software an entity of extreme relevance in running the life's events and the development of software has therefore increased over time. Ebert [4] has identified that the producers and consumers of software applications have increased, software applications have rationally become more customizable to meet requirements of businesses.

Khan [5] has identified that in a typical software development environment, where procedures, standards and team structures are followed, developing software is an expensive activity. The firms, who do not spend planned time and necessary fiscal resources, invite software risks in the development activity and increase the financial liabilities and losses [5]. Developing software, however, is a time and cost intensive activity that requires many other resources to be placed in accordance with requirements of the development activity [6]–[8].

Many software models can guide about the cost and time estimates. Firms evaluate the cost of buying a piece of software versus developing it on their own. While both paradigms have their pros and cons, the decision of build or buy is not easy to take and has a number of factors to be considered before reaching a conclusion. Gomez *et al.* has argued that the build versus buy decision is really tough to make, especially when the decision is being taken for the first time [9]. In his opinion, the organizations, while making a build versus buy decision, find themselves in one of the three possible scenarios that include: buy a package solution and customize it to fit, buy a package and change the organizational needs to fit the package, or develop a customized in-house software solution that has its own advantages and disadvantages. A. Wilson has defined some pros and cons of the in-house and the vendor-oriented development [10]. In his opinion, the in-house development should be adopted if there is an ad-hoc single business process to address, the problem to address is unique, the problem is isolated, and the staff to develop software is available.

The in-house software development has certain advantages and disadvantages [11]. One of the advantages is the complete control over the application code and development schedule. On the other hand, the cons of the in-house development include: constant staff engagement, lower functionality, and

TABLE 1. Literature selection with respect to relevance.

Search Term	Results Returned	Results Discarded	Relevant	Time
Software make vs buy decision for large scale projects	382,000	-	Partially	Open
Software make vs buy decision for large scale projects	174,000	208,000	Partially	Open
"Software make vs buy" decision for large scale projects	94	207,906	Partially	Open
Software make vs buy decision" for large scale projects	41	53	Partially	2000-2017

updates are hard to make as a complete iteration of development is required for this purpose which causes expensive time, cost, and human resource commitments.

A. Wilson has argued that a software application may be bought from a vendor when the use of software is critical, the problem is common, the application of software is organization-wide, and a sound software development department is not available [10]. Buying software from a vendor, however, has its own advantages and disadvantages. The pros of buying software from a vendor include the ready-made solution, flexible software, support and training, and enhanced functionality through customer's feedback. The cons of buying software from a vendor include: the functionality determination by the vendor, vendor's right on code, and dependency on the vendor for the support of issues and updates.

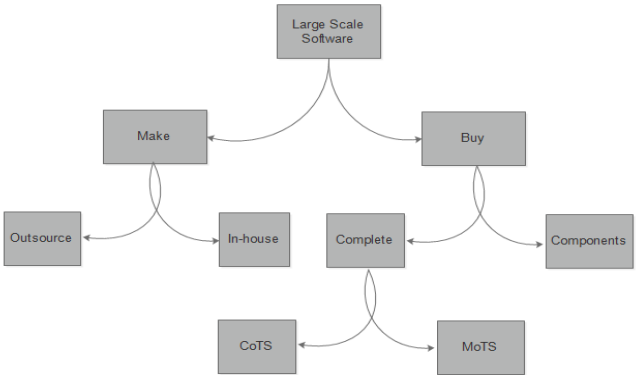

FIGURE 1. Software build versus buy decision.

In this research article, we consider the case for large scale software development and discuss several possible paths in this regard. The build versus buy decision, in developing the large scale software, may take any shape as shown in Fig. 1. Figure 1, describes the possible path of software's build versus buy decision. When an enterprise needs a piece of software to be developed, it can either decide to make it or buy it. The decision to build software requires the environment to be available for the purpose of software development [8], [12], [13]. If the organization opts otherwise, i.e., to buy the software, it is further decided that if the organization is in need of the components or a complete solution. Components and complete solutions both are available in the form of the CoTS (Commercially of The Shelf) product or MoTS (Modifiable of The Shelf) [14], [15].

In this research study, we consider four out of the five terminal points as shown in Figure 1. We conduct this study with the assumption that the organization buys software only when it currently does not have the capability to develop. In this case, 4 cases are considered as shown in Table 1. The fifth case is not considered as we are dealing with the large-scale software projects in this paper, which are developed in totality and not by integrating the open components. The development of software by integrating the CoTS and MoTS is always possible and rather popular development paradigm [16], [17].

It is, however, evident that the build versus buy decision is not a straight decision and requires ample consideration of facts, parameters, the economic situation, and the realization of the organizational strengths and weaknesses [10], [18]. The process of decision-making is discussed throughout this paper. Section 2 of the paper elaborates the available literature on the subject and research questions are formed at the end of the literature review section. Factors for making the make versus buy decision are identified in Section 3 and the methodology is discussed in Section 4. The qualitative and quantitative methods and their application are discussed in Section 5 and 6, respectively. The triangulation process is discussed in Section 7 and a detailed discussion on the results is given in Section 8. The paper is concluded in Section 9.

II. RELATED WORK

In order to reach the decision whether to build software or buy, a considerable amount of literature is available that provides ample information about the factors and the environment for making such decision. It is important to note that the outsourcing of software has emerged as a consolidated activity and the volume is growing. P. G. Klein has identified that in 2003 the US was spending $170 billion on software outsourcing while the spending on the outsourced projects was projected to further grow to $ 575 billion in 2016 [19]. The growth in outsourcing software projects indicates that the outsourcing is a trustworthy business and has a good

return-on-investment. However, on the other hand, it also creates a deficit in the local job market.

Daneshgar *et al.* has performed a detailed mixed method research study and has identified that there are certain factors that contribute in taking this decision [18]. The study started with the initial proposal of 10 factors including strategy and competitive advantage, cost, scale and complexity, requirement fit, time, in-house development expertise, risk, support structure, operational factors, and intellectual property. The factors were largely admitted by the respondents. The factors for considering the make-buy decision are relevant in the small and medium enterprise and the author has not considered the large-scale projects in this case. Usmanij *et al.* is of the view that buying software is convenient as compare to development [20]. The author believes that most firms will reach to the conclusion of buying software instead of developing if the strategic aspect of buying vs developing is considered. The vendor must be vetted before the decision to buy software is made. The world bank in one of its recent report on make or buy decision factors [21] has reported the key parameters in making a decision that whether a software product must be bought or developed in-house. In this report, they identified three key and six preliminary decision factors that are recommended to be taken into consideration before making such decisions. The key criteria include richness of the required functionality, volatility of functional requirements, and organization's capacity for software maintenance. The report suggests that the organization may opt to develop a piece of software instead of buying it if it has the competence in business and technical functionality, expertise in system development, legacy systems, and integrated application suites. While in the packaged solutions, the key concerns like adaptation, customization, integration, and vendor selection may always be present.

