Adoption, Implementation and Management of Self-Adaptive and Self-Managing Software Systems in Pakistan

Shahbaz Rahim, Ali Ahsan, and Sara Siddique

***Abstract*—Self-adaptive and self-managing software systems have become a burning topic in recent years. Self-adaptability is being suggested as an effective and innovative approach to adapt and manage software systems to the changing operational environment. In this paper, we attempt to investigate and explore the adaptation approaches, types, and need for new engineering processes for self-adaptive software systems. The understanding of the fundamentals of a self-adaptive system and its properties is a prerequisite to engineering a dynamic software system. Examination of several aspects has helped in identify the aspects and dynamics related to software engineering that needs to be considered for Adoption, Implementation and Management of these systems in Pakistan.**

***Index Terms*—Self-adaptive, self-managing, software systems, software engineering.**

I. Introduction

With the unceasing evolution of information technology and software-intensive systems, need for new and innovative approaches to engineer these software systems is becoming increasingly important. To be able to change with changing environments, system requirements and operational contexts, software systems have to be robust, customizable, configurable, energy-efficient, self-optimizing at runtime and autonomic. For this purpose, systems with the ability to self-manage and self-adapt are being proposed [1]. The ability of a system to self-adapt and self-manage is becoming more and more significant for software-intensive systems.

By adapting at runtime to preserve its operations and manage changes in its operational environment like resource variability or change in user needs, a system must continually self-govern. Such systems with self-adapting and self-managing capabilities are employed in a variety of emerging software application areas including autonomic computing, cloud and mobile computing, and robotics. Furthermore, software solutions in other areas like artificial intelligence, machine learning, and control systems have also been proposed.

A. Background

The software systems continually being developed in the world play important role in the organization they are deployed in. These systems typically involve human supervision and human interaction to complete their operations. To engineer the systems with self-adapting and self-managing capability that cater the changes in runtime without requiring excessive human interaction, researchers and practitioners are looking for different ways to build, run and manage these software systems. These systems are different from the typical systems thus these cannot be handled using classic software engineering methodologies.

Typical software systems for different industries like healthcare, automobiles, and consumer electronics require interaction with the outside world but self-governing systems do not just interact with outside world, they also have the ability to reconfigure and act according to their changing operational environment.

B. Purpose

This research has been conducted to help explore and acquire an understanding of the self-adaptive and self-managing software systems, to investigate how these software systems differ from typical software systems, and the aspects and dynamics to be deliberated upon for adoption, implementation and management of these systems in Pakistan.

C. Scope

Scope of this research is limited to the self-adaptive and self-management software systems. This research explores and reviews the earlier work done in this area and the aspects and dynamics required to be catered for to design, build, implement and manage these systems for Pakistan. Development or prototype of these self-adaptive & self-managing software systems do not fall under the purview of this research.

D. Research Methodology

This research study is qualitative in nature and employs Delphi Technique for data collection and Content Analysis for interpretation of the data. Source of primary data collected for this study is software engineers currently working on software-intensive systems aiming to transition towards self-adaptive and self-managing software systems.

II. Literature Review

A. Self-Adaptive Software Systems (SASs)

Self-adaptive software systems are the ones that adapt themselves to their operating environment in response to any change that occurs [2]. These systems continually evaluate

Manuscript received November 29, 2017; revised December 26, 2017.

S. Rahim is with the Advanced Military Maintenance, Repair and Overhaul Center (AMMROC) LLC, Abu Dhabi, 93443 UAE (e-mail: shahbaz.se@gmail.com).

A. Ahsan and S. Siddique are with the Faculty of Business and Technology, Foundation University, Islamabad, 44000 PK (e-mail: al_ahsan1@yahoo.com, sara-sid@live.com).

doi: 10.18178/joebm.2017.5.12.542

their own behavior and modify it if they see that they are not performing optimally and are not accomplishing what they are intended to accomplish [3]. Self-adaptive software systems have the ability to evaluate their performance and reconfigure their operations at the runtime without any human intervention. Maintenance, upgradation, and repair tasks that are normally performed by humans, can be performed by software with self-* properties. This also lessens the cost of human labor and other costs associated with human error [4]. These autonomic systems take many names in literature with respect to their capabilities that are known as self-* properties. These include self-configuring, self-governing, self-healing, self-optimizing, self-protecting, self-awareness, self-adapting, etc. [5]

B. Self-Managing Software Systems (SMSs)

Concept of self-managing is very similar to the self-management in computer systems. The main purpose behind these systems is to remove the element of human interaction from their administration and management [6]. Following four capabilities, known as self-CHOP, describe the Self-* properties of these systems:

- Self-Configuring,
- Self-Healing,
- Self-Optimizing, and
- Self-Protecting.

