

Scaling for agility: A reference model for hybrid traditional-agile software development methodologies

Asif Qumer Gill¹ · Brian Henderson-Sellers¹ · Mahmood Niazi^{2,3}

Published online: 16 July 2016
© Springer Science+Business Media New York 2016

Abstract The adoption of agility at a large scale often requires the integration of agile and non-agile development elements for architecting a hybrid adaptive methodology. The challenge is “which elements or components (agile or non-agile) are relevant to develop the context-aware hybrid adaptive methodology reference architecture?” This paper addresses this important challenge and develops a hybrid adaptive methodology reference architecture model using a qualitative constructive empirical research approach. In this way, we have uncovered the agility, abstraction, business value, business policy, rules, legal, context and facility elements or components that have not been explicitly modelled or discussed in International Standards (IS) such as the ISO/IEC 24744 metamodel. It is anticipated that a context-aware hybrid adaptive methodology can be architected by using the proposed context-aware hybrid adaptive methodology reference architecture elements for a particular situation when using a situational method engineering approach.

Keywords Agile methods · Reference architecture · Methodology architecture · Method engineering

1 Introduction

Large projects are difficult to handle and often fail to meet stakeholders’ expectations (Janssen et al. 2015). While agile

methods have engendered significant interest in industry, there is little investigative research regarding their architecture and their mode of adoption within a large scale environment involving outsourcing, multiple programs, projects and methodologies - currently seen as a major challenge in the industrial context (Rodríguez et al. 2012; Lee and Young 2013; Asan, and Bilgen 2013). It is widely agreed that there can be no single methodology that can be universally applied to all projects; thus all agile and non-agile methodologies need to be tailored and integrated to support multiple projects (Mahanti 2006). This draws attention to the fact that a software development capability may combine agile and traditional elements to create a hybrid software development method in order to tackle these challenges for using agile approaches within large projects (Boehm and Turner 2003; Gill 2014). Hence, the adoption of agility at a large scale may require the integration of agile and non-agile elements for architecting context-aware hybrid adaptive methodologies (Sommer et al. 2014). We argue here that this can be best undertaken with the assistance of a context-aware hybrid adaptive methodology reference architecture (abbreviated here as ‘HAMRA’) model.

Architecture is defined as the “fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution” (ISO/IEC 42010 2011). Here, we apply systems thinking (Miller 1995) and use an architecture-driven approach to a software or information systems development (ISD) methodology – methodology as a living system. A reference architecture is a blueprint or template for developing a concrete architecture (Harrison 2011). In this context, a methodology reference architecture is a template that describes the methodological system elements (components), concepts or properties (classes) and their relationships to each other and the environment at an abstract or high level, allowing for drilling down for each of these elements as needed for the creation

✉ Asif Qumer Gill
asif.gill@uts.edu.au

¹ University of Technology Sydney, Ultimo, NSW 2007, Australia

² King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia

³ Riphah International University, Islamabad, Pakistan

of a context specific project methodology. Thus, the challenge is:

“Which elements or components (agile or non-agile) are relevant for developing a context-aware hybrid adaptive methodology reference architecture?”

This research addresses this important question and identifies the context-aware hybrid adaptive methodology elements for developing the proposed HAMRA by using the qualitative constructive empirical research approach (Jarvinen 2001). The proposed HAMRA is intended to allow project teams to architect the context-specific configurations of hybrid adaptive methodologies by using the well-known situational method engineering (SME) and tailoring techniques (e.g. Kumar and Welke 1992; Brinkkemper et al. 1998; Henderson-Sellers and Ralyté 2010; Henderson-Sellers et al., 2014). Finally, HAMRA can also act as a first step towards the further investigation and development of a standard metamodel for context-aware hybrid adaptive methodologies (a research gap in agile development) – providing an opportunity to extend an ISO standard such as ISO/IEC 24744 (ISO/IEC, 2014) to explicitly support agility for large scale development.

This paper is organized as follows: Section 2 defines the research background and problem. Section 3 presents the research methodology. Section 4 presents the HAMRA based on the literature and on the results from two industry empirical studies. Section 5 describes how the improved and final version of the HAMRA has been applied in architecting the configuration of a context-specific hybrid adaptive methodology in two teaching cases. In Section 6 we discuss the overall research contribution and limitations before concluding in Section 7.

2 Research background and problem

Plan-driven or non-agile software development approaches (Boehm 1988) can be described as predictive, process-focused, document-driven and plan-driven whereas an agile approach (Agile Manifesto 2001; Fowler 2003) is seen as primarily adaptive and people-oriented. It has been suggested (Qumer and Henderson-Sellers 2008a) that traditional plan-driven software development practices (e.g. waterfall, spiral) may be more applicable in large projects, whereas agile practices are (claimed to be) applicable in small to medium size projects. Both agile and traditional plan-driven approaches have their own individual business values and benefits. However, agile methods and their practices offer many potential tangible benefits to an organization over traditional plan-driven approaches, e.g. improved time-to-market, productivity and quality software while reducing development cost and documentation (Reifer 2002). While many organizations are interested in adopting these agile methods or practices suitable

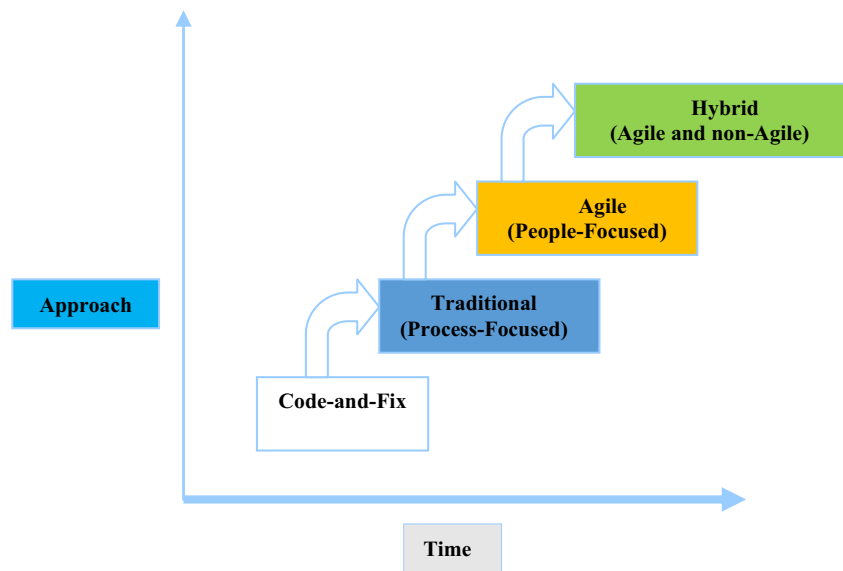
to their local circumstances, there are still concerns regarding the adoption of agile in large-scale distributed project development environments (Vijayasarathy and Turk 2008; Laanti 2013).

Agile adoption does not simply depend on available universal agile methods. The core of agility in ISD is focused on the “cognition” or “mindset” of the people (Agile Manifesto 2001; Chance 2011) who are actually involved in the project. The success of agile adoption requires a gradual shift from a traditional to an agile mindset. Indeed, some organizations may find it more beneficial to initially move from a traditional non-agile approach to a hybrid approach and then from a hybrid to a fully agile mindset through the increasing adoption of all the agile principles. The agile mindset emerges through the ongoing tailoring and adoption of a context-specific agile methodology architecture underpinning agile values, principles and practices.