D. Margan and S. Čandrlić has compared the open source and the customized software development process with empirical data and has concluded results in the context of software maintenance [22]. Software maintenance emerges as a big challenge in the customized software. The integration of the packaged software is not easy and the help for problem fixation is remote. Further, the frequent problems can be unique with respect to the development environment of the organization. In the in-house developed software, however, the problems are frequent but known and the development team is already and always available to rectify them. The term outsourcing is also referred to as right-sourcing in the context where its benefits are proven. [23] and [24] have elaborated that the outsourcing for the purpose of buying a component is done primarily for some (or all) reasons like cost reduction, superior quality, and the need for flexibility. Chiesa also argued that it is not always the case that the organization proceeds for the complete outsourcing. Even in the case, when an organization considers buying a piece of software instead of developing it, the organization may partially outsource the project by keeping the doors of economic opportunity open to them. J. Rundquist has elaborated that the development of new software may possibly move through three stages, namely, considering outsourcing of new software (or part of the software), selecting the outsourcing partner, and choosing which activities to outsource [23]. Karlsbjerg *et al.* has highlighted the software make or buy decision based on a taxonomy of Intranet approaches [25]. The paper outlined the widespread adoption of intranet by information centers, typically to provide communication between geographically spread organizational units such as members, employees, and others authorization. The knowledge obstacles to adoption have been let down by the emergence of ICT tools and afterwards availability of ready-made intranet-in-a-box packages and advancement of general awareness among users. The study aimed at varying risk over time and focus the prerequisite for psychological indentures to resolve such outsourcing connections [26]. T. Rands help identifying policies for making or buying software [15]. A. Moses and P. Åhlström said that in-house development is observed common when the organization is technologically progressing. opined to determine the dimensions along which make or buy decision processes change in excess of time [27]. In this study, nine extents were found along which the make or buy decision process had improved over time. The dimensions were cross-functionality, construction, symmetry, stiffness, consciousness, compulsory, information distribution, management, and tractability.

C. Fill and E. Visser has demonstrated a composite outsourcing approach to take the build or buy decision [28]. These configurations consist of three major components, i.e., to determine the exclusive appropriate features associated with apiece decision, the policies consequence of deciding to outsource, and the customary cost facets. P. Humphreys and his fellows has indicated that a software make or buy decision can be made by considering five factors, including: weighting and classifying, examining technical skills, comparing internal and external know-hows, analyzing supplier institutional abilities, and whole acquisition cost study [29]. M. G. Jacobides and S. Billinger has proposed that a firm might buy more universal, more straightforwardly exchanging commodities apparatuses whereas manufacture the rest in-house [30]. Further, this study explored the economics theories and relates when firms make, buy, and parallel source over trickery minor manufacturing firms.

III. RESEARCH QUESTIONS AND OBJECTIVES

From the above discussion, it is evident that the make or buy decision for software development is not an easy one and requires ample consideration from the unified activity of the organization's management and technologists [31]–[33]. The software projects fail due to reasons that are mostly known but not easily controllable due to the cost factor associated with the risk management. [34]–[40] has elaborated on the reasons of software failures. The outsourcing of the software projects, however, provides the sense of relaxation to the client firm as the risk of software failures shifts to the outsourcing partner. From the above

discussion, some research questions have emerged which are as following:

RQ 1: What are the parameters, used to take a 'build or buy' decision for a software component?

RQ 2: How the build versus buy decision is made in software projects?

RQ 3: How the make versus buy decision is made in large scale software projects?

The objective of this study is to identify the parameters and the process of making the make vs buy decision for the software development in general and large scale software development activity in particular. The study is conducted to meet the following objectives.

A. To identify the parameters for making make vs buy decision for software components
B. To identify the decision making process in buying or making the software
C. To identify the decision making process in buying or making the large scale software

IV. FACTOR IDENTIFICATION

Later in this section, in Table 2 and Table 3, factors are presented for identifying the make vs buy decision for large scale software projects. The selection criteria for the selection of the related model and approaches went through a systematic review process. In this regard, following statistics were used. The search on Google scholar was initiated with the term (''software risk management'' approaches for large scale projects), this returned 2150 result, while the time span was left open. In the next iteration, we limited the time span from 2000-2017, which resulted in a reduction of 300 papers. In the third iteration, we used the term (''software risk management'' approaches for ''large scale projects'') to obtain the relevant papers discussing the risk management approaches for the large scale projects only. This resulted in significant reduction of the papers which was further purified by eliminating the books, which resulted in 56 articles. When these 56 articles were searched individually, some, 18 were found either not available or not relevant. The study, therefore, is based on the 38 articles, and their information is given in Table 1, and in Figs. 1–3 and 5.

The paper is based on 38 core references and 6 supportive references. The information graph about the selected papers is given here.

In the process of the literature review, different factors have been identified that are deemed relevant in making a build versus buy decision. In Table 2, these factors are listed along with the source reference.

It is evident from Table 2 that the number of factors mentioned for taking the build versus buy decision for software consists of a list which is pretty long and some factors are possessing redundancy as they have been reported by more than one author. In order to eliminate the redundancy, the alike (not necessarily similar by words but also by theme) repeating factors have been gathered in one heading. The outcome is listed in Table 3.

TABLE 2. Factors for making 'build versus buy' decision.