C. Difference between Self-Adapting and Self-Managing Systems

The only difference between both these self-capable systems is that SASs take input form the environment in which they are working to adapt themselves with any change, whereas, SMSs are only capable of managing their internal operations without human intervention [7].

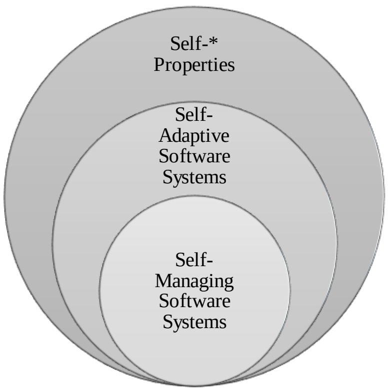

Fig. 1. Self-adapting and self-managing systems.

Fig. 1 shows that every self-adaptive system has some of the self-* properties and it has the capabilities of a self-managing system while self-managing system does not possess the capabilities of a self-adaptive system.

D. Adoption, Implementation and Management

First and the most crucial stage in software development lifecycle is that of requirement engineering [8]. Understanding the requirements and adaptation semantics is very critical for successful implementation of the self-adaptive and self-managing software systems. Adaptation semantics explain how systems behave during adaptation to changing environment in runtime [9]. Some of the requirement engineering activities have been focused and studied more than the others. These include requirement specification, verification, modeling and monitoring, and adaptation mechanism [10]. Some questions to address for requirements engineering are what changes to monitor, and what, when and how to adapt [11]:

- What changes to monitor, and
- What, when, and how to adapt

Feedback Loops are considered a noteworthy property and a key aspect in building of self-adaptive and self-managing software systems [12], [1]. A number of adaption loops like CADA, and MAPE-K are proposed in the literature [13], [14].

After Requirement Engineering approach, the next famous approach for developing self-adaptive systems is architecture based. In this approach, reconfiguration of systems is based on system's architecture models [15]. The Rainbow Framework is largely architecture-based design approach for engineering self-adaptive systems [16]. Rules for adaptation are required to continually monitor the operational environment of the system and to define the actions based on the monitoring results. Recommended language for this framework is ACME architecture description language as it allows the designer to define different architectural alternatives [17], [18]. A study suggests that a combination of requirements and architectural models should be employed to capture target system's adaptation needs [19].

To ensure a software system's reliability, Validation and Verification (V&V) is considered very important. V&V is required at runtime for quality assessment and to ensure confidence and certifiability of self-adaptive and self-managing software systems [1].

III. ASPECTS RELATED TO ENGINEERING OF SASS AND SMSS

Autonomic computing helps systems in adapting to changes. A number of different aspects play a role in adoption, implementation and management of SASs and SMSs. Focus is placed on the following major phases in engineering of these systems:

- Requirement Engineering
- Architecture Development
- Implementation and Testing

One key aspect is of adaptation that should be incorporated in SASs and SMSs. Adaptation transforms a system from conventional open-loop to closed-loop using feedback mechanisms. This feedback loop gives the system feedback about what is happening inside it (self) and in its environment (context).

TABLE I: QUESTIONS TO ELICIT THE ADAPTATION REQUIREMENTS OF SASS AND SMSS

Where	Where the need for change is?
When	When does the change need to be applied?
What	What attributes can be changed through actions about adaptation?
Why	Reason to resolve these systems (e.g. robustness, reliability).
Who	Level of automation and human involvement.
How	How the adaptable objects can be changed?

Another key aspect is the life-cycle of the system. Lifecycle of SASs and SMSs keeps evolving, evaluating and responding, and does not stop after development and deployment of the system.

Requirements of SASs and SMSs can be recorded by asking 5Ws and 1H questions. Questions in Table I are considered vital in drawing adaptation requirements [20] and need to be answered during the two phases of these sytems. These phases are developing phase and operating phase. Developing phase is concerned with the development and implementation of the system by answering the six questions in order to develop the capability of self-adaptiveness and self-management in the system. Operating phase is concerned with the operational issues of self and context. Decisions taken regarding the type and method of adaptation in the first phase affect how the system adapts itself in the second phase.

