

# Hybrid Adaptive Software Development Capability: An Empirical Study

Asif Qumer Gill

School of Software, University of Technology, Sydney, Australia

Email: asif.gill@uts.edu.au

**Abstract**—Software intensive organisations require the integration of agility, people, process, product and tool elements for establishing a hybrid adaptive software development capability. This paper presents the results of the empirical study that has been conducted to identify the important underlying characteristics of the hybrid adaptive software development capability elements. Based on this investigation, the most critical and the least critical characteristics of the hybrid adaptive software development capability elements have been identified. The findings of this empirical study have several implications, and can be recasted into making practical recommendations for establishing a situation-specific hybrid adaptive software development capability.

**Index Terms**—Agile Methods, Empirical Software Engineering, Information Systems

## I. INTRODUCTION

Traditional plan-based software development approaches work well if the project requirements are fixed; but are often considered heavy when the project requirements are being changed frequently [18, 32]. This issue led to the development of new agile software development methods [1]. Agile methods are being embraced by programming shops as providing a much needed release from the overheads typically perceived as being imposed by traditional plan-based methods [23]. Agile methods focus on delivering software early rather than on the unnecessary documentation and reporting of deliverables and this is often seen as a welcome shift of balance towards the most important factor in software project development [2].

Agile methods, originated in the context of small and medium projects, claim to offer many tangible benefits over traditional methods, e.g. improved time-to-market, productivity and quality software while at the same time aiming to reduce development cost [2, 6, 25]. Reifer et al. [24] suggested that “using agile methods to develop large systems presents a thorny set of issues”. Organizations can develop or tailor hybrid adaptive software development environment or capability [27] to deal with the challenge of agile at large scale. A hybrid adaptive software development capability can be established by integrating agility, people, process, product and tools elements [7, 11, 12, 19, 20, 33, 34] (e.g. integration of agile and non-agile elements). The aim of this paper is to obtain feedback from industry professionals as well as

researchers on these elements and underlying characteristics, which are important for establishing a hybrid adaptive software development capability. This paper presents the results of 46 structured interviews that have been conducted in order get feedback on these main elements: agility, people, process, product and tool. The findings of this empirical study have several implications, and would help organisations in making a decision about the importance of the elements’ characteristics (from more critical to less) when establishing their local hybrid adaptive software development capability.

This paper is organized as follows: Section II presents literature review. Section III presents the research methodology. Section IV presents the analysis and results of the empirical study interviews. Finally, Section V, discusses the validity and limitations of this study; before concluding in Section VI.

## II. LITERATURE REVIEW

Traditional plan-driven software development approach was an attempt to address the issues of code-and-fix approach by introducing repeatable process and practices. The plan-driven approach was advocated to manage large and life-critical projects (Boehm 1988). Some of the key issues, related to plan-driven approaches, that have been suggested and argued from time-to-time are: its emphasis on fully elaborated documentation, fixed contract, upfront detailed planning and design, top-bottom command and control process, slow response to changing requirements, focus on reporting deliverables rather than on actual delivery [5, 18]. These issues encouraged the software community to develop new lightweight and agile methods for software development [8].

As a reaction to so-called traditional plan-driven methods, a number of agile methods have been proposed over the last two decades and are being continually updated (e.g. XP, Feature Driven Development, Adaptive Software Development and Scrum). Agile software development is being embraced by IT shops as an alternative to traditional plan-based development [23]. Agile software development methods are argued to have several benefits over traditional plan-based methods - in particular, their ability to handle projects where the project requirements are not fixed [18, 28, 29, 30]. The Agile Manifesto [1] provides twelve agile principles and four agile values to qualitatively characterize the concept

of “Agility” in agile methods [10, 13, 16, 17, 22]. The ISO/IEC 24744 metamodel [9] describes the key elements of the traditional plan-driven methods in terms of “People”, “Process”, “Product” and “Tool”. Organizations can develop or tailor hybrid adaptive software development capability by combining the following agility [1, 19-22] and traditional plan-driven elements [9].

- Agility [1, 19-22]
- People [9]
- Process [9]
- Product [9]
- Tool [9]

However, the question is: What are the important underlying characteristics of these five elements? The aim of this paper is get feedback from practitioners on these elements and identify underlying characteristics, which are important for establishing a hybrid adaptive software development capability.