F#	Factor	Reference
1	Strategy and competitive advantages	[18]
2	Cost of developing software (component)	[18]
3	Scale and complexity	[18]
4	Requirement fit	[18]
5	Time to develop	[18]
6	In-house development expertise	[18]
7	Risk elimination/transfer	[18]
8	Support structure	[18]
9	Operational factors	[18]
10	Intellectual property	[18]
11	Richness of the required functionality	[21]
12	Volatility of functional requirements	[21]
13	Organization's capacity for software maintenance	[21]
14	Legacy systems	[21]
15	Integrated application suites	[21]
16	Strive for superior quality	[21]
17	Ease of ready-made solution	[10]
18	Flexibility in design	[10]
19	Support and training	[10]
20	Enhanced flexibility through customer feedback	[10]
21	Time to determines business needs and development,	[10]
22	Constant staff engagement	[10]
23	Lower functionality and updates	[10]
24	Cost reduction	[23, 24]
25	Superior quality gain	[23, 24]
26	Need for flexibility	[23, 24]
27	Convenience of buying	[20]
28	Acquisition cost	[29]
29	Analyzing supplier's organization capability	[29]
30	Examining technical skills of employees	[29]
31	Comparing internal and external know-how	[29]
32	Software maintenance issues	[22]
33	Integration of software components	[22]
34	Risks in software development environment	[22]

V. METHODOLOGY

The outcome of the literature review is shown in Table 2 that comprises of 34 factors that are deemed suitable by different authors for making a build versus buy decision. In Table 3, a list of 16 factors has been finalized by merging similar factors. It is evident that some factors recurred for some 4 or 6 times while quite a few appeared for only once. Most papers cited in this regard are published during 2011-2016, while some have been published in earlier 2002 and 2003. Despite the fact, that the literature review is based on recent publications, the area is so dynamic that the decision parameters change with the innovations in the area.

In order to make study more reliable, it is important that the study is based on the recent literature, observations, and other methods used to conduct the study. The study has to answer two questions raised in section 1. In order to answer the

TABLE 3. Factors for making 'build versus buy' decision after merging similarities.

F#	Factor	Freq	Reference(s)
1	Strategy and competitive advantages	2	[18], [10]
2	Cost of developing software /component	3	[18], [23, 24],[29]
3	Scale and complexity	1	[18]
4	Requirement fit	4	[18],[21],[10]
5	Time to develop	1	[18]
6	In-house IS expertise	6	[18] ,[21],[10],[29]
7	Risk elimination/transfer	2	[18],[22]
8	Operational factors	1	[18]
9	Intellectual property	1	[18]
10	Legacy systems	1	[21]
11	Integrated application	2	[21],[22]
	suites		
12	Strive for superior quality	2	[21],[23, 24]
13	Ease of ready-made solution	2	[10],[20]
14	Flexibility in design	3	[10],[23, 24]
15	Support and training	2	[10],[22]
16	Analyzing supplier's organization capability	1	[29]

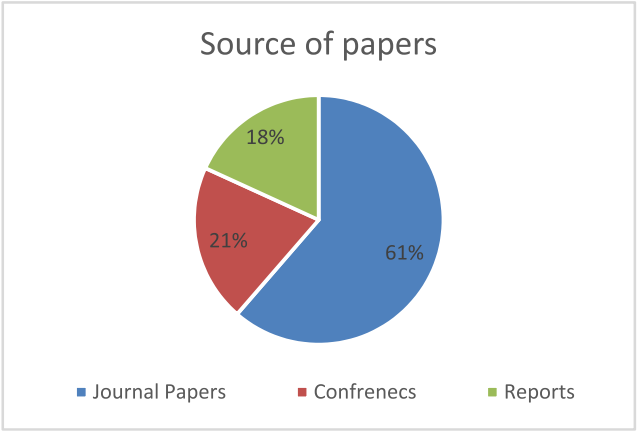

FIGURE 2. Source and number of selected papers.

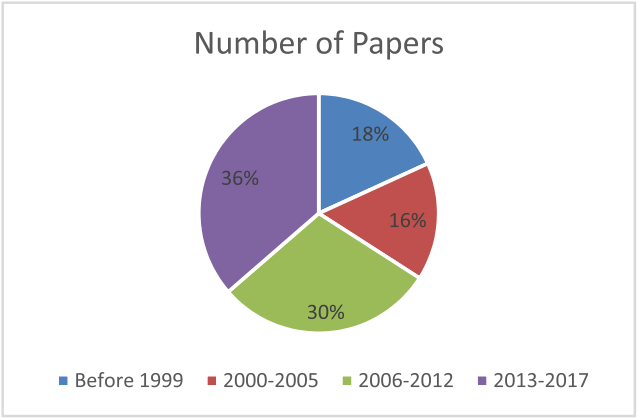

FIGURE 3. Source and number of selected papers.

Question 1 and 2, we conduct a mixed method research based on the factors identified in Table 3. The mixed method, multi-method or hybrid research method [41], [42] is a combination of qualitative and quantitative methods that are triangulated later. The quantitative studies are conducted by using any of the quantitative methods, including surveys and the card sorting technique. The qualitative study on the other hand is conducted with the help of interviews from the domain experts. In this article, we use the survey to conduct the quantitative study while later we conduct interviews for the purpose of the qualitative study. The interviews are conducted with academicians and practitioners of immense maturity to get an in-depth knowledge of the questions under observation. The survey questions are prepared in accordance with the survey guidelines. The interviews can be structured, semi structured or open. In structured interview, the questions and their sequence are strictly followed while in the open interviews the questions sequence and core topic is left open and it is conducted as more of a discussion [43]. The semi-structured interview method, in contrast, is focused and flexible [43]. The interviewer is limited to ask the questions on the topic and a structure is loosely followed in a way that the interviewer only asks out of the box questions when he needs some clarification or needs in-depth response. Since, this study is focused to address the questions in Sect. 1, we used the semi-structured interview method that is expected to take the benefits of the power of semi-structured method's flexibility and consistency during the interviewing process.

[44] emphasizes on three different dimensions of the mixed method research, namely mixing, time and emphasis. The mixing dimension determines whether we use partially mixed method or fully mixed method for the research [45], [46]. The time dimension considers that if the mixing is concurrent or sequential, the third dimension demonstrates that whether equal or dominant status is used. In this research study, we use partially mixed-sequential-dominant status research methodology. In this regard, the quantitative and qualitative studies will be done in a sequence. The triangulation process [47] identifies whether findings of both methods are consistent or not? If the findings are consistent, they are merged. If the findings of both methods are different, the findings of the quantitative method will prevail and the finds of the qualitative method will be dropped because we are following the sequential-partially mixed-dominant status research paradigm, as shown in Figure 4.

VI. QUANTITATIVE STUDY (DESIGN, CONDUCT, AND RESULTS)

Quantitative studies can be conducted by surveys, card sorting, or other techniques like voting. Card sorting method is

TABLE 4. Cases for build versus buy decision.