As noted above, the ongoing tailoring and adoption of an agile method may or may not be straightforward (Laanti 2008). However, “a gradual transition from a heavyweight or plan-driven (e.g. waterfall) to an agile process can make the change easier on the development team” (Cohn and Ford 2003). It is also possible to have an agile or hybrid software development method (Maharmeh and Unhelkar 2008) that can be built and/or tailored by mixing agile and traditional (non-agile) elements. It has been suggested (Gains and Hawkins 2007) that “Moving a traditional waterfall organisation into the world of agile methods is one thing, but when that organisation has many customers that operate with plan based on milestones and fixed delivery dates, some hybrid method is called for”. The development of the architecture of an agile or hybrid method is not a one-time up-front activity; rather, it is an ongoing activity that involves actual onshore and offshore (i.e. distributed) project team members. These team members continuously tailor and evolve their hybrid methodology architectures through the gradual adoption of situation-specific agile values, principles, practices and tools suitable to their context. The ongoing tailoring and evolution of context-specific hybrid adaptive method (see Fig. 1) architectures is a key motivation for our research.

Here, it may be suggested that agile and non-agile elements are two extremes and have their own individual advantages over each other (Qumer and Henderson-Sellers 2008a). Rather, it may be appropriate to architect a hybrid adaptive methodology (e.g. a combined set of agile and non-agile elements) for a specific situation. The architecture of the hybrid methodology for a large-scale development can be engendered by using the proposed context-aware HAMRA reference architecture and the situational method engineering approach (Henderson-Sellers et al. 2014). The research presented in this paper is aimed at developing and evaluating the context-aware HAMRA reference architecture for architecting hybrid adaptive methodologies.

Fig. 1 Emergence of hybrid approach



3 Research method

This research has been conducted by the application of an iterative qualitative constructive empirical research approach (adapted from Kasanen et al. 1993; Jarvinen 200; Hevner et al. 2004; Peffers et al. 2006), which is appropriate to the development and evaluation of the proposed HAMRA. This research approach is a way of linking research and practice (March and Smith 1995; Peffers et al. 2006). It is an iterative constructive research process (Jarvinen 2001; Osterle et al. 2010) that permits the development and evaluation of novel artefacts in short iterations. Here, we describe the six research process steps that have been used in our research spanning well over seven years (2006–2013) for the iterative development and evaluation of the HAMRA.

- **Step 1 (Iteration 0: Initiation)** - Firstly, based on the initial literature review, we identified the research problem, objective and motivation for the development of the HAMRA (as discussed in Section 2).
- **Step 2 (Iteration 0: Initiation)** - Secondly, we identified the qualitative constructive empirical research approach most appropriate for developing and evaluating the proposed HAMRA (this section).
- **Step 3 (Iteration 1: Development)** - Thirdly, based on the review of the literature and related work (see Section 4), the initial elements for the HAMRA have been identified.
- **Step 4 (Iteration 1: Evaluation)** - Fourthly, the initial HAMRA elements have been empirically analysed by the means of two empirical case studies (case ‘A’ and case ‘B’) in industry (see Section 4). The initial results of two empirical case studies have been described in (Qumer and Henderson-Sellers 2008b). These empirical studies are extremely important to get the practical guidance and inputs

from the industry to further develop and refine the proposed HAMRA elements. These cases have been very useful to validate the theory and theorize the practice in terms of HAMRA elements.

- **Step 5 (Iteration 2: Development)** - These empirical study results have been further analysed in order to identify the hybrid adaptive methodology element classes, relationships and additional element(s), if any, based on the empirical evidence. Based on theory (step 4) and the empirical study analysis (step 4), the HAMRA has been refined, updated and reported in this paper (see Section 4). The proposed HARMA is a unique blend of both theory and practice. It has both academic rigour and practical industry applicability and relevance. It provides concrete elements and underlying detail to support the large-scale hybrid agile development, which is perceived as challenging.
- **Step 6 (Iteration 2: Evaluation)** - The application of the refined and final version of the HAMRA is further demonstrated with the help of two teaching case studies (‘C’ and ‘D’) in academic settings (see Section 5). This concludes our research project.

The theory-based initial construct of the HAMRA includes both core and extended hybrid adaptive methodology elements. The empirical study was done by conducting hybrid and agile method tailoring workshops in industry (see Appendices A and B) by using the theory-based HAMRA elements in two different case organizations ‘A’ and ‘B’ (coded names). Empirical study is an inquiry that “investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomena and context are not clearly evident” (Yin 2003; Ojala 2003). It can be used to develop, test or describe a theory or phenomenon. The

practical applicability of the initial HAMRA elements have been tested by means of two empirical study case studies in industry. The initial results of these two empirical study cases have been published in detail in (Qumer and Henderson-Sellers 2008b).

This paper builds on these empirical studies, and further analyses the empirical evidence and lessons learned from both industry cases in order to further understand and identify each of the methodology element classes (concepts or properties) and their relationships by using qualitative grounded theory analysis techniques (Glaser 1998), which were found to be very helpful for systematically inspecting the empirical evidence and further identifying the elemental classes and their relationships. The theory-based hybrid methodology elements, empirical study-based relevant element classes and their relationships form the final version of the theory and practice-based HAMRA. The final version of the HAMRA elements, classes and their relationships has been further evaluated by means of two teaching cases ('C' and 'D') in academic settings at the University of Technology, Sydney (UTS).

In the first empirical industry case study, a hybrid methodology had been designed in a large organization 'A' (code name) by using the HAMRA elements (see Appendix A – Hybrid Agile Product Enhancement Process). Organization 'A' provides largely object-oriented paradigm-based product development, support and maintenance services to their customers while following a traditional software development approach. They decided that they had to seek a product enhancement process that would enable them to quickly identify and develop new product features to keep them competitive in the market. Their interest was piqued by the attributes of agile methods. However, organization 'A' was not ready to commit fully to the use of an agile software development method on a large scale, because they realized that a sudden change in the organizational culture and mind-set might be risky and problematical. Therefore, they decided to tailor and adopt a partially agile approach, only utilizing new features of the agility 'toolbox' in the identification process area, together with a partially traditional approach (for the detailed implementation process area) in order to keep things under control whilst retaining flexibility.

In the second industry case study, 'B', a full-scale service-oriented agile methodology (different from organization 'A') had been designed for a medium-size project-based organization by using selected HAMRA elements (see Appendix B – Agile Service Oriented Process). Their focus was on e-health application development using a service-oriented abstraction paradigm. The description of these case studies was reported in Qumer and Henderson-Sellers (2008b).

In the third teaching case study 'C', undertaken at UTS, a hybrid adaptive business analysis methodology was designed and used for teaching the project-based business requirements

modelling (BRM) subject at UTS. UTS offers this project-driven undergraduate (approx. 100+ students in autumn semester, and 250+ students in spring semester) subject over a period of 14 weeks through face-to-face lectures (1 h) and tutorial sessions (2 h). The lectures introduce students to the business analysis methodology. Students play the role of a business analyst and apply the business analysis methodology to a project case study, working in teams of 2–3. A hybrid adaptive business analysis methodology was designed by the first author by using the HAMRA elements. Students applied the hybrid adaptive business analysis methodology to their projects in small teams of 2–3 students during 2013–14 (four semesters). Teaching case study 'C' is discussed in Section 5.