### III. RESEARCH METHOD

Based on the literature review (as summarized in Section II), a structured close-ended interview questionnaire was developed. This questionnaire contains nine questions related to five main elements and their underlying characteristics: (1) agility; (2) people, (3) process, (4) product; and (5) tool. The structured interview questionnaire was used to capture research participants’ response via face-to-face and Skype interviews. Participants from both research and industry organizations were invited via professional and research network and also through LinkedIn social media network. They included software consultants, coaches, managers, research scientists and developers. A total of 46 interviews were completed and included in this study. The interviewees had worked in a range of organizations from medium to large. Of the 46 interviewees, 24 were in industry and 22 were researchers.

All the interviews were analyzed as a single sample with no attempt (purposefully) made to discriminate responses on either industry size or characteristics of the respondents themselves (e.g. position in company, practitioner versus researcher) since it can be assumed that there is little impact from these factors on the overall knowledge of the project process and their (research participants’) personal experienced based opinions i.e. the opinions are of individuals and not of their organizations. Consequently, it is not claimed that this is a truly statistically representative sample since a fully representative sample is hard or perhaps impossible to obtain [15]. The main sampling challenge in this type of research is to identify and engage the relevant research participants. This sort of study is not easy to conduct and always interesting in terms of outcomes, especially if, as here, they are recast into practical recommendations. In other words, this research uses the notion of a convenience sample rather than a random sample since a

response was sought from a participants with a specific software development related experience and knowledge. Analysis of the data leads to a distribution of responses that we depict, for most questions, as percentages. When appropriate, we also analyze the tabulated percentages in terms of a ranking based on the perceived importance (from most to least important). This ordered ranking set is referred to as perceived “priority ranking”. It is argued that this “priority ranking” can be helpful in making recommendations to which elements’ characteristics are more critical and which, perhaps, may require more attention than of lesser importance. These priority rankings are included as an additional column in the tabulated results. In summary, this empirical study analyzes the data collected from 46 structured interviews and organizes the elements’ characteristics from most important to least important, based on the participants’ perceived importance or weight, called here perceived “priority ranking”.

### IV. ANALYSIS AND RESULTS

This section presents the detailed analysis of the agility, people, process, product and tool elements of a hybrid adaptive software development capability.

#### A. Agility

This section first presents the detailed analysis of the “agility” element characteristics [22] that are important for establishing a hybrid adaptive software development capability. It discusses the agility characteristics, degree of agility, agility adoption and improvement model. The first question is:

1. What are the key desirable characteristics of “agility” in your opinion?

Here, the responses of the participants are analyzed to find out the perceived important characteristics of ‘agility’. The five identified agility characteristics [22] are presented here together with their relevant importance (Table I), which is called here the agility priority ranking (see the italic and shaded column). The identified agility characteristics priority ranking, based on the responses of the participants of this research, shows that the most important characteristic of agility is “flexibility” (76%), and the least important is “leanness” (22%). A hybrid adaptive software development capability can pay more attention to the high priority ranking agility characteristics than the low ranking characteristics. These agility characteristic can be considered by project teams to calculate the degree of agility (see [19] on how to calculate degree of agility) of a hybrid adaptive software development capability. Agility priority rankings can be observed in Table I.

TABLE I.  
AGILITY CHARACTERISTICS

#	<i>Agility</i>	<i>Responses</i>	<i>%</i>	<i>Priority Ranking</i>
1	Flexibility	35	76	<i>Flexibility</i>
2	Speedy	23	50	<i>Learning</i>
3	Responsiveness	20	43	<i>Speedy</i>
4	Learning	25	54	<i>Responsiveness</i>
5	Leanness	10	22	<i>Leanness</i>

The degree of agility of a hybrid adaptive capability can be calculated, however, the question is:

2. How important is to calculate the degree of agility (quantitative value) of an existing or to-be developed hybrid adaptive software development capability?