Case	Make	Outsource	In-house	Buy	C.CoTS	C.MoTS	Remarks
1	√	√					Opt to make and outsource
2	√		√				Opt to make and develop in-house
3				√	√		Opt to buy and buy complete CoTS
4				√		√	Opt to buy and buy complete MoTS

TABLE 5. Interviewee's profile.

No	Experience Type	Level	Education	Software Development Experience	Experience in years
1	Academic	Professor	PhD	Yes	27
2	Industry	Application Manager	MSc	Yes	15
3	Industry	Business Analyst	MSc	Yes	18
4	Industry	Project Manager	BSc	Yes	17
5	industry	Team Leader	MSc	Yes	12

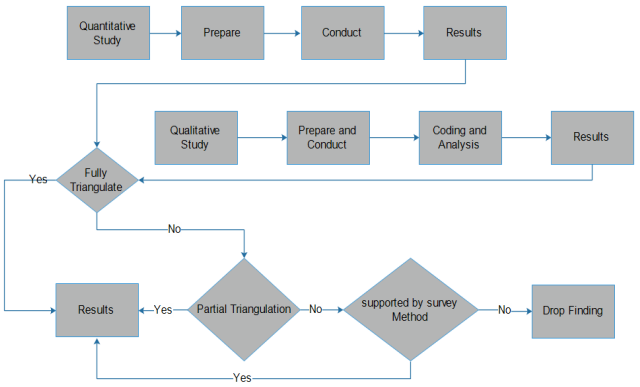

FIGURE 4. Conduct of mixed method research and the triangulation process.

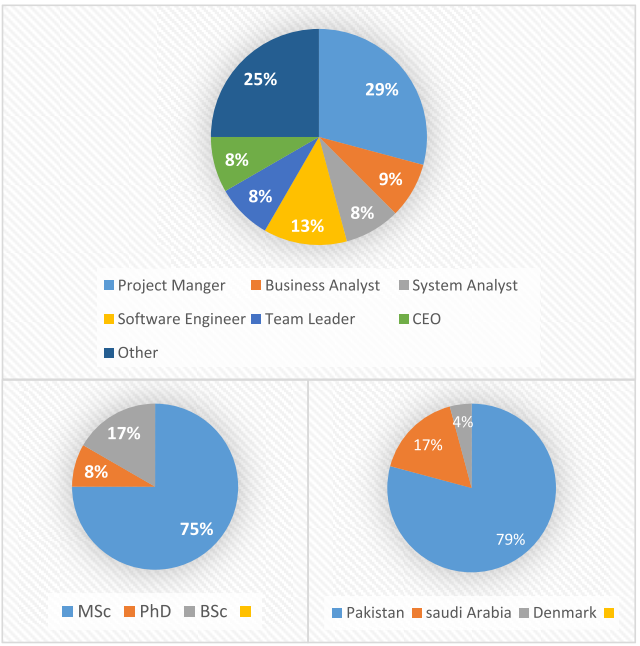

FIGURE 5. Demographic information of survey respondents.

used in a much focused environment where the number of respondents is very small, normally 8-10. Since the magnitude of this study is larger, card sorting is not a right method to adopt. In this study, the survey technique has been used as a quantitative study tool to help addressing a large number of individuals and taking their opinion. In order to implement the survey method, random sampling technique is used among the identified population.

A. SURVEY DESIGN AND CONDUCT

The criteria for the survey respondent, is to have experience in software development and minimum bachelor's degree. The survey was sent to 50 respondents and the means for spreading the survey were the google talk link forwarding, telephone calls, Facebook group requests, Twitter request forwarding etc. The respondents were responsive at large and valuable responses were received from them. In response to the 50 targeted requests, 24 responses were received from different countries. Out of these 24 responses, 19 were received from Pakistan, 4 from Saudi Arabia, and 1 from Denmark. Since the sample size from Saudi Arabia and Denmark is too small, we focus this study to demonstrate the trends in build versus buy decision from Pakistan's software industry.

It is important that in order to gauge the diversity and influence of the make vs buy decision, the respondents of the quantitative study are capable enough to provide an adequate opinion. In this regard, the authors agreed to spread the survey among the potential respondents meeting the minimum criteria of education and have a significant decision making role in the organization. It is important to note that 83% of the respondents have graduate or postgraduate degree while 79% of the respondents are from Pakistan. The survey consists of 22 questions and is presented in the Annex-II. The survey was responded by 24 respondents. In Table 4, we present the average response of the respondents.

VII. QUALITATIVE STUDY (DESIGN, CONDUCT, CODING PROCESS, RESULTS)

The interviewing mechanism was used as the tool for conducting the qualitative study. The interview design guidelines have been taken into consideration before preparing the interview's structure. Since the semi structure interview is used, the question formulation and restructuring is done if needed after the answer of a question from the interviewee.

TABLE 6. Proposed factors.

F#	Factor
1	Strategy and competitive advantages
2	Cost
3	Scale and complexity
4	Requirement fit
5	Time
6	In-house IS expertise
7	Risk
8	Support structure
9	Operational factors
10	Intellectual property
11	Customization
12	Any other factor deemed suitable?

A. INTERVIEW DESIGN

The interview questions are given in Appendix 1. The profiles of five interviewees are given in Table 5. The process of coding the interview transcripts is performed by doing the inductive coding. The coding is called inductive when the categories are prepared by the analyst while in deductive coding the categories are formed by the historical data [48]. The analysis of the interview contents requires a systematic method deployment consisting of the eight activities, as mentioned by De Wever [49] , Elo [50], and Lewis [51]. These activities include: preparing data, defining unit of analysis, developing coding scheme, testing the coding scheme, coding the transcripts, assessing the coding consistency, drawing conclusions, and reporting the results.

VIII. MIXING THE RESULTS AND TRIANGULATION PROCESS

In Tables 7 and 8, the qualitative and quantitative results have been presented. The triangulation and analysis processes identify the overlapping areas which can triangulate. The respective findings are discussed here. It is important to note that not all the questions asked during the interviews and surveys contribute in the triangulation process as some (sometimes several) questions are asked to retrieve the information for the interviewee and the survey respondent. Excluding the introductory, demographic, and repeating questions, we identify that the questions asked in the table below form the crux of the study. We map the respective questions in the survey and the interview.

The elimination of the repeating and the foundation questions, limits us to twelve core questions that help us meeting the basic objective of the study presented in this article. In the table below, we observe the triangulation of the resultant data from qualitative and quantitative studies. The triangulation is a process of identifying whether results of two studies converge to similar findings or not. The findings of two studies may or may not triangulate necessarily.