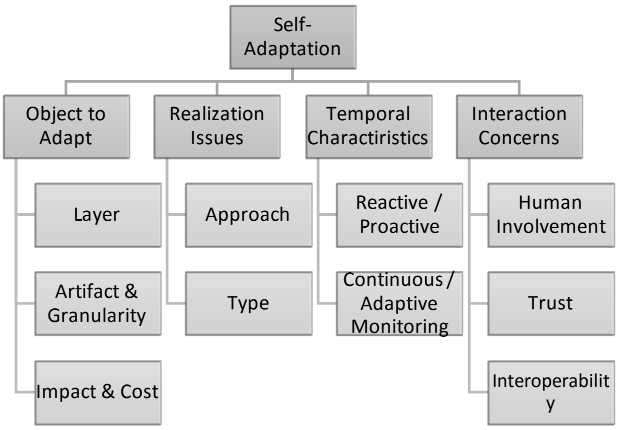

Fig. 2. Classification of self-adaptation.

TABLE II: ASPECTS OF ADAPTATION, IMPLEMENTATION AND MANAGEMENT OF SASS AND SMSS

Aspects	Questions	Description
Objects to Adapt	*Where, What*	This deals with the 'where' and 'what' aspects of the change.
Realization / Awareness Issues	*How*	This deals with methods and techniques that can be used to add adaptation. **Adaptation Approaches:** Deals with adding adaptation into the system using different approaches like Static / Dynamic Decision-Making, External / Internal Adaptation. **Adaptation Types:** Specifies adaptation type employed like Open / Close, Model-Based / Free, Specific / Generic.
Time-based Characteristics	*When*	Deals with 'when' of the required change. **Reactive / Proactive Adaptation:** Controls the anticipatory properties, and the detection and deciding processes. **Continuous / Adaptive Monitoring:** Captures monitoring process and its effects on the cost and detection time.
Interaction	*Where, When, What, How, Who*	Addresses the interaction concerns. **Human Involvement:** Relate to who the agent of change is. **Interoperability Support:** Deals with interaction among components, subsystems or systems.
Supporting Disciplines	-	Broad aspect dealing with various disciplines related to and supporting the adoption, implementation and management of SASs and SMSs.

A. *Taxonomy of Self-Adaptation*

Starts from static approaches, self-adaptivity moves towards the dynamic ones [2]. These approaches combined with the technologies and techniques in this domain [21] form a classification to introduce aspects that contribute in engineering SASs and SMSs [20].

Fig. 2 shows the aspects related to self-adapting systems.

Table II shows that the aspects for adaptation, implementation and management of SASs and SMSs identified in the literature and shown in Fig. 2 are further mapped on to the requirement questions mentioned in the Table I.

B. *Chosen Aspects to be Studied*

TABLE III: CHOSEN ASPECTS OF ADAPTATION, IMPLEMENTATION AND MANAGEMENT OF SASS AND SMSS

Aspects	Reason
Realization / Awareness Issues (How)	This aspect plays a critical role in realization of Self-Adaptation of the system during its different phases of engineering like adoption, implementation, and management. **Adaptation Approaches:** This deals with the deciding processes and approaches to help achieve self-adaptation making it the backbone of SASs and SMSs. **Adaptation Types:** This deals with the types of adaptation employed. Every adaptation type comes with its own advantages and dis advantages that making this decision of selecting an adaptation type a very important one.
Time-based Characteristics (When)	The need of SASs and SMSs to react in runtime makes 'when to react' and 'how to react' the key decisions for engineering of these software systems. **Reactive/Proactive Adaptation:** The cost associated with these software systems make the decisions regarding detecting and deciding processes vital to ensure that system fulfills the criteria of self-adapting and self-managing. **Continuous/Adaptive Monitoring:** SASs and SMSs require more resources like sensors, agents, controller, etc. than typical software systems making the cost of monitoring and detection high.
Supporting Disciplines	Self-adaptive and self-managing software systems are interdisciplinary. The designs, approaches and methods adopted for engineering these systems depend on the combination of disciplines of these systems. Main disciplines are Artificial Intelligence, Distributed Computing, Control Theory, and Software Engineering.