In the earlier section, the five identified characteristics of agility were recognized together with their relevant importance – an agility priority ranking. An agility measurement model can be used (see [19] for details) to facilitate the calculation of the degree of agility of a hybrid adaptive capability, since the agility may be yet another element that may affect the cost estimation of a development or production of a software. For instance, a hybrid adaptive software development capability, with a certain (high or low) degree of agility, may produce software in less or high cost. Here, research participants’ responses are analyzed to assess the importance of such calculations. Only 7% of all the participants reported that knowing the degree of agility of a hybrid adaptive software development capability is not important, with 15% remaining neutral. A total of 78% (28% very important, 35% important and 15% minimal important) of all the participants reported that this is perceived as an important calculation (Table II).

TABLE II.  
DEGREE OF AGILITY

#	<i>Degree of Agility</i>	<i>Responses</i>	<i>%</i>
1	Very Important	13	28
2	Important	16	35
3	Minimal Important	7	15
4	Neutral	7	15
5	Not Important	3	7

The establishment and improvement of a hybrid adaptive software development capability can be assisted by an agility adoption and improvement model, however, the question is:

3. How important is the use of an agility adoption and improvement model in the context of hybrid adaptive software development capability establishment and improvement?

An agility adoption and improvement model [21] can be used as a roadmap to assess, adopt or improve the hybrid adaptive software development capability. The analysis results showed (Table III) that only 4% of all the participants reported that such model is not important, and 20% were unsure and remained neutral. However, in total 76% (26% very important, 48% important and 2% minimal important) of all the participants reported that such model would be important and helpful.

TABLE III.  
AGILITY ADOPTION AND IMPROVEMENT MODEL

#	<i>Degree of Agility</i>	<i>Responses</i>	<i>%</i>
1	Very Important	12	26
2	Important	22	48
3	Minimal Important	1	2
4	Neutral	9	20
5	Not Important	2	4

B. People

This section presents the detailed analysis of the “people” element of the hybrid adaptive software development capability. People is an important element of people-focused agile approaches, however the question is:

4. Which are the important characteristics of an individual (people) and a team that should be considered when establishing a hybrid adaptive software development capability?

Here, the responses from the participants have been obtained and analyzed to identify the important desired characteristics of an individual or a team in the context of a hybrid adaptive software development capability. Based on the analysis of the responses (Table IV), a characteristic priority ranking (italic and shaded column) has been identified, which shows that the communication & collaboration (78%), technical (76%) and problem solving & self-organizing (65%) are the top people characteristics for the hybrid adaptive software development capability. More interestingly, it can also be observed that the “education” characteristic (15%) is considered as being very low in the opinion of the participants. People alert characteristic should be considered and assessed for a specific situation in order to determine when an individual should or should not be appropriate to use. Only 2% of the participants suggested exploring other people related characteristics.

TABLE IV.  
PEOPLE CHARACTERISTICS

#	People	Responses	%	Priority Ranking
1	Education	7	15	Communication and Collaboration
2	Technical	35	76	Technical
3	Experience	26	57	Problem Solving and Self Organizing
4	Communication and Collaboration	36	78	Experience
5	Problem Solving and Self Organizing	30	65	Degree of Agility
6	Degree of Agility	15	33	Education
7	People Business Value	6	13	People Constraints and Risks
8	People Constraints and Risks	7	15	People Business Value
9	Alerts	1	2	Alerts
10	Others	1	2	Others

TABLE V.  
SOFTWARE PROCESS TYPES

#	Process Types	Responses	%	Priority Ranking
1	Process Engineering and Management	27	59	Process Engineering and Management
2	Product Engineering and Management	12	26	Team Engineering and Management
3	Team Engineering and Management	20	43	Knowledge Engineering and Management
4	Workspace Engineering and Management	10	22	Product Engineering and Management
5	Knowledge Engineering and Management	14	30	Workspace Engineering and Management
6	Governance	10	22	Governance
7	Others	2	4	Others

C. Process

This section presents the detailed analysis of the “process” element [9] of the hybrid adaptive software development capability. A hybrid adaptive software development capability can be realized by more than one types of software processes (product engineering, process engineering etc.); indeed, it is likely that all the software processes will not have the same importance. Therefore, the question is:

5. Which are the important software process types that you will include in establishing a hybrid adaptive software development capability?