If the findings of both methods are exactly same or considerably alike (within the range of the difference defined by the researcher) the findings are considered to triangulate. This means that no difference is found in the individual findings and the findings are alike and consistent with each other. It is, however, not necessary that findings of the two studies necessarily match with each other by same values.

A. VALIDATION THREATS

It is common that the qualitative studies yield a different value while its counterpart quantitative yields a value which is not in the defined acceptable range. If finding of two approaches did not match it pose threats to the validation of data. In this case, findings of two studies do not triangulate. If the findings do not triangulate, the outcome/findings of the dominant study will prevail.

Another form of triangulation is the partial triangulation. In this form, results of both studies do not match completely but they are not also completely different. For example, if the qualitative study has 5 findings and the quantitative study has 8 findings, 3 or 4 may match while the rest exist without being similar. In partial triangulation, the matching/similar results are accepted and for the conflicting findings the findings of the dominant status study prevail. In this study, the quantitative method is considered dominant for its versatility and number of responses that it has. In this study, we assume $\pm 10\%$ acceptable range of difference. Hence, if the findings of qualitative and qualitative method differ by $\pm 10\%$ or less, we will consider that the findings are alike. If the difference is more than $\pm 10\%$, we will observe the partial or no triangulation, respectively.

IX. RESULTS AND DISCUSSION

The triangulation results presented in Tables 5 and 6 show that while some modules triangulate, others do not. As mentioned is Sect. 7, the outcomes are considered to converge to alike findings and triangulate, some results partially triangulate, while some do not triangulate at all. In order to analyses the

TABLE 7. Triangulation process.

Question	Qualitative Findings	Quantitative Finding	Triangulate?
Demographic Information			
1. Organization scale	80% [Large]	62.5% [Large]	Triangulate
2. Staff turnover	20%	20%	Partial
3. Experience of working abroad	80% [yes]	62% [yes]	Partial
4. Nature of software development	80% web and mobile	33% web applications , 37% business applications	Partial
Reasons of Not Developing software			
5. Risk transfer	60% [yes]	58.6% [yes]	Triangulate
6. Complexity of software	80% [yes]	75% [yes]	Triangulate
7. Strategic importance of software	80% [yes]	79.2 [yes]	Triangulate
8. Software maintenance	80% [yes]	75% [yes]	Triangulate
Outsourcing as a Solution			
9. Is outsourcing a solution?	60% [yes]	68.3% [yes]	Triangulate
10. What is probable destination?	Domestic 80% [yes]	Domestic 75% [yes]	Triangulate
Factors to make the Build versus Buy Decision			
11. Factors in making 'build versus buy' decision	• Cost of developing software 80%	• Cost of developing software /component 83.3%	Triangulate
	• Requirement fit 60%	• Requirement fit 62.5%	
	• Time to develop 80%	• Time to develop 87.5%	
	• In-house IS expertise 60%	• In-house IS expertise 62.5%	
	• Risk elimination/transfer 60%	• Risk elimination/transfer 58.3%	
	• Scale and complexity 40%	• Scale and complexity 54.2%	Partially Triangulate
	• Intellectual property 40%	• Intellectual property 20.8%	
	• Support and training 40%	• Support and training 20.8%	
	• Legacy systems 20%	• Legacy systems 12.5%	
	• Integrated application suites 20%	• Integrated application suites 8.3%	
	• Strive for superior quality 20%	• Strive for superior quality 41.7%	
		• Strategy and competitive advantages 37.5% • Operational factors 16.7% • Ease of ready-made solution 20.8% • Flexibility in design 12.5% • Analyzing supplier's organization capability12.5%	No

TABLE 8. Interview results.

Q#	Interviewee 1	Interviewee 2	Interviewee 3	Interviewee 4	Interviewee 5
1	27 years	15 years	18 years	17 years	12 years
2	Large	Large	Large	SME	Large
3	2000	20000	1000	20	more than 1000
4	20%	10-25%	20%	NA	15%
6	No Ex	Yes	Yes	Yes	Yes, as a consultant
7	Web & Mobile Application	Web & Mobile Application	Web & Mobile Application	Web application	Web, mobile and desktop applications
8	Outside	Yes (KSA), in-house	Yes (Outside)	Yes Out side	Yes
9	Cost & Complexity, Risk Transfer	yes	Strategy and Risk Transfer	Strategy	yes they do, cost, time, risk
10	NA	Yes	Yes	NA	Yes
11	NA	All factors important The decision making should be documented	make decision irrespective of the comparisons	Unity 3D	The cost of modules and tasks are compared
12	NA	yes complexity	yes	Yes bought outside	yes
13	NA	Risk of Technology	NA	Yes	yes, mainly
14	NA	Yes, major factor	yes	Yes for understanding source code	yes
16	outsource	not buy all time	outsource	inside build	yes
17	Domestic	Foreign	Domestic	domestic	Domestic
18	Personal Business	Budget	both side challenges	vender	both have their in-build challenges
19	faster/cheaper	Depend on client	NA	purchase through vendor	NA
20	NA	Make, examples (KSU website, KSU repository)	Make	Unity 3D, MS office	Make
21	• Cost to develop • Time to develop, • Scale, • Risk Transfer, • Support structure, • IP, • Customization • Requirement fit	• Cost to develop, • Scale and complexity, • Requirement fit, • Time to develop, • In-house IS expertise, • Support structure, • customization, • Team Requirement	• Cost to develop • Time to develop • Intellectual property • Legacy systems • Integrated application suites • Strive for superior quality • Strategy and competitive advantages	• In-house IS expertise • Risk transfer	• Cost to develop, • Time to develop, • risk transfer, • in-house expertise

TABLE 9. Survey results.