In the fourth teaching case study 'D', also at UTS, a hybrid adaptive project development methodology was designed by the first author and used for teaching the project-based software engineering practice (SEP) subject at UTS. UTS offers this project-driven undergraduate (approx. 200+ students in spring semester) subject over a period of 14 weeks through face-to-face lectures (1.5 h) and tutorial sessions (1.5 h). The lectures introduce students to agile and non-agile development methodologies. Students play different roles (e.g. business analyst, developer, and tester) and apply the hybrid adaptive project development methodology to projects in a team of 5–7 students. Students applied the hybrid adaptive project development methodology (mixing agile and non-agile elements) to their projects in small teams during 2013–14 (two semesters). Teaching case study 'D' is discussed in Section 5.

So far, we have discussed steps 1 and 2 of our research in sections 2 and 3, respectively. The next section discusses step 4 of our research, which describes the proposed "HAMRA" based on theory (literature) and practice (empirical work).

4 The HAMRA

As a result of our theoretical literature analysis and our two industry empirical cases in industry, we developed the final and improved version of the HAMRA (Figs. 2, 3, and 4). The initial elements for the proposed HAMRA were identified based on a review of the traditional ISO/IEC 24744 SEMDM (Software Engineering Metamodel for Development Methodologies) standard metamodel, agility and related concepts (Table 1).

The literature review helped us to identify a set of 11 hybrid adaptive methodology elements. The 12th element methodology "Facility" was identified during the analysis of our two industry empirical cases. All 12 elements form the final version of the HAMRA. The HAMRA construct can be viewed from three viewpoints: an element viewpoint (Fig. 2), a classification viewpoint (Fig. 3) and a multidimensional viewpoint (Fig. 4). A viewpoint is a template that can be used to create context-specific views (Harrison 2011). The HAMRA

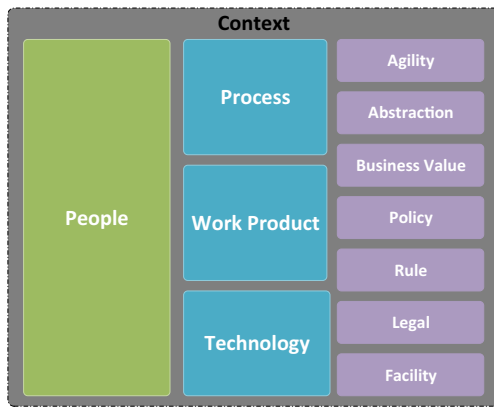


Fig. 2 Element viewpoint

element viewpoint (Fig. 2) is a template that provides the building blocks or elements of a hybrid adaptive methodology, which is a common practice in representing reference architectures (Harrison 2011). The HAMRA elements are classified into core and extended elements. The classification of the HAMRA elements is shown in the HAMRA classification viewpoint (Fig. 3). A hybrid adaptive methodology can be configured through the integration of HAMRA elements for a specific context. Thus, the relationships between the HAMRA elements are shown using the HAMRA multidimensional viewpoint (Fig. 4). Each element is represented as a dimension in the multidimensional viewpoint, which can be used to configure a hybrid adaptive methodology (fact).

Agility is about making complex things simple or, at least, simpler. Unlike a traditional complex relational metamodeling approach, here we used a simple and adaptable multidimensional modelling approach from the data warehousing literature (Kimball and Ross 2002) to show the relationships between different dimensions for a hybrid methodology (fact). This allowed the adaptability or flexibility to delete, modify or add new classes in each element dimension of a methodology as required and capture links between all dimensions in the hybrid adaptive methodology, which is called here a fact element. We can create many facts or hybrid methodologies through the integration of different methodology dimensions. The multidimensional view harnesses the autonomy of individual elements or components allowing

for drilling down for each of these element classes (e.g. classes and hierarchies) as needed for a specific project or context. The multidimensional approach can be further used to develop a repository or warehouse of hybrid adaptive methodologies, which can be further used to support methodology-related analytics.

Here, we described the final updated version of the HAMRA in terms of these three viewpoints. Organizations may create additional viewpoints suitable to their local context. The key sources and empirical analysis of the HAMRA elements are summarized in Table 1. The last column of Table 1 presents the consolidated and final version of the HAMRA elements.

4.1 The HAMRA - Core

The tailoring of hybrid methods for a specific context can be assisted by using a situational method engineering approach and a standard metamodel. For example, the Software Engineering - Metamodel for Development Methodologies (SEMDM) metamodel (ISO/IEC 2007) can be used to describe and specify methodology elements and their relationships (Ralyté 1999; Henderson-Sellers 2003). The SEMDM (ISO/IEC 2007) is a comprehensive and established standard that combines different traditional non-agile metamodels and software development methodologies. The review of ISO/IEC 24744 SEMDM resulted in the identification of four core well-established methodology elements: people, process, work product and tool (see Table 1). People use tools and perform processes to produce work products (Fig. 3). These elements are classified as core well-known elements. We discuss these four elements in detail based in both the literature and empirical study analyses in this section.

4.1.1 People

The adaptive hybrid methodology (Fig. 4) includes the people element (ISO/IEC 2007), which is one of the important constituents of people-focused (Agile Manifesto 2001) agile software development methodologies and approaches. People use agile or non-agile processes and tools for producing agile or