IV. DATA COLLECTION

For data collection to gather valuable opinion of industry experts, Delphi Technique was employed. Questionnaire designed for this research consisted of mostly open-ended questions that were analyzed using Content Analysis.

The purpose behind the questionnaire designed for this research was to gather input from industry experts about the identified aspects, engineering, and challenges faced while engineering of SASs and SMSs. To cover the multidisciplinary fields, experts were targeted from different fields of Computer Sciences and Computer Engineering with main focus on Robotics and Autonomic Systems. Experts with at least three years of hands-on experience were selected.

Questions gathered and extracted from identified aspects of engineering SASs and SMSs focused on the techniques and approaches available in literature to engineer and manage these systems. In first step, designed questionnaire was given to experts for endorsement. After receiving a number of endorsements from experts, final questionnaire was floated among industry experts for their input. Final questionnaire consisted of 13 open-ended and 2 close-ended questions related to introduction of SASs and SMSs (Q. 1-3), adaptation approaches and types (Q. 4-6), temporal characteristics (Q. 7-8), interaction (Q. 9), supporting disciplines (Q. 10-11), and future of SASs and SMSs in Pakistan (Q. 12-15).

Data was gathered from an expert panel chosen from industry and research professionals using the questionnaire. Final questionnaire is given in the Appendix A.

Fig. 3. Role of expert panel.

Fig. 3 shows the percentages of participants in expert panel with respect to their role.

V. Analysis and Discussion

After receiving the responses from expert panel, Common consensus was reached on the first introductory question that these systems keep evolving with the changes in their environment whereas other typical systems act as per their original program. These systems do not need human intervention or supervision for effective functioning. These systems can adapt themselves to deal with the changes not known during system design.

When asked if ‘drone’ and ‘simple robot’ are SASs and SMSs, one-thirds of the participants of the panel responded that these are SASs and SMSs. This shows that there is lack in understanding towards these systems. These are not self-adapting systems. Half of the respondents thought ‘Mars Rover’ is not a self-adapting system.

In response to questions related to adaptation approaches and types, again the consensus was reached and experts suggested that Dynamic Decision-Making should be considered for adding adaptation logic as most of the environment is dynamic in nature. For an environment where not much is known, dynamic decision-making abilities will help prevent chaos. While most experts chose that eternal sensors be added to adaptation logic, some viewed that it really depended on the system and the type of sensor.

Adaptive actions are critical to any self-adapting and self-managing system. Experts showed total consensus on not limiting the adaptive actions if the cost of the system is not a deciding variable. With adaptive actions, complexity of a system increases in turn increasing the cost. Runtime decision-making ability of a system depends upon the environment the system is deployed in.

In adaptation approaches, Proactive Adaptation is considered to be better. Experts were of the view that if the accuracy of the prediction is high, predicting a change is better than waiting for one. If late detection of a change could result into chaos, the system should have continuous monitoring ability like the one in self-healing systems.

According to the experts, supporting disciplines like Artificial Intelligence, and Control Theory are interlinked with self-adapting systems to a great extent whereas disciplines like Network and Distributed Computing and Human Computer Interaction are somewhat interlinked with these systems.

Upon analysis of last section covering the future of SASs and SMSs in Pakistan, and the companies currently working on these systems, experts were of the opinion that there is good scope of SASs and SMSs in Pakistan and the current trend of robotics industry is the proof of that.

Currently companies like Nexgin RC, Avengate, Boston Dynamics and DARPA are working on self-adaptive systems.

VI. Findings

The key findings of this research are as follows:

1) System should fulfill all self-* properties to be considered a self-adaptive system.
2) Dynamic Decision-Making should be present at different levels within the system.
3) System should adapt itself with changes in Self and Context.
4) Proactive Adaptation should be considered instead of Reactive Adaptation.
5) Continuous Monitoring should be used instead of Adaptive Monitoring.
6) Human interaction should be minimized to a near zero.
7) New software processes are required to engineer these systems.
8) Supporting disciplines like Artificial Intelligence, Control Theory, Software Engineering and Distributed Computing should be studied and made use of in developing these systems.