Q#	Q						
1	Qualification	PhD 8%	MSc 75%	BSc 17%	Other		
2	Experience (years)	<5 years 12.5%	>5 and <10 41.7%	>10 and <20 37.5%	>20 8.3%		
3	Role	Project Manager 29.2%	Business Analyst 8.3%	System Analyst 8.3%	Software Engineer 12.5%	Team Leader 8.3%	
		CEO 8.3%	Other 25%				
4	Respondents Origin	Pakistan 79%	Saudi Arabia 17%	Denmark 4%			
5	Organization Scale	Large 62.5%	SME 33.3%	Small 4.2%			
6	Employee vs Scientific task	>70% 70.8%	>50% 12.5%	>20% 16.7%			
7	Staff Turnover	20% 58.3%	40% 16.7%	60% 12.5%	80% 0%	Other 12.5%	
8	Experience of Working abroad	Yes 62.5%	No 25%	May be 8.3%	Other 3.2%		
9	Nature of software	Business Applications 37.5%	Multimedia Applications 0%	Web Applications 33.3%	Healthcare Applications 12.5%	Other 16.7%	
10	Factors						
11	Impact of strategic importance	Yes 79.2%	No 4.2%	Maybe 16.7%	Other 0%		
12	System Complexity	Yes 75%	No 0%	Maybe 25%	Other 0%		
13	Risk Transfer	Yes 50%	No 33.3%	Maybe 16.7%	Other 0%		
14	Maintenance	Yes 75%	No 8.3%	Maybe 16.7%	Other 0%		
15	Outsourcing a solution	Yes 58.3%	No 20.8%	Maybe 20.8%	Other 0%		
16	Outsourcing destination	Domestic 75%	International 16.8%	Other 8.3%			
17	Outsourcing parameters	Cost 12	Risk Reduction 3	Time to Market 10	Task automation 1	Paperless 1	Expertise 7
		Requirement fit 1	Customization 3	Ease of use 1	Maintenance 3	Physical resources 1	Complexity 2

findings of the study, we divide them in four sections and analyze them respectively.

A. DEMOGRAPHIC INFORMATION

The demographic information, in itself, does not make any significant decision, however, by having the significant demographic information the results of the main study become more trustworthy. It can be observed from Table 5 that 80% of the interviewees and 62% of the survey respondents belonged to the large scale organization where the staff turnover is around 20% and 80% respectively. 62% employees working in these organizations also have the experience of working abroad according to the interviews and survey findings. The nature of software development done by the interviewees and the respondents portrays a divided picture, where the interviewees look more involved in the web and mobile applications while the respondents were more involved in the web and business application development. Partial triangulation of results is identified in this section.

B. REASONS FOR NOT DEVELOPING SOFTWARE

There are number of reasons that govern the possibility of developing software or buying from the market. There were identified reasons that were collected from literature and presented in Table 3. The outcome of the mixed method study and the triangulation process, endorses the existence of these factors as deciding factors for making the build versus buy decision for software. It is important to note that all factors in this section triangulated completely (i.e., findings from both methods did not differ from each other by more than $\pm 10\%$).

The outcome of the triangulation process suggests that the software maintenance, strategic importance of software, and complexity of software are among the main causes of buying software instead of building. For these three factors, the interviewee's support was 80% for all the factors while the survey respondents, responded with 75%, 79.2%, and 75% respectively. Another important factor in this regard was risk transfer that is supposed to be an important component in making the build versus buy decision. However, the support for this factor was less than the other three factors, and it was endorsed by 60% interviewees and 58% survey respondents. Complete triangulation of results is identified in this section.

C. OUTSOURCING AS A SOLUTION

The results of the study demonstrate that the support for buying software from market instead of building is promising. In this regard, the results of the study demonstrate that outsourcing the software development is also vital to transfer the risks during the software development. While the respondents and the interviewees are pro to outsourcing, yet they believe that the outsourcing should be domestic. This approach has the economic benefits for the national economy. Complete triangulation of results is identified in this section.

D. FACTORS TO MAKE THE BUILD VERSUS BUY DECISION

In this subsection, 16 factors are present. Out of these 16 factors, five factors completely triangulate while six factors partially triangulate. Five other factors do not triangulate at all. In the design of the study, it was mentioned, that we are following sequential-partially mixed-dominant status design of the mixed method study. In this study, the dominant status has been given to the quantitative study, which means that if identification is done through the quantitative method but not supported by the qualitative method, it still holds its existence. In this regard, the study concluded with the identification of 16 factors that contribute in making the build versus buy decision for the software development.

Two research questions have been addressed in the study. The first question was about the identification of the factors required to make the build versus buy decision while the second question was about the process of making this decision for large scale projects. Research question 1 is answered by the outcome presented in Table 5. The factors identified in Table 5, affect the decision making process in making the make vs buy decision of the software. The factors have deep influence in gauging the impact of multiple dimensions including the managerial, technical, social, environmental, development, political, marketing, and socio-technical aspects of the process. The research question 2 is answered by the fact that the study has been conducted to address large-scale software projects only and the decision of making software (components) or buying them is influenced by the factors presented in Table 5.

X. ASSUMPTIONS

This study focuses on identifying the prominent factors that have a vital role in making the build vs buy decision for the large scale software projects. There are some assumptions that we have considered in this study. We assume that, while initiating the process of deciding between building or buying software, the decision makers are well aware of market needs, pricing scenarios, profit analysis, and have a good command on software economics. A decision not consistent with the market or taken without considering the market opportunities may turn out to be infeasible. It is also assumed that the individuals on the decision making positions are aware of the market trends in software development and have adequate knowledge of the economy, technology, and software development environment.

XI. CONTRIBUTIONS

There are three research questions raised in this study. In this paper, the questions have been addressed by adapting the mixed-method methodology, and the results have been discussed in section VIII. The outcome of the study, constitutes the following contributions in the existing knowledge:

- It determines the important factors that are considered in making the build vs buy decision for a software project.
- It elaborates the process of making the decision of build vs buy for the software projects by considering the multi-dimensional software dynamics.
- The process of decision making for the build vs buy is elaborated for the large-scale software projects.

XII. CONCLUSION

In this study, the case for large-scale application development was considered. It has been identified that there are several reasons for software firms to decide in favor of buying software instead of developing that may include but not limited to the software development risk transfer, the hardship in providing the software maintenance support, and the strategic importance of software. The core question of the factors identification for making the make vs buy decision was considered along with the process of making these decisions. This is important to note that the 'buying' option is more viable for the software houses that are not heavily resourced and buying software does not make the firm-level resource wastage. However, at large scale application development firms, which have dedicated resources, may not have the structural liberty of buying software because of the above-mentioned reasons; rather the buying decision is made because of cost and resource intensity. The study also aimed at identifying the other possible factors that contribute in making the build versus buy decision, and several contributing factors were identified. Some of these factors are time, cost, quality, in-house expertise, and customization dimensions of the software development activity. These factors, can be used as a reference, to make the build versus buy decision.