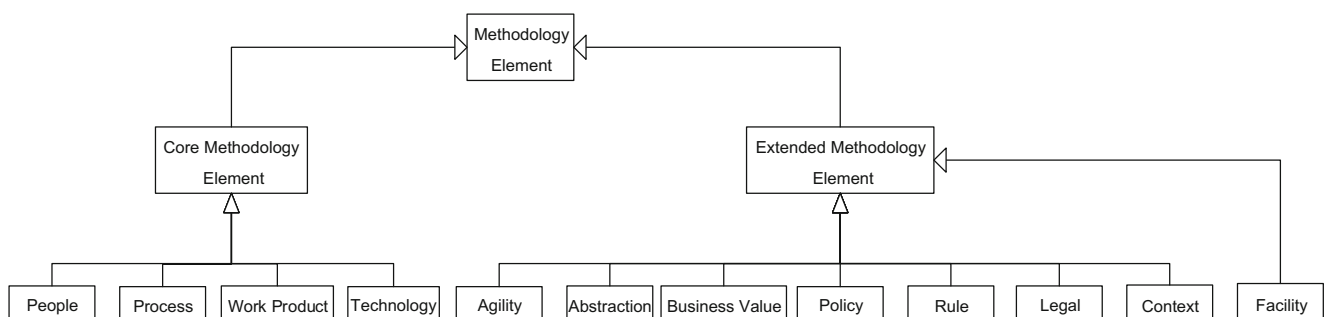


Fig. 3 Classification viewpoint

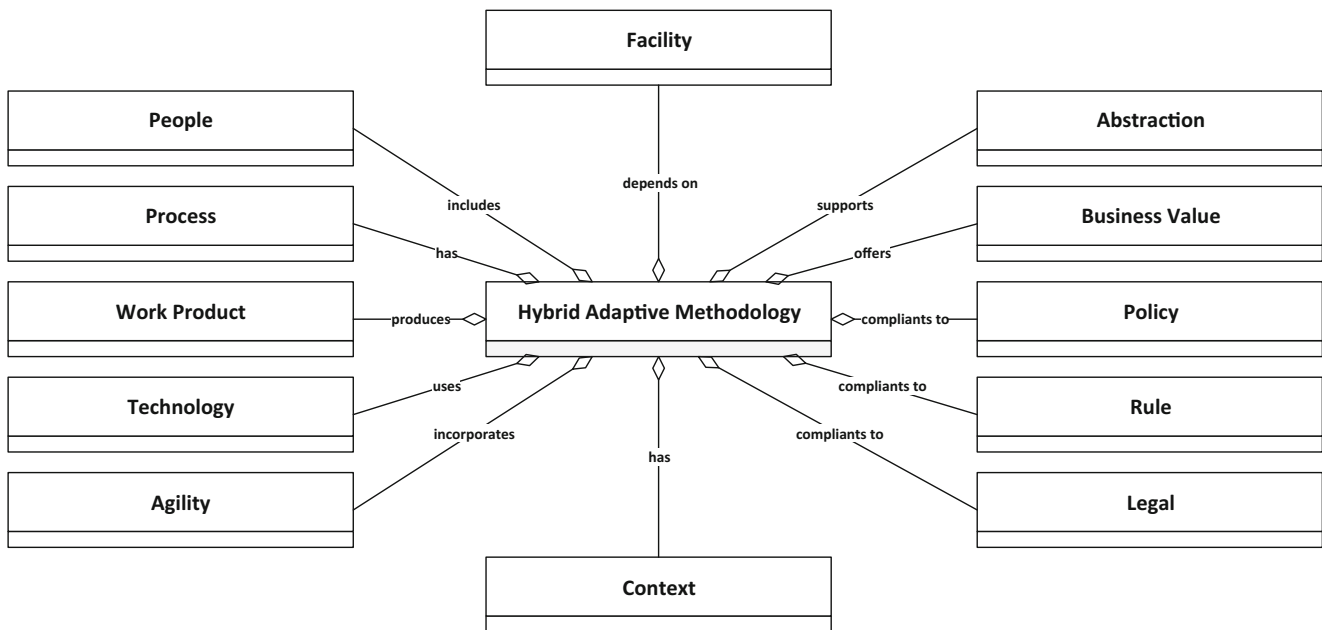


Fig. 4 Multidimensional viewpoint

non-agile work products. They may also have an agile competency (e.g. *agile* skills, education and experience) and capacity (e.g. availability and workload bandwidth). Conboy and Fitzgerald (2010) discuss the people or developers' characteristics such as "familiarity with a range of methods and methods" in the context of disciplined, purposeful and context-specific agile method tailoring and adoption.

The people element was used in defining the hybrid and agile methodologies for empirical case study organizations 'A' (see Appendix A) and 'B' (see Appendix B). For case study 'A', in the context of the people element, the agile team choreography model (describing a self-organizing, empowered and co-located team) had also been outlined in order to direct the behaviour of the agile team. The empirical study analysis highlighted that the case study organizations have different teams containing individuals who, regarding them as actors (e.g. producer, consumer), can be said to have competency and capacity to perform different roles (e.g. senior developer, product manager). Hence, the empirical study analysis helped us to identify seven sub-elements or classes and their relationships within the people element: individual, team, organization, actor, role, competency, capacity. These people element classes have been included in the updated final version of the HAMRA (Table 1).

4.1.2 Process

The adaptive hybrid methodology (Fig. 4) has a process element (as represented in the metamodel of ISO/IEC 2007) – indeed, any methodology has one or many integrated processes. The empirical study analysis of the tailored processes for the case study organizations helped us to elicit six process element

classes and their relationships: event, stage (or phase), activity, task, gateway (decision point) and technique (agile and non-agile practices). Therefore, these six classes have been included in the updated HAMRA under the process element (Table 1). For instance, in case 'A', the phases (e.g. post-mortem) and events (e.g. end of iteration) are mentioned such as "the post-mortem phase is executed after the end of each iteration, phase and process for the purpose of future learning and decision making". Similarly, both cases highlighted activities, tasks, decision points or gateways and techniques within the overall context of the process element. It has been found that the technique concept can cover concepts in both agile and non-agile practices. The analysis of the empirical study indicates that a process can be triggered by an event (e.g. request for new feature or service requirement) and may have one or many gateways (key process activity flow control and decision points), activities, tasks and techniques (practices), which can be organized into different stages (phases). Organizations may include additional process element classes, if required.

Process elements can be associated to other HAMRA elements; for example, process and agility elements together can describe an agile process. A process may be required in order for it to be compliant with rules, policy, legal and more abstract elements such as object-oriented or service-oriented processes. Similarly, other context-specific relationship types and relationships can be derived from the process element perspective.

4.1.3 Work product

The adaptive hybrid methodology (Fig. 4) produces work products (ISO/IEC 2007). The case study 'A' highlighted the

Table 1 The HAMRA elements and classes – theoretical and empirical analysis results

#	Classification	Theory	Elements/ Dimensions	Industry Case Study ‘A’	Industry Case Study ‘B’	HAMRA Elements
1	Core	ISO/IEC 24744 SEMDM	People	Individual Team Organization ActorRole Competency Capacity	Individual Team Organization ActorRole Competency Capacity	People: Individual Team Organization ActorRole Competency Capacity
			Process	Hybrid Agile Product Enhancement Process: Event Gateway Stage Activity Task Technique (agile and non-agile practices)	Agile Service Oriented Process: Event Gateway Stage Activity Task Technique (agile practices)	Process: Event Gateway Stage Activity Task Technique
			Work Product	Backlog Document Model Component Software Application	Backlog Document Model Matrix Component Software Application Interface Service Function	Work Product: Backlog Document Model Matrix Component Software Application Interface Service Function
			Tool	Platform Infrastructure	Platform Infrastructure	Technology: Platform Infrastructure
2	Agile (Extended)	Randell and Zurcher (1968)	Agility	Agile Value	Agile Value	Agility:
3		Stapleton (1997)		Agile Principle	Agile Principle	Agile Value
4		Wong and Whitman (1999)		Agile Level	Agile Level	Agile Principle
5		Highsmith (2000)				Agile Level
6		Agile Manifesto (2001)				
7		Palmer and Felsing (2002)				
8		Boehm and Turner (2004)				
9		Conboy and Fitzgerald (2004)				
10		Beck and Andres (2004)				
11		Henderson-Sellers and Serour (2005)				
12		Cockburn (2006)				
13		Schwaber (2007)				
14		Conboy (2009)				
15	Related Concepts	Berzins et al. (1986)	Abstraction	Object Oriented	Service Oriented	Abstraction:
16	(Extended)	Theodorakis et al. (1999)				Object Oriented
17		Odell (2002)				Service Oriented
18		Feuerlicht (2006)				
19		Royce (1970)	Business Value	Time	Time	Business Value:
20		Chapman (1997)		Scope	Cost	Time
21		Tyndale et al. (2002)			Quality	Scope
22		Boehm and Turner (2003)			Training	Cost
23		Fitzgerald et al. (2006)				Quality
24		Elssamadisy (2007)				

Table 1 (continued)