VII. Conclusion

The field of SASs and SMSs proposes a new way to increase robustness of the systems. Existing literature in the field gives a good start for further work in the area but it is still its initial stages and far from being acceptable in today’s changing environment. Self-adaptive systems offer new breaks and challenges for computer and software engineers concerning the adaptation, implementation and management. New process need to be developed to fulfill the requirements of these systems that cannot be fulfilled by the processes of typical software systems.

This research has reviewed the basics of engineering SASs and SMSs, and outlined the main aspects of adaptation. Experts had been presented with different aspects connected to the engineering of self-adaptive software systems from the existing literature for their input. Analysis shows the best available aspects to be followed for engineering these systems. These aspects, such as self-* properties, adaptation approaches and types, and processes are highlighted in the findings.

This research has also identified supporting disciplines for the SASs and SMSs. Artificial intelligence, control theory, software engineering, and distributed computing are few such disciplines. Techniques and processes used to engineer the self-adaptive systems depend on the combination of these disciplines. Many other disciplines can be added to this list in future.

Pakistan shows big scope in the field of SASs and SMSs. Introducing self-adaptive and self-managing capabilities in robotic systems will help transition robotics industry into this field.

Companies like Nexgin RC, Avengate, Boston Dynamics, and DARPA are currently working on self-adaptive systems on a large scale.

Appendix

A. Questionnaire

1) Do you know about self-adaptive and self-managing software systems? Can you briefly tell us the difference between normal system and self-adaptive & self-managing software system?
2) Do you think "Drone" and "Simple Robot" are self-adaptive and self-managing systems? (Yes / No)
3) Do you think "Mars Rover" is self-adaptive and self-managing system? (Yes / No)
4) While adding adaptation logic, which option will you consider? Static Decision Making or Dynamic Decision Making, and does any other type of decision making exist?
5) Do we need external sensors or engine to add adaptation logic? Or can we do this at application programming level?
6) Do you want to limit your adaptive actions? Or you want to decide at the runtime? For example, when water comes in a way then you jump, or you will decide at runtime that you need to walk or jump?
7) Which monitoring process you will consider from the following:

   Continuous or Adaptive? Also, tell us the benefits of your chosen choice/answer? Please note that Continuous monitoring is more cost hungry than Adaptive.
8) System should respond when change has happened or system should predict change? What are your thoughts about this?
9) Do you think that human involvement or human interaction is not desirable in self-adaptive and self-managing system? If it's required, then to which extent?
10) Can you list down the supporting disciplines that are interlinked with self-adaptive and self-managing software systems? Are these disciplines behaving in a similar way like engineering of normal software system? One of the example is Artificial Intelligence and one is Software Engineering. Are these disciplines treating Self-Adaptive and Self-Managing software systems differently? For example, Requirement Engineering, Architecture Design and Testing/QA from software engineering are same for Self-Adaptive and Self-Managing software systems?
11) What are the trends in applying formal methods in self-adaptive systems?
12) Do you know any companies or groups who are working on development or R&D of these type of systems?
13) Name some of the companies or groups that are working on the development or R&D of these type of systems?
14) Future scope of these Self-Adaptive and Self-Managing software systems in Pakistan?
15) Do you think Robotic industry should go towards these systems? And should add Self-Adaptive and Self-Managing system's features in upcoming robots?

References

[1] R. de Lemos, H. Giese, H. A. Muller, and M. Shaw, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems*, Heidelberg, Germany, Springer-Verlag Berlin, 2013, pp. 1-32.

[2] P. Oreizy, M. M. Gorlick, R. N. Taylor, D. Heimbigner, G. Johnson, N. Medvidovic, A. Quilici, D. S. Rosenblum, and A. L. Wolf, "An architecture-based approach to self-adaptive software," *IEEE Intelligent Systems*, pp. 54-62, May-June 1999.

[3] R. Laddaga, "Self adaptive software: Technical report 98-12," DARPA Broad Agency Announcement (BAA), 1997.

[4] J. J. Mulcahy and S. Huang, "Autonomic software systems: Developing for self-managing legacy systems," *EEE International Conference on Software Maintenance and Evolution*, Victoria, British Columbia, Canada, 2014.

[5] Berns and S. Ghosh, "Dissecting self-* properties," in *Proc. Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, San Francisco, California, 2009.

[6] P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Corporation, 2001.

[7] H. A. Muller, "Engineering self-adaptive systems," *SEAMS*, 2008.