REFERENCES

[1] CMMI Product Team. (2006). "CMMI for development, version 1.2: Improving processes for developing better products and services." Software Engineering Process Management Program. Tech. Rep. CMU/SEI-2010-TR-033 ESC-TR-2010-033. [Online]. Available: http://www.sei.cmu.edu

[2] B. Curtis, B. Hefley, and S. Miller, "People capability maturity model (P-CMM) version 2.0," Carnage Melon Univ., Pittsburgh, PA, USA, Tech. Rep. ADA512354, 2009.

[3] M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for Development: Guidelines for Process Integration and Product Improvement*. London, U.K.: Pearson Education, 2011.

[4] C. Ebert, "Looking into the future," *IEEE Softw.*, vol. 32, no. 6, pp. 92–97, Nov./Dec. 2015.

[5] P. M. Khan and M. Beg. (May 2014). "Measuring cost of quality (CoQ) on SDLC projects is indispensible for effective software quality assurance." [Online]. Available: https://arxiv.org/abs/1405.4824

[6] N. Nahar and P. Student, "Managing requirement elicitation issues using step-wise refinement model," *Int. J. Adv. Studies Comput., Sci. Eng.*, vol. 2, no. 5, pp. 27–33, 2013.

[7] W. M. Wong and M. C. Hui, "Method and system for modeling and analyzing computing resource requirements of software applications in a shared and distributed computing environment," U.S. Patents 8 584 131 B2, Nov. 12, 2016.

[8] T. Ambreen, N. Ikram, M. Usman, and M. Niazi, "Empirical research in requirements engineering: Trends and opportunities," *J. Requirements Eng.*, pp. 1–33, 2016. [Online]. Available: https://link.springer.com/article/10.1007/s00766-016-0258-2

[9] O. Gomez, P. Wriedt, and F. Zhao, "Build or Buy: A case study for ERP system selection in SMEs," in *Proc. Int. Conf. Human-Comput. Interaction*, 2016, pp. 23–33.

[10] A. Wilson. (Apr. 10, 2014). *Build vs. Buy: Should You Buy Software or Develop it Internally?* [Online]. Available: http://blog.kainexus.com/continuous-improvement-software/build-vs-buy/should-you-buy-software-or-develop-it-internally

[11] S. Sundararajan, M. Bhasi, and P. K. Vijayaraghavan, "Case study on risk management practice in large offshore-outsourced Agile software projects," *IET Softw.*, vol. 8, no. 6, pp. 245–257, 2014.

[12] J. Gong, J. Lu, and L. Cai, "An induction to the development of software quality model standards," in *Proc. 3rd Int. Conf. Trustworthy Syst. Their Appl. (TSA)*, 2016, pp. 117–122.

[13] M. Waseem and N. Ikram, "Architecting activities evolution and emergence in agile software development: An empirical investigation," in *Proc. Int. Conf. Agile Softw. Develop.*, 2016, pp. 326–332.

[14] T. Rands, "The key role of applications software make-or-buy decisions," *J. Strategic Inf. Syst.*, vol. 1, no. 4, pp. 215–223, 1992.

[15] T. Rands, "A framework for managing software make or buy," *Eur. J. Inf. Syst.*, vol. 2, no. 4, pp. 273–282, 1993.

[16] S. Malbašić, L. Tančić, and V. Petrović, "Technology risk assessment as part of risk management process," *Serbian Project Manage. J.*, vol. 6, no. 1, pp. 51–95, 2016.

[17] R. Colomo-Palacios, Ed., *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement*. Hershey, PA, USA: IGI Global, Feb. 2014.

[18] F. Daneshgar, G. C. Low, and L. Worasinchai, "An investigation of 'build vs. buy' decision for software acquisition by small to medium enterprises," *Inf. Softw. Technol.*, vol. 55, no. 10, pp. 1741–1750, 2013.

[19] P. G. Klein, "The make-or-buy decisions: Lessons from empirical studies," in *Handbook of New Institutional Economics*. Springer, 2008, pp. 435–464.

[20] P. A. Usmanij, R. Khosla, and M.-T. Chu, "Successful product or successful system? User satisfaction measurement of ERP software," *J. Intell. Manuf.*, vol. 24, no. 6, pp. 1131–1144, 2013.

[21] *Make or Buy Decision Factors: World Bank Experience and Guidance Note for Treasury Management System*, World Bank, Washington, DC, USA, 2016.

[22] D. Margan and S. Čandrlić, "The success of open source software: A review," in *Proc. 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, 2015, pp. 1463–1468.

[23] J. Rundquist, "The interactive decision when outsourcing new product development," Tech. Rep., 2011.

[24] V. Chiesa, R. Manzini, and F. Tecilla, "Selecting sourcing strategies for technological innovation: An empirical case study," *Int. J. Oper. Prod. Manage.*, vol. 20, no. 9, pp. 1017–1037, 2000.

[25] J. Karlsbjerg, J. Damsgaard, and R. Scheepers, "A taxonomy of intranet implementation strategies: To make or to buy?" in *Advanced Topics in Global Information Management*. Piscataway Township, NJ, USA: IGI, 2005, p. 179.

[26] L. Legorreta and R. Goyal, "Managing risks of it outsourcing," in *Outsourcing Management Information Systems*. New York, NY, USA: ACM SIGMIS Database, 2007, pp. 242–269.

[27] A. Moses and P. Åhlström, "Dimensions of change in make or buy decision processes," *Strategic Outsourcing, Int. J.*, vol. 1, no. 3, pp. 230–251, 2008.

[28] C. Fill and E. Visser, "The outsourcing dilemma: A composite approach to the make or buy decision," *Manage. Decision*, vol. 38, no. 1, pp. 43–50, 2000.

[29] P. Humphreys, R. McIvor, and G. Huang, "An expert system for evaluating the make or buy decision," *Comput. Ind. Eng.*, vol. 42, no. 2, pp. 567–585, 2002.

[30] M. G. Jacobides and S. Billinger, "Designing the boundaries of the firm: From 'make, buy, or ally' to the dynamic benefits of vertical architecture," *Org. Sci.*, vol. 17, no. 2, pp. 249–261, 2006.

[31] R. Sharma, "Quantitative analysis of automation and manual testing," *Int. J. Eng. Innov. Technol.*, vol. 4, no. 1, pp. 252–257, 2014.

[32] A. I. Ianov, H. Kawamoto, and Y. Sankai, "Development of noise resistant hybrid capacitive-resistive electrodes for wearable robotics, computing and welfare," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 4249–4254.