#	Classification	Theory	Elements/ Dimensions	Industry Case Study 'A'	Industry Case Study 'B'	HAMRA Elements
25		Smith and Sidky (2009)				Training
26		Racheva et al. (2009)				
27		Feiler and Humphrey (1993)	Policy	Product Development Policy	Project Development Policy	Policy: Product Development Policy
28		Kenens et al. (1998)				
29		Wallin et al. (2002)				
30		Reis et al. (2002)				Project Development Policy
31		Vahaniitty and Rautiainen (2008)				
32		Nasir et al. (2008)				
33		Lee and Young (2013)				
34		Brinkkemper (1996)	Rule	Communication Cooperation Decision	Communication Cooperation Decision	Rule: Communication Cooperation Decision
35		Leite (1998)				
36		Knolmayer et al. (2000)				
37		Cockburn (2002)				
38		Cockburn (2003)				
39		Cockburn (2005)				
40		Orriens et al. (2006)				
41		Commonwealth FOI ACT (1982)	Legal	Contract	Contract	Legal: Contract
42		Leonardi and Leite (2002)				
43		Beznosov and Kruchten (2004)				
44		Harmsen et al. 1995	Context	Nature Industry Domain Structure Culture	Nature Industry Domain Structure Culture	Context: Nature Industry Domain Structure Culture
45		Rolland and Prakash 1996				
46		Harmsen 1997				
47		Bucher 2006				
48		Tolfo and Wazlawick (2008)				
49		Harrison 2011				
50		Komyshova et al. 2011	Facility	Spatial	Spatial	Facility: Spatial

backlog (e.g. feature backlog), document (e.g. requirements specification document), model (e.g. architecture), component (e.g. product component) and software application (e.g. software product) classes of the work product element. In addition to these five classes, case study 'B' also highlighted the matrix (e.g. business and service mapping matrix), service (e.g. e-health consultancy services), function (e.g. business operation of a service) and interface (e.g. service interface) classes of the work product element. These nine classes have all been included in the final version of the HAMRA under the work product element (Table 1).

The work product element can be associated to other HAMRA elements such as work product and agility elements that can describe the agile work product. The work product could be executable or non-executable (e.g. test cases, product backlog, iteration and executable artifacts) that can be used or produced to add business value to a project or a product development environment. Work products may be based on various abstraction mechanisms (e.g. object-oriented and service-

oriented) and are compliant to rules, policy and legal requirements. Similarly, other context-specific relationship types and relationships can be derived from the work product element perspective.

4.1.4 Technology

The original tool element given in ISO/IEC (2007) has been renamed to technology element in HAMRA subsequent to the empirical analysis. This is because it is more than a tool or set of tools. It is about complex technology platforms and infrastructure that may be required to support real world, large-scale software development work. The empirical analysis highlighted the platform (e.g. J2EE development platform) and infrastructure (e.g. Web logic application server) classes of the technology element. These two classes have been included in the final version of HAMRA under the technology element (Table 1). The technology element can describe agile and non-agile technology (e.g. automated testing, continuous

integration, deployment, communication) that can be used to add value by supporting agile and non-agile project development environments. Technology may be based on various abstraction mechanisms and is compliant to rules, policy and legal requirements. Similarly, other context-specific relationship types and relationships can be derived from technology element perspective. Organization may include additional technology element classes, if required.

4.1.5 Summary

In summary, four ISO/IEC 24744 (2007) SEMDM elements have been identified, empirically analysed and then included in the HAMRA. However, the ISO/IEC 24744 (2007) SEMDM does not explicitly specify the other related elements that were uncovered during this research for developing the HAMRA (as discussed in the following section) - called here ‘extended’ methodology elements: agility, abstraction, business value, business policy, rules, legal, context and facility. These extended elements extend ISO/IEC 24744 SEMDM and cover contemporary thinking and recent developments in software and ISD methodologies. Of course, care has been taken in identifying new candidate elements and classes (extending the ISO/IEC 24744 SEMDM) with regard to whether these new elements or classes under scrutiny are worthy of representation as a new concept (i.e. an element or class in the reference architecture) or whether it is the slot value of an attribute (possibly needing to be added) of an existing element in the ISO/IEC 24744 SEMDM. However, these emerging methodology elements and classes seem to bring interesting insights and further research opportunities for the possible extension of ISO/IEC 24744 SEMDM.

4.2 The HAMRA - agile (extended)

4.2.1 Agility

The adaptive hybrid methodology (Fig. 4) incorporates an agility element. The concept of agility is not new, although its quantitative scientific definition is less well determined. It has been found that existing software development methodology standards (e.g. ISO/IEC 2007, 2014) do not explicitly discuss the element of agility. The need for an agility element has been identified based on the review of basic agility attributes (e.g. Randell and Zurcher 1968; Wong and Whitman 1999; Boehm and Turner 2004; Conboy and Fitzgerald 2004; Henderson-Sellers and Serour 2005; Conboy 2009), agile values and principles (Agile Manifesto 2001), and agile practices of six well-known agile method practices such Extreme Programming (Beck and Andres 2004), Feature Driven Development (Palmer and Felsing 2002), Adaptive Software Development (Highsmith 2000), Dynamic Software Development Method (Stapleton 1997), Scrum

(Schwaber 2007) and Crystal (Cockburn 2006). The Agile Manifesto (2001) specifies the concepts of agility in terms of abstract agile values and principles that are concretized in the agile practices and techniques of these agile methods.

Our empirical analysis has highlighted agile value, agile principle and agile level of the agility element (Table 1). These classes collectively define our agile adoption and improvement model (AAIM) (Qumer 2010). For instance, in case ‘A’, the communication-cooperation protocol had been developed to specify and enable an effective face-to-face communication among the empowered, self-organizing and cross-functional agile team (senior developers of the company) and their clients. The purpose of this protocol was to reduce the documentation and waste while focusing on the development of the new features, rather than the documentation. This practice clearly highlighted the value of the empirical evidence regarding the usability of the three agile values of “reduced documentation”, “individuals and interactions” and “keeping the process cost effective.

Agile values and principles can be organised into six agile levels (see Appendix C). The degree of agility of an adaptive methodology ranges from basic level 0 to advanced level 6. An experimental environment or scenario can be initiated at agile level 0 in which a team may try out some of the individual agile practices and tools prior to beginning with the systematic adoption of specific agile levels and relevant practices (from 1 to 6). For instance, in case ‘B’, “instead of a full scale agile process, we envisaged an experimental scenario and constructed only three agile process fragments: enhanced pair programming (EPP), pair review (PR) and on-site developer (OSD) for a service-oriented e-health project to familiarise the case study organization with an agile approach” (Qumer and Henderson-Sellers 2008b). Each agile level (from 1 to 6) specifies the agile values and principles to be followed in order to achieve the particular level: infancy, initial, realization value, smart and progress. An organization may assess their current agile level and architect a hybrid adaptive methodology for a desired transition(s) and final target agile level (some sort of an agile adoption and improvement roadmap). For instance, in case ‘B’, an agile service oriented process (ASOP) was engineered (by focusing on AAIM level 1 agile practices) to iteratively carry out a systematic test-driven development of the e-health software applications. The concept of an agile practice can be covered by the process element and its technique class; hence, there is no need to define the agile practice as a separate concept or class since these are already included in the HAMRA. [Please see Appendix C and Appendix D for the detailed analysis and mapping of agile values, principles, levels and practices.]

It has been noted in our two industry empirical study cases that agility can be associated with people (agile people: agile actor, role, team, competency), process (e.g. case ‘A’: hybrid agile product enhancement process), work product (e.g. agile

work product: user story backlog, agile executable models) and technology elements (e.g. *agile* technology: agile development platform and infrastructure). Similarly, other context-specific relationship types and relationships can be derived from the agility element perspective. Organizations may also include additional agility element classes, if required, by virtue of method tailoring, as practised in Situational Method Engineering (Henderson-Sellers et al., 2014) – as noted earlier.