The responses of the 46 participants have been analyzed (Table V) to identify the relative importance of different software process types (to be constructed or already constructed) for a hybrid adaptive software development capability. The process engineering and management (59%), team engineering and management (43%), knowledge engineering and management (30%) and product engineering and management (26%) processes perceived to be top processes for the hybrid adaptive software development capability. It can be observed from the analysis that the workspace engineering and management (22%), and governance (22%) processes have been considered to be less important in comparison with the rest of the processes. The priority ranking of the software processes will help and guide the process engineer to pay special attention to the top ranking processes when engineering processes for establishing the hybrid adaptive software development capability. Only 4% of the participants suggested to explore other software process types.

A hybrid adaptive software development capability can be realized by different types of software processes, however, the question is:

6. Which are the important characteristics of a software process?

The responses of the participants have been analyzed in the search for the important characteristics to describe a software process (Table VI). Here, the identified characteristics priority ranking may guide a software process engineer in consideration of which attributes of a process are important and need to be described and modeled in detail. This will also facilitate the reduction of the production waste and will allow focusing on more important process characteristics. The process purpose and objective (78%), pre and post conditions (39%), degree of agility (39%) and business value (39%) are perceived to be top characteristics of a software process in the context of a hybrid adaptive software development capability. It can be observed from the analysis that the process related tools and people (35%), constrains and risks (24%), abstractions mechanism (20%) and alerts (4%) characteristics have been considered to be less important (22%) in comparison with the rest of the other process characteristics. Process alert characteristic provides the information about the situation when a process should not be used. The priority ranking of the process characteristics will help and guide the process engineer to pay special attention to the top ranking process characteristics when describing processes of the hybrid adaptive software development capability. Only 4% of the participants suggested to explore other software process characteristics.

TABLE VI.  
SOFTWARE PROCESS CHARACTERISTICS

#	Process	Responses	%	Priority Ranking
1	Purpose and Objective	36	78	Purpose and Objective
2	Tools and People	16	35	Pre and Post Conditions
3	Pre and Post Conditions	18	39	Degree of Agility
4	Abstraction Mechanism	9	20	Business Value
5	Constraints and Risks	11	24	Tools and People
6	Degree of Agility	18	39	Constraints and Risks
7	Business Value	18	39	Abstraction Mechanism
8	Alerts	2	4	Alerts
9	Others	2	4	Others

#### D. Product

This section presents the detailed analysis of the “product” element [9] of a hybrid adaptive software development capability. A hybrid adaptive software development capability is established to produce software products or work products, however, the question is:

7. Which are the important characteristics that can be used to describe a software product?

The responses of the participants have been analyzed to identify the important characteristics that can describe and model a software product. Table VII shows the results of the analysis together with the identified priority ranking, which may be a useful guide for determining which product characteristics are important and need to be described or modeled in the context of a hybrid adaptive software development capability. The product complexity (72%), business value (54%), size (46%), and criticality (43%) are perceived to be top characteristics of a software product in the context of a hybrid adaptive software development capability. Interestingly, abstraction mechanism characteristic was ranked very low very low by the participations (only 7%), thought it refers to a key product modelling paradigm (e.g. object, service or agent oriented modelling). Only 2% of the participants suggested to explore other software product characteristics.

TABLE VII.  
SOFTWARE PRODUCT CHARACTERISTICS

#	Product	Responses	%	Priority Ranking
1	Size	21	46	Complexity
2	Complexity	33	72	Business Value
3	Criticality	20	43	Size
4	Development Duration	11	24	Criticality
5	Abstraction Mechanism	7	15	Constraints and Risks
6	Constraints and Risks	14	30	Development Duration
7	Business Value	25	54	Abstraction Mechanism
8	Alerts	2	4	Alerts
9	Effort	1	2	Effort
10	Others	1	2	Others

#### E. Tool

This section presents the detailed analysis of the “tool” element [9] of a hybrid adaptive software development capability. A hybrid adaptive software development capability applies a number of people, processes and tools to produce software products or work products, however, the question is:

8. Which types of tools are important for establishing a hybrid adaptive software development capability?

Agile processes are called people-focused processes, but we cannot eliminate the tools factor since tools help to ease and speed up the development when required and appropriate (e.g. automated testing tools). Here, the responses of the research participants have been analyzed to find out the important types of tools for establishing a hybrid adaptive software development capability or environment. Interestingly, in total, 85% (Table VIII) of the 46 participants supported the use of fully automated (35%) and semi-automated (52%) tools for establishing a hybrid adaptive software development capability or environment. However, it can be observed from the following table (Table VIII) that only 7% of the participants were agreeable to using a minimal-automated or tools-oriented environment. This shows that, however much agile methods are people-focused and give less importance to tools and processes, in order to make them workable and applicable in the industry, other important aspects such as tools and processes cannot be eliminated altogether. A balanced recipe of software development may be considered to be more practicable and workable. This analysis will help the practitioner to make their decisions about the setup of the tools to support hybrid adaptive software development capability.



TABLE VIII.  
TOOL TYPES

#	Product	Responses	%	Priority Ranking
1	Automated	16	35	Semi-automated
2	Semi-automated	24	52	Automated
3	Minimal automated	3	7	Minimal automated
4	Non-automated	3	7	Non-automated

A hybrid adaptive software development capability applies a number of different types of tools, however, the question is:

9. Which are the important characteristics that that can be used to describe the tool element of a hybrid adaptive software development capability?

The responses of the research participants have been analyzed in order to find out the relevant and important characteristics to describe a hybrid adaptive software development capability tool element (software development tools and environment). The analysis of the data (Table IX) shows the identified priority ranking among the characteristics of the “tool” element. The three most important characteristics of the tool element are identified: tool usability (65%), degree of agility (48%) and business value (30%). The priority ranking of the tool characteristics will help and guide the capability engineer to pay special attention to the top ranking tools characteristics when assessing and selecting tools for establishing the hybrid adaptive software development capability.

TABLE IX.  
TOOL CHARACTERISTICS

#	Product	Responses	%	Priority Ranking
1	Usability	30	65	Usability
2	Pre and Post Conditions	10	22	Degree of Agility
3	Constraints and Risks	12	26	Business Value
4	Degree of Agility	22	48	Constraints and Risks
5	Business Value	14	30	Pre and Post Conditions
6	Alerts	1	2	Alerts
7	Others	0	0	Others

V. DISCUSSION AND LIMITATIONS

This paper analyzed the agility [1, 19-22] and traditional plan-driven elements [9] of a hybrid software development capability from 46 practitioners’ perspectives. The agility element of a hybrid adaptive capability is described here in terms of agility characteristics, degree of agility, agility adoption and improvement model. The people element of a hybrid capability refers to the people that perform different roles (e.g. developer, iteration manager, tester, analyst roles) in different teams (e.g. development, testing, and deployment teams). The analysis presented in this paper

highlighted important and generic characteristics of people when involving them in different roles and different teams. The process element of a hybrid capability refers to the type of processes and their characteristics. A hybrid adaptive software development capability can be established by tailoring and integrating a number of different types of processes from different agile and non-agile methods.

The analysis presented in this paper highlighted important types of processes and their characteristics that are relevant in the context of a hybrid adaptive software development capability. The product element of a hybrid capability refers to the products that are produced through the application of people and process elements. The analysis presented in this paper highlighted the important and generic characteristics of software products in the context of a hybrid adaptive software development capability. A hybrid adaptive software development capability uses an integrated development environment containing different types of agile and non-agile tools to support people and process elements. The analysis presented in this paper highlighted the important type of tools and their characteristics that are relevant in the context of a hybrid adaptive software development capability.

There are some limitations, which we think quite reasonable and worth mentioning. A disadvantage of the use of the structured close-ended questionnaire, which may be considered here, is that the research participants were provided with a list of possible choices and asked to select from that list. This pre-determination may limit the focus of the research participants to the characteristics that were listed in research questionnaire. However, this issue was addressed by providing an extra option of “others” wherever possible and appropriate. This allowed the research participants to specify any additional characteristics that they might think should be considered in the context of this research. Another issue, which is worth considering, is that the responses of the participants are based on and limited to their personal opinion, experience, knowledge, beliefs and attitudes regarding the various aspects of software development methodologies. This situation may cause problems when participants’ perceptions are inaccurate or when characteristics identified as important for a hybrid adaptive software development capability may not in fact be important at all. However, similar to many other opinion-based research studies (for example [3, 4, 14, 15, 31]), we have full confidence that the findings of this research are based on the data that have been collected from the research participants, who have been involved and have vastly diversified experience in the research, engineering and implementation of software development projects.