[33] P. Marques, P. F. Cunha, F. Valente, and A. Leitão, "A methodology for product-service systems development," *Proc. CIRP*, vol. 7, pp. 371–376, Jan. 2013.

[34] A. S. Al-Mudimigh, B. Shahzad, and Z. Ullah, "Identification of methodology for analysis of the risk factors in software development environment," *Global J. Comput. Sci. Technol.*, vol. 10, no. 1, pp. 1–11 2010.

[35] H. I. Mathkour, B. Shahzad, and S. Al-Wakeel, "Software risk management and avoidance strategy," presented at the Int. Conf. Mach. Learn. Comput., Singapore, 2011.

[36] B. Shahzad and S. A. Safvi, "Effective risk mitigation: A user prospective," *Int. J. Math. Comput. Simul.*, vol. 2, no. 1, pp. 70–80, 1998.

[37] B. Shahzad, A. S. Al-Mudimigh, and Z. Ullah, "Risk identification and preemptive scheduling in software development life cycle," *Global J. Comput. Sci. Technol.*, vol. 10, no. 2, pp. 1–9, 2010.

[38] B. Shahzad and J. Iqbal, "Software risk management-prioritization of frequently occurring risk in software development phases. Using relative impact risk model," in *Proc. 2nd Int. Conf. Inf. Commun. Technol. (ICICT)*, 2007, pp. 16–17.
[39] B. Shahzad and S. A. Safvi, "Risk mitigation and management scheme based on risk priority," *Global J. Comput. Sci. Technol.*, vol. 10, no. 54, pp. 1–6, 2010.
[40] B. Shahzad and A. M. Said, "Identification of software risk factors in large scale projects: A quantitative study," *Arch. Des Sci.*, vol. 65, no. 12, pp. 47–58, 2012.
[41] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, CA, USA: Sage Publications, 2013.
[42] M. Zohrabi, "Mixed method research: Instruments, validity, reliability and reporting findings," *Theory Pract. Lang. Studies*, vol. 3, no. 2, p. 254, 2013.
[43] S. Mann, "Research interviews: Modes and types," in *The Research Interview*. Springer, 2016, pp. 86–113.
[44] N. L. Leech and A. J. Onwuegbuzie, "A typology of mixed methods research designs," *Quality Quant.*, vol. 43, no. 2, pp. 265–275, 2009.
[45] V. Venkatesh, S. A. Brown, and Y. W. Sullivan, "Guidelines for conducting mixed-methods research: An extension and illustration," *J. Assoc. Inf. Syst.*, vol. 17, no. 7, p. 435, 2016.
[46] O. Bentahar and R. Cameron, "Design and implementation of a mixed method research study in project management," *Electron. J. Bus. Res. Methods*, vol. 13, pp. 3–15, Dec. 2015.
[47] A. Hussein, "The use of triangulation in social sciences research: Can qualitative and quantitative methods be combined?" *J. Comparative Social Work*, vol. 4, no. 1, pp. 1–12, 2015.
[48] J. L. Campbell, C. Quincy, J. Osserman, and O. K. Pedersen, "Coding in-depth semistructured interviews: problems of unitization and intercoder reliability and agreement," *Sociol. Methods Res.*, vol. 42, no. 3, pp. 294–320, 2013.
[49] B. De Wever, T. Schellens, M. Valcke, and H. Van Keer, "Content analysis schemes to analyze transcripts of online asynchronous discussion groups: A review," *Comput. Edu.*, vol. 46, pp. 6–28, Jan. 2006.
[50] S. Elo, Kääriäinen, O. Kanste, T. Pölkki, K. Utriainen, and H. Kyngäs, "Qualitative content analysis," *SAGE Open*, vol. 4, pp. 1–10, Jan. 2014.
[51] S. Lewis, "Qualitative inquiry and research design: Choosing among five approaches," *Health Promotion Pract.*, vol. 16, no. 4, pp. 473–475, 2015.

BASIT SHAHZAD received the M.Sc. degree from the National University of Science Technology, Islamabad, Pakistan, the Ph.D. degree from University Technology Petronas, Malaysia. He is currently a Visiting Scientist with the University of Cambridge, U.K., and the Visiting Fellow at Macquarie University, Australia. He is a Collaborating Researcher with the Hagenberg Centre for Software Competence, Austria. His research and teaching career spans over 16 years. He is currently an Assistant Professor with the Faculty of Engineering and Computer Science, NUML, Islamabad. He has been with King Saud University, Riyadh and was an Assistant Professor with the COMSATS Institute of Information Technology, Islamabad. He is the Reviewer of several high impact journals and is in the program committee of several distinguished conference. He values research a lot and his career objectives revolve around conducting and supervising research in information systems (enterprise architecture, software cost and risk modeling, mitigating technological risks in modern banking), advancements in research methodologies (quantitative, qualitative, mixed method), mobile healthcare, and social networks (theory, empirical social network analysis, micro-blogging based analysis). He has numerous publications in journals and conferences of international repute and has a very active research profile. He has an Editorial Role in several conferences and journals of high repute and has edited numbers of special issues in significant journals in the areas of software engineering, social networks, and mobile healthcare.

ABDULLATIF M. ABDULLATIF received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K., 2011. His area of interest is social network analysis and software engineering. He is currently serving as an Assistant Professor with King Saud University. He has numerous publications in journals and conferences of international repute and has a very active research profile.

NAVEED IKRAM (SM'16) is currently an Associate Professor with Riphah International University, Islamabad, Pakistan. He is a chartered IT professional, holding the Ph.D. degree in computer science from the University of Salford, U.K. He has over 25 years of mixed experience with in academia and industry. He has expertise in Empirical Software Engineering and has published over 50 articles in peer-reviewed conferences and journals. He is a Senior Member of ACM.

ATIF MASHKOOR received the master's degree in computer science from Umeå University, Sweden, and the Ph.D. degree in computer science from the University of Lorraine, France. He is currently the Scientific Head of the Rigorous Methods with Software Engineering Research Focus, Software Competence Center Hagenberg, GmbH. He has several years of experience in rigorous systems and software engineering ranging from domain modeling to testing of executable code. He has taught with the University of Lorraine, France, COMSATS Institute of Information Technology, Pakistan and the National University of Modern Languages, Pakistan. He was also a Research Associate with the University of Minho, Portugal. Additionally, he has studied computational linguistics with Rovira i Virgili University, Spain.