4.3 The HAMRA - related concepts (extended)

4.3.1 Abstraction

The adaptive hybrid methodology (Fig. 4) supports abstraction. Abstraction element has been identified based on the underlying paradigmatic mindset¹ (object-oriented and service-oriented) embedded in software and ISD methodologies (e.g. Berzins et al. 1986; Theodorakis et al. 1999; Odell 2002; Feuerlicht 2006). Abstraction, whilst the keystone of a modelling paradigm or mindset, is an important element of a software or ISD methodology (e.g. Berzins et al. 1986) and can be associated with other elements such as people, process, work product (e.g. object-oriented classes, services) and technology (e.g. object-oriented tools to develop classes).

Case study ‘A’ was mainly “focused” on an object-oriented hybrid agile approach whereas case study ‘B’ was focused on a service-oriented agile approach. Based on our analysis of the literature and our empirical evidence, the abstraction element and its two classes have been included in the final version of the HAMRA (Table 1). Organization may include additional abstraction classes, if required.

Furthermore, a software development project may in fact combine more than one abstraction mechanism for a specific situation; for example, one project may involve the use of object-oriented and service-oriented abstractions together. In order to develop such a project, we may need to have a tailored methodology to support both abstractions at the same time. Most of the agile software development methods are not uniquely tied to any specific abstraction mechanism (e.g. object-oriented and service-oriented) or, perhaps more accurately, agile methods are abstraction neutral. However, it is possible that agile and abstraction-based practices can be integrated in order to tailor a situationally-specific multi-abstraction agile or hybrid method (e.g. such as agile object-oriented, and agile service-oriented) for a multi-abstraction project development. Hence, it may be suggested that abstraction is an important element of the proposed reference architecture.

¹ Abstraction element may be named as ‘mindset’.

4.3.2 Business value

The business value element describes the financial and non-financial value that may influence method tailoring and adoption. Traditionally, business value is more concerned with the actual project business value. However, it is important to understand the business value of a methodology, which will be tailored and applied to deliver the project and associated business value. The business value element has been identified as relevant, based on the concepts of business benefits or business values that may be linked to both agile and non-agile or plan-driven methods such as rapid time to market, improved handling of large scale projects, reduced documentation, reduced defect density and improved quality (Royce 1970; Boehm and Turner 2003; Fitzgerald et al. 2006; Elssamadisy 2007; Smith and Sidky 2009). Racheva et al. (2009) suggested that “business value is a key concept in agile software development approaches” and “essentially, in agile software projects, the development process is a value creation process”. Business value can be associated with people (e.g. contribution relevant to people), process (e.g. business value contribution of process or practices), work product (e.g. business value contribution of a work product) and technology (e.g. business value contribution of a development platform) elements. The expected business value can be useful, for example, when architecting a methodology with a view to achieving specific business objectives and when comparing different agile and non-agile methodologies and their contributions to the overall business value. Indeed, the notion of business value is a way to link a methodology to an overarching business strategy. The integration of a business value element to other agile and non-agile elements is one way of mapping the business decisions to major software product and project milestones. It is important to understand the link between business and software development for software business-intensive organizations.

The business value and agile method adoption alignment is an issue that has not been investigated to any great extent by the agile community. Here, based on the analysis of data from our empirical study, we propose that it should be, because it has an impact on both the tailoring and adoption of agile methods in practice (such as the alignment of business and agile software development goals). Case study ‘A’ highlighted the time (e.g. product time to market) and scope (e.g. product features) classes of the business value element. In addition to these two classes, case study ‘B’ also highlighted the cost (e.g. decrease the development cost by about 20 % in comparison with the existing traditional approach), quality (e.g. improve the quality of the services) and training (e.g. train team member) classes of the work product element. These five business value classes have been included in the final version of the HAMRA under the business value element (Table 1).

4.3.3 Policy

The adaptive hybrid methodology (Fig. 4) needs to be compliant to relevant policy. Policy guides the specific behaviour of an organization, which may be expressed in terms of processes to achieve business values (Chapman 1997) or goals (Tyndale et al. 2002). In the context of an agile methodology, Cockburn (2003) suggests that “people construct rules and policies for their upcoming interactions, deciding on work habits, communication standards, project policies and conventions”. The policies are developed and adopted at a high level by top management in order to drive the business of software development or acquisition. Although top management is responsible for the overall success of the business in the market, agile or non-agile or hybrid practices or processes should be owned by the team actually doing the software development work. Nevertheless, it is often their organization’s policies that may have the largest impact on their choices of adoption of agile or non-agile practices.

Senior management and business policies may be one of the influencing elements of software process improvement in terms of agile process adoption and improvement. Policies may guide the organizationally-specific strategic goals and objectives related to software process improvement; the lack of business policy and software process alignment may be one of the resistance factors to the agile process adoption and improvement efforts (Nasir et al. 2008). Reis et al. (2002) suggested an approach to model software development-related policies that may support the reuse of policy instances across different software processes in a software organization. Software development-related policies constitute “the guiding principles for process development and/or enactment” (Feiler and Humphrey 1993). The changes in policy may also impact other associated elements of a methodology. It has been suggested (Vahaniitty and Rautiainen 2008) that “for a software company, it is essential to understand how to link business management and software development and employ a solid, business-oriented approach in its development decision-making”. Hence, it may be suggested that the policy element, together with the other software development elements, is an important element of a software methodology (Kenens et al. 1998) and, therefore, it has been included in the proposed reference architecture. The case study ‘A’ highlighted the product development policy and case study ‘B’ highlighted the project development policy classes of the policy element. These two classes have been included in the final version of the HAMRA under the policy element (Table 1).

4.3.4 Rule

The adaptive hybrid methodology (Fig. 4) needs to be compliant to relevant business rules. ‘Business rule’ is an important element and can be seen as generic statements about the

organization’s way of doing business (Leite 1998). Knolmayer et al. (2000) suggested that “business rules are defined as statements about guidelines and restrictions with respect to states and processes in an organisation”. The rule element may influence method tailoring and adoption. It has been suggested that business rules often drive and govern the software development process in an adaptive and collaborative environment (Orriens et al. 2006). Cockburn (2005) suggested that “projects for systems that can cause more damage need added hardness in the methodology, more validation and verification rules”. Furthermore, Cockburn (2002) discussed the characteristic of “rule” in the context of agile methods as being elaborated into “rules of project behaviour and human-and communication-oriented rules”. Hence, it may be suggested that the rules may be the means to enforce policy and regulatory requirements in order to provide important checkpoints in the processes and practices of an organization. Here, it is included in the proposed reference architecture.

Brinkkemper (1996) describes a software development method as a systematic approach that encompasses directions, rules and a specific way of thinking in order to perform development activities, with the corresponding development of products. It has been suggested (Cockburn 2003) that “treating people’s individual personalities as significant input to the methodology produces the consequences that possible different team members should use different rules within certain bounded work area” and also “every one to three months, they meet for a few hours to a day to discuss the conventions and rules that are operating by and adjust them to better fit their needs”. Both case studies ‘A’ and ‘B’ highlighted the communication, cooperation and decision rule classes of the main rule element. For instance, in case study ‘A’, a communication-cooperation protocol was developed. These three rule classes have been included in the final version of the HAMRA under the rule element (Table 1).