The sample size of the research participants may be another concern since the data collection, due to the time constraint in this research, restricted further data collection so that only the original 46 participant responses could be collected and analyzed. To get a broader view on this research topic, and to make the

results of this research more general, more time, participants and organizations need to be involved.

## VI. CONCLUSION

A hybrid adaptive software development capability can be established for a particular organization through the integration of agility [1, 19-22], and traditional people, processes, product, and tool [9] elements. This paper presented the analysis and results of the responses from the 46 research participants that highlighted the important characteristics of agility, people, processes, product, and tool elements of a hybrid adaptive software development capability. These elements and underlying characteristics provide a baseline information for facilitating the establishment of a hybrid adaptive software development capability. The findings of this empirical study have several implications, and would help organizations in making a decision about the importance of the elements' characteristics (from more critical to less) when establishing their local hybrid adaptive software development capability. This research will be further extended based on the future learning, experience and research involving more participants to make the results of this research more general.

## ACKNOWLEDGMENT

The author wishes to thank the experts from industry and research organizations who helped with their valuable feedback and experience.

## REFERENCES

- [1] Agile Manifesto, Manifesto for Agile Software Development. <http://agilemanifesto.org/>.
- [2] L. Barnett, and C. Schwaber, "Adopting agile development processes: improve time to benefits for software projects, Forrester Research, March 2004.
- [3] N. Baddoo, and T. Hall, "Motivators of software process improvement: An analysis of practitioner's views", *Journal of Systems and Software*, vol. 62, pp. 85-96, May 2002.
- [4] S. Beecham, H. Tracy, and R. Austen, "Software process problems in twelve software companies: an empirical analysis", *Empirical Software Engineering*, vol. 8, pp. 7-42, March 2003.
- [5] B.W. Boehm, A Spiral Model of Software Development and Enhancement, *Computer*, IEEE JNL, (21:5), p. 61-72, 1988.
- [6] B. Fitzgerald, Hartnett, G., and K. Conboy, "Customising agile methods to software practices at Intel Shannon", *Journal of Information Systems*, vol. 15, pp. 200-213, April 2006.
- [7] B. Henderson-Sellers & A. Qumer, 'Using Method Engineering to Make a Traditional Environment Agile', *Cutter IT Journal*, 2007.
- [8] J. Highsmith & A. Cockburn, 'Agile Software Development: The Business of Innovation', IEEE Computer Society, *Computer*, vol. 34, no. 9, 2001.
- [9] ISO/IEC: *Software Engineering: Metamodel for Development Methodologies. ISO/IEC 24744*. International Standards Organization/International Electrotechnical Commission, Geneva, Switzerland; 2007.
- [10] S. Kuppuswami, K. Vivekanandan, P. Ramaswamy, and P. Rodrigues, "The effects of individual XP practices on software development effort", *ACM SIGSOFT Software Engineering Notes*, vol. 28, p. 6, November 2003.
- [11] A. Mahanti, "Challenges in enterprise adoption of agile methods – A Survey", *Journal of Computing and Information Technology - CIT*, vol. 14, pp. 197–206, 2006.
- [12] M. Maharmeh, B. Unhelkar, Investigation into the Creation and Application of a Composite Application Software Development Process Framework (CASDPF). *Fifth International Conference on Information Technology: New Generations, IEEE ITNG*, 2008.
- [13] J. McAvoy, and T. Butler, "A failure to learn in a software development team: the unsuccessful introduction of an agile method", in *Information Systems Development. Challenges in Practice, Theory, and Education* (eds. C. Barry, K. Conboy, M. Lang, G. Wojtkowski and W. Wojtkowski), Springer-Verlag, New York, USA, vol. 1, pp.1-13, 2009.
- [14] M. Niazi, and M.A. Babar, "De-motivators of software process improvement: An analysis of vietnamese practitioners' views". *PROFES*, Springer-Verlag, vol. 4589, pp. 118-131, August 2007.
- [15] M. Niazi, D. Wilson, and D. Zowghi, "Critical success factors for software process improvement: An empirical study". *Software Process Improvement and Practices Journal*, vol. 11, pp. 193-211, May 2006.
- [16] S. Nerur, R., Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies", *Communciations of the ACM*, vol. 48, pp 72-78, May 2005.
- [17] M. Pikkarainen, O. Sal, and J. Still, "Deploying agile practices in organisations: A case study", *EuroSPI*, Springer-Verlag, vol. 3792, pp. 16-27, October 2005.
- [18] F. Paetsch, A. Eberlein, & F. Maurer, 'Requirements Engineering and Agile Software Development', *Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linz, Austria, pp. 308 -313, 2003.
- [19] A. Qumer, and B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering", *Information and Software Technology (IST)*, vol. 50, no. 4, pp. 280-295, 2008.
- [20] A. Qumer & B. Henderson-Sellers, 'Construction of an Agile Software Product-Enhancement Process by Using an Agile Software Solution Framework (ASSF) and Situational Method Engineering', *International IEEE Computer Software and Applications Conference (IEEE COMPSAC2007)*, Beijing, China, 2007.
- [21] A. Qumer, B. Henderson-Sellers & T. McBride, 'Agile adoption and improvement model', *European and Mediterranean Conference on Information systems (EMCIS2007)*, Valencia, Spain, 2007.
- [22] A. Qumer & B. Henderson-Sellers, 'Crystallisation of Agility: Back to Basics', *International Conference on Software and Data Technologies*, Setubal, Portugal, pp.121-126, 2006.
- [23] D. Reifer, "How good are agile methods?", *IEEE Software*, vol. 19, pp. 16-18, July 2002.
- [24] Reifer, D. 2003 'XP and the CMM', *IEEE Software*, vol. 20. No. 3, pp. 14-15.
- [25] Salesforce. Agile development meets cloud computing for extraordinary results at salesforce.com, 2008.
- [26] B. Schatz, and I. Abdelshafi, "Primavera gets agile: a successful transition to agile development", *IEEE Software*, vol. 22, pp. 36-42, May/ June 2005.
- [27] A. Sidky, "A structured approach to adopting agile practices: The agile adoption framework", Ph.D. Thesis,