[8] S. Easterbrook and B. Nuseibeh, "Requirements engineering: A roadmap," in *Proc. Conference on the Future of Software Engineering*, 2000.

[9] F. Macias-Escriva, R. Haber, R. del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7267-7279, 2013.

[10] S. Sucipto and R. S. Wahono, "A systematic literature review of requirements engineering for self-adaptive systems," *Journal of Software Engineering*, vol. 1, no. 1, pp. 17-27, 2015.

[11] Z. Yang, Z. Li, Z. Jin, and Y. Chen, "A systematic literature review of requirements modeling and analysis for self-adaptive systems," *Requirements Engineering: Foundation for Software Quality*, vol. 8396, pp. 55-71, 2014.

[12] Y. Brun, G. D. M. Serungendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Muller, M. Pezze, and M. Shaw, "Engineering self-adaptive systems through feedback loops," *Self-Adaptive Systems*, pp. 48-70, 2009.

[13] S. Dobson, S. Denazis, A. Fernandez, D. Gaiti, *et al.*, "A survey of autonomic communications," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 2, pp. 223-259, 2006.

[14] J. Kephart and D. Chess, "The vision of autonomic computing," *IEEE Computer*, vol. 36, no. 1, pp. 41-50, 2003.

[15] D. Garlan, "Model-based adaptation of self-healing systems," in *ACM SIGSOFT Workshop on Self-Healing Systems*, 2002.

[16] D. Garlan, A. C. Cheng, B. Schmerl, and P. Steenkiste, "Rainbow: Architecture-based self-adaptation with reusable infrastructure," in *Proc. International Conference on Autonomic Computing*, New York, 2004.

[17] A. Garlan, R. Monroe, and D. Wile, "ACME: An architecture description interchange language," in *Proc. Conference of the Center for Advanced Studies on Collaborative Research*, 1997.
[18] B. Schmerl and D. Garlan, "Exploiting architectural design knowledge to support self-repairing systems," in *Proc. 14th International Conference on Software Engineering and Knowledge Engineering*, 2002.
[19] K. Angelopoulos, V. E. S. Souza, and J. Pimentel, "Requirements and architectural approaches to adaptive software systems: A comparative study," in *Proc. 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, San Francisco, USA, 2013.
[20] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 4, no. 2, 2009.
[21] P. K. McKinley, S. M. Sadjadi, E. P. Kasten, and B. H. C. Cheng, "Composing adaptive software," *Computer*, vol. 37, no. 7, pp. 56-64, 2004.

Shahbaz Rahim has earned his bachelor's degree in software engineering and master's degree in engineering management with focus on software management and other related fields of ICT. He is a certified professional and holds many certifications on leading edge technological solutions from IBM and Microsoft.

Muhammad Shahbaz Rahim has extensive technical and team lead experience. Muhammad Shahbaz Rahim has worked with many organizations within and outside Pakistan. Currently he is working as senior consultant in AMMROC, Abu Dhabi UAE.

Muhammad Shahbaz Rahim has worked on many ICT projects with focus on enterprise content management and business process automation to ensure paperless environment in major ICT organizations. He is registered member of Pakistan Engineering Council.

Ali Ahsan specializes in field of 'Management', 'Information Systems' and 'Software Engineering'. Essentially a core 'Software Engineer', Dr. Ahsan earned his masters and doctorate degree in the field of 'Engineering Management' with focus on 'Software Engineering / Information Systems' and 'Management'.

He has extensive managerial and technical experience. Over the years; he had the opportunity of enjoying core strategic positions both within industry and academia. He is a consultant to many well reputed multinational organizations and has completed major industrial projects.

Dr. Ahsan has research publication in various international conferences and journals. He is a member of IEEE (USA), SEI (USA) and IIIS (USA). He is a gold medalist and has represented many organizations at many national and international forums. Other than his consultancy work, he is currently the CEO of The WISSEN GROUP, Dean at Foundation University and Visiting Professor at Stratford University, USA.

Sara Siddique has earned her bachelors' degree in electrical engineering with specialization in computer engineering and her masters' degree in the field of project management.

She has been working in academia for last two years. She is currently working as a research associate in Faculty of Business and Technology, Foundation University Islamabad, Pakistan. Her research interests are in the areas of management, engineering management, and project management.

Ms. Siddique is a registered member of Pakistan Engineering Council.