4.3.5 Legal

The adaptive hybrid methodology (Fig. 4) also needs to comply with relevant legal requirements. The legal requirements (compliance and regulatory) may impact the software processes and practices of software business intensive organizations. Although top management is responsible for the business policy, strategy and legal compliance needs, it has been suggested that “organisations have policies in order to: satisfy the business objectives, satisfy customers, make good use of resources, and conform to laws or general business conventions” (Leonardi and Leite 2002). Beznosov and Kruchten (2004) highlighted the legal aspect of the software development process in the context of security-critical requirements. Here, we may argue that top management may be interested in verifying whether the adopted or to-be-adopted

agile practices are in alignment with their business policy, legal and statutory requirements (e.g. especially agility adoption and improvement in the context of large government and military organizations, state departments, ministries). For instance, one of the agile values suggests having an open contract (e.g. “Customer collaboration over contract negotiation”) as opposed to having a fixed contract (e.g. plan-driven). This situation may restrict an organization to adopt this particular agile practice due to their legal compliance requirements of mandatorily having a fixed contract. In another example, the agile practices of “sit together” and “pair programming” may raise legal issues related to personal safety, performance and privacy (e.g. legal and regulatory compliance - industrial award or certified agreements). Also, on another note of agile practices, an “on-site customer” may be exposed to the information (e.g. personal, project, contract) of other customers, which may not be desirable from the perspective of local Freedom of Information (FOI) Acts (e.g. Commonwealth FOI ACT 1982) or commercial in confidence. In other words, it may be suggested that the business policy, rule and legal elements are tied to the adoption and tailoring of agile and non-agile methodology. Here, the legal element is included in the proposed reference architecture and empirically analysed. Both studied organizations need to be compliant to legal requirements for a mutually agreed contract for a specific product and service. Hence, the contract class has been identified and included in the HAMRA under the legal element. Organizations may also include additional legal element classes according to their context using method tailoring.

4.3.6 Context

The eleventh element of the HAMRA construct is an overarching context element that describes the overall environmental context, which is associated with all other methodology elements. It describes a context or environment in which a methodology is developed and applied. Context is a project environment or situation that can be characterized by the subject area or problem domain in a particular organization (Rolland and Prakash 1996). A project context can be described by using a set of factors called contingencies or situational factors (Harmsen et al. 1995). An organization or an extended organization includes both its internal and external environment, which incorporates not only its internal business units but also its partners, suppliers and customers (Harmsen 1997; Kornysheva et al. 2011; Qumer and Henderson-Sellers 2009). It is important to identify and understand the contextual information of a given situation or project environment when tailoring a method (Bucher 2006). Tolfo and Wazlawick (2008) pointed out that “it is necessary to choose a software development method that suits the organizational culture”. The empirical study highlighted that “a step-by-step approach may be considered reasonable for a gradual, successful

transition or adoption of agile ideas, rather than all at once, which may pose several risks and problems”. Our empirical study has suggested the inclusion of classes to represent nature (e.g. size, complexity, and process change risk aspects of the large and complex environment of case ‘A’ organization), industry (e.g. case ‘B’ consulting organization), domain (e.g. case ‘B’ e-Health), structure (e.g. formal organizational structure) and culture (formal process-driven exiting culture of case ‘A’ organization) within the context element; hence these have been included in the final version of the HAMRA (Table 1). Organizations may include additional context classes appropriate to their context.

4.3.7 Facility

The adaptive hybrid methodology (Fig. 4) has an associated facility element, which was identified during the empirical analysis. The initial construct of the HAMRA containing eleven elements provided sufficient methodology coverage. However, the empirical study analysis uncovered a new methodology “facility” element, which was not present in the initial HAMRA construct. Facility provides the physical or virtual and co-located or distributed spatial workspace to support other methodology elements. For instance, case study ‘A’ highlighted the need for co-located workspace for developers. Facility is an important element, because some of the agile practices such as face-to-face communication or daily stand up meetings may not be feasible if the team members are not co-located or in the same facility. Hence, the facility element and spatial class have been included in the final version of the HAMRA (Table 1).

Here, we learned through the theoretical and practical empirical study analysis that a hybrid adaptive methodology can be configured using the HAMRA elements. This highlights the theoretical and practical relevance of the HAMRA elements. Furthermore, it can be observed from case study ‘A’ that it is possible to have a hybrid adaptive methodology (see Appendix A) for a large organization instead of a full scale agile method such as is the case for case study ‘B’, which can be built or tailored by mixing agile and existing traditional (non-agile) elements. Based on the theoretical and practical empirical analysis, a set of twelve integrated hybrid adaptive methodology elements and their relevant classes have been identified, which then forms the proposed revised HAMRA. The HAMRA incorporates already well-established four core elements (process, people, work product, technology) and newly identified eight extended elements (agility, abstraction, business value, policy, rules, legal, context and facility). The HAMRA is a blueprint that describes the methodological elements and their relationships at an abstract or high level, allowing for drilling down for each of these identified element’s classes as needed for a specific context.

5 Application

This section demonstrates how the final version of the HAMRA multidimensional viewpoint (Table 1, Fig. 4) has been applied in configuring two hybrid adaptive methodologies for teaching and developing software projects in two teaching cases (cases ‘C’ and ‘D’) at UTS. In teaching case ‘C’ at UTS, a hybrid adaptive business analysis (BA) methodology was designed by using the HAMRA multidimensional viewpoint or template for teaching the large-scale project-based undergraduate business requirements modelling (BRM) subject and delivering associated project requirements (see Table 2). In teaching case ‘D’ at UTS, a hybrid adaptive project development (PD) methodology was designed by using the HAMRA multidimensional viewpoint for teaching project-based software engineering practice (SEP) subject (see Table 2). This was done to further evaluate the applicability of the final version of the HAMRA elements. This is also a way to link theory and industry based empirical research to teaching. This completes a full cycle of theory-empirical research-teaching. These methodologies are incorporated into the delivery of both subjects. [Please see Table 2, which shows the application of HAMRA elements and created methodologies in the academic project context.]

5.1 Adaptive requirements methodology

The project-based BRM subject introduces students to agile and non-agile BA elements. The goal is to architect a hybrid adaptive BA methodology for the Job Advertisement and Application Submission (JASS) system analysis team project. In order to do so, we used the HAMRA multidimensional viewpoint (Fig. 4) to architect a hybrid adaptive BA methodology (see Table 2 – BRM Teaching Case) through the integration of traditional BA elements from Business Analysis Body Of Knowledge (BABOK) (IIBA 2009) and agile BA elements from the Scrum (Schwaber 2007) (published in the public domain). The project brief and the created hybrid adaptive BA methodology were explained to the students by the lecturer at the beginning of the semester. Students applied the adaptive BA methodology to their projects in small teams (2–3 students in a team) during the semester of 14 weeks. The adaptive BA methodology is organized into two stages and four activities (Fig. 5). The details of the adaptive BA methodology are mapped in Table 2.