- Virginia Polytechnic Institute and State University, USA., 2007.
- [28] J.L. Taylor, 'Lightweight Processes for Changing Environments', *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 27, no. 11, pp. 82, 2002.
- [29] L. Williams & A. Cockburn, *Agile Software Development: It's about Feedback and Change*, IEEE Computer Society, *Computer*, vol. 36 (6), 2003.
- [30] L. Williams, R.R. Kessler, W. Cunningham, and R. Jeffries, "Strengthening the case for pair programming", *IEEE Software*, vol. 17, pp. 19-25, 2000
- [31] U. K., Durrani, J. Richardson, and J. Lenanrcic, "Investigation on Australian Agile Software Development Organizations: An Exploratory Study of Adaptable SCM Process Implementation. *JCP*, vol. 8(7), pp.1799-1803, 2013.
- [32] C. Yuan, W. Wang, Y. L. Lin, and Y. Chen, "Design Method of Product Agile Customization Based on Artificial Neural Network and Its Application". *JCP*, vol. 6, pp.776-783, 2011.
- [33] H. Wang, and Z. Zheng, "Collective Self-adaptive Software Architecture Specification: Understanding Uncertainty in Cyber-Physical Convergence". *JCP*, vol 9(4) pp.802-811, 2014.
- [34] A.Q. Gill, "Towards the Development of an Adaptive Enterprise Service System Model". Americas Conference on Information Systems, Chicago, USA, 2013.

**Dr. Asif Q. Gill** is a TOGAF 9 Certified Enterprise Architect, Lecturer and Researcher at the School of Software at the University of Technology, Sydney. He specializes in software-intensive agile enterprise architecture and engineering practices, tools and techniques. He is an experienced author, coach, consultant, educator, researcher, speaker, trainer and thought leader. He has extensive experience in both agile, non-agile, cloud and non-cloud complex software development environments, displaying a deep appreciation of their different perspectives in a number of IT-enabled business process improvement and transformation industry projects of varying sizes. He can be reached at [asif.gill@uts.edu.au](mailto:asif.gill@uts.edu.au).