5.2 Adaptive PD methodology

The project-based SEP subject introduces students to agile and non-agile PD elements. The goal is to architect a hybrid adaptive PD methodology for the Travel Funding Approval (TFA) system team project. In order to do so, we used the HAMRA multidimensional viewpoint (Fig. 4) to architect a hybrid adaptive PD methodology (see Table 2 – SEP Teaching Case) through the integration of traditional waterfall (Royce 1970)

together with agile elements from Scrum (Schwaber 2007) and Extreme Programming (Beck and Andres 2004) methodologies (published in the public domain). The project brief and the created adaptive PD methodology were explained to the students (different from the first teaching case) by the lecturer at the beginning of the semester. Students applied the adaptive PD methodology to their projects in small teams (5–7 students in a team) during the semester of 14 weeks. The adaptive PD methodology is organized into three stages and six activities (Table 2, Fig. 6 with marked circle). The details of the adaptive PD methodology elements are mapped in Table 2.

The people element classes were used to describe the people dimension of both the hybrid adaptive BA and PD methodologies (Table 2). For instance, students played the role of business analyst (traditional role), whereas the teaching staff played the role of a product owner (agile role) in case ‘C’. Similarly, students played different agile and traditional roles in case ‘D’.

The process element classes describe the adaptive BA and PD methodologies (Table 2). For instance, the activity class of the process element describes two BA and six PD activities. It can be observed from Table 2 (see case ‘D’ Task) that, instead of traditional detailed up-front planning and analysis, the planning and analysis activity tasks are done iteratively at different levels – from the high-level project to just-in-time iteration-level planning and analysis (pre-iteration). Furthermore, it is important to note that the technique class describes both agile and non-agile techniques to support the tasks of hybrid adaptive BA and PD methodologies.

The work product element classes describe the traditional non-agile (e.g. requirements specifications document, uses cases) and agile work products (e.g. user story backlog). Furthermore, it also highlights the abstraction-specific object (cases ‘C’ and ‘D’) and service (case ‘D’) work products. Students as business analysts, in the adaptive BA methodology (case ‘C’), mainly focus on capturing requirements using high level user-story work products and clarifying those user stories during face-to-face conversations with the product owner (tutor or lecturer) during the tutorial sessions. It is important to note that in the hybrid work product context, students only detail complex user stories in terms of system use case model and narratives.

The technology element classes describe the platform and infrastructure that are required to support the adaptive BA and PD methodologies. For instance, in the adaptive BA methodology, agile (e.g. IBM Jazz Hub) and traditional non-agile (e.g. MS Office MS Visio) platforms and infrastructure have been included in order to support the production of agile (e.g. product backlog) and traditional non-agile (e.g. requirements specifications document) work products. It is important to note here that students were allowed to use any additional or different technology, other than that specified in the adaptive methodologies and provided by UTS labs.

The agility element classes refer to the agile values, principles and levels supported by the adaptive BA and PD methodologies.

Table 2 The HAMRA teaching cases – multidimensional viewpoint

HAMRA			BRM Teaching Case ‘C’	SEP Teaching Case ‘D’
#	Element Dimension	Class	Adaptive BA Methodology	Adaptive PD Methodology
1	People	Individual	Student Staff	Student Staff
		Team	BRM Student Team (2–3 Students Per Team)	SEP Student Teams (5–7 Students per Team)
		Organization	UTS	UTS
		Actor	Individuals	Individuals
		Role	Student: Business Analyst Staff: Product Owner	Student: Business Analyst, Architect, Developer, Tester, Iteration Manager, Release Manager, Project Manager Staff: Product Owner, Program Manager, Portfolio Manager, Sponsor
		Competency	Student: Business Analysis Staff: Product Owner	Student: Software Engineering Staff: Product Owner, Management
		Capacity	Student: 9–12 Hours Per Week Staff: 3–5 Hours Per Week	Student: 9–12 Hours Per Week Staff: 3–5 Hours Per Week
2	Process	Event	Project Start Assessment Start Assessment Reviews Assessment End Project End	Project Start Assessment Start Showcases Assessment End Project End
		Gateway	Two Assessments	Three Assessments
		Stage	Planning and Elicitation Analysis and Management	Discovery Design Development
		Activity	Planning and Elicitation: Requirements Planning Requirements Elicitation Analysis and Management: Requirements Analysis Requirements Management	Discovery: Planning Analysis Design: Architectue Design Development: Implementation Testing
		Task	Requirements Planning: Conduct Stakeholder Analysis Plan Analysis Activities Plan Analysis Communication Plan Requirements Management Plan Analysis Performance Requirements Elicitation: Prepare for Elicitation Conduct Elicitation Document Elicitation Results Confirm Elicitation Results Requirements Analysis: Organize Requirements Prioritize Requirements Specify and Model Requirements Determine Assumptions and Constraints Verify Requirements Validate Requirements	Planning and Analysis: Conduct Adaptive Project Planning and Analysis Conduct Adaptive Release Planning and Analysis Conduct Adaptive Iteration Planning and Analysis Architectue and Design: Conduct Adaptive Project Architecture and Design Conduct Adaptive Release Architecture and Design Conduct Adaptive Iteration Architecture and Design Implementation and Testing: Conduct Adaptive Iteration Implementation Conduct Adaptive Iteration Testing

Table 2 (continued)

HAMRA			BRM Teaching Case ‘C’	SEP Teaching Case ‘D’
#	Element Dimension	Class	Adaptive BA Methodology	Adaptive PD Methodology
		Technique	Requirements Management: Manage Solution and Requirements Scope Manage Requirements Traceability Communicate Requirements Brain Storming Process Modelling User Story User Story Mapping User Story Narrative Use Case Modelling Object Modelling Interface Modelling	Process Modelling User Story User Story Wall User Story Mapping User Story Narrative Use Case Modelling Object Modelling Database Modelling Service Modelling Interface Modelling Agile Analysis Agile Planning Agile Architecture Agile Design Agile Implementation – pair programming, test-driven development Agile Testing – automated unit testing (e.g. JUnit) and acceptance testing (e.g. JBehave)
3	Work Product	Backlog	User Story Backlog	User Story Backlog Project Tracking Defect Log Risk and Issue Register Individual Contribution Log
		Document	Requirements Specifications Data Dictionary	Requirements Analysis and Planning Architecture and Design Implementation and Testing
		Model	Process Model Data Model Use Case Model Object Class Model Object Interaction Model Object State & Event Model User Interface Model	Process Model Solution Architecture Model Use Case Model Backend Data Model Object Class Model User Interface Model Service Model
		Matrix	Requirements Matrix	Requirements Matrix Test Matrix
		Component Software Application	JAAS Software Components JAAS Software Application (Prototype)	TFA Software Components TFA Software Application
		Interface Service	-	TFA Service Interfaces TFA Services
4	Technology	Platform	MS Window MS Office MS Visio UTSONline Email Facebook IBM Jazz Hub	MS Window MS Office MS Visio UTSONline Email Facebook IBM Jazz Hub J2EE Platform .Net .Net Platform DB Platform

Table 2 (continued)

HAMRA		BRM Teaching Case ‘C’	SEP Teaching Case ‘D’	
#	Element Dimension	Class	Adaptive BA Methodology	Adaptive PD Methodology
		Infrastructure	MS Window Work Station MS Window Server UTSOnline Server Email Server Facebook Server IBM Jazz Hub Server	Jenkins Platform (CI) Go Platform (CD) GitHub MS Window Work Station MS Window Work Station MS Window Server UTSOnline Sever Email Server Facebook Server IBM Jazz Hub Server J2EE Server .Net Server DB Server Jenkins Server (CI) Go Platform (CD) GitHub Server
5	Agility	Agile Value Agile Principle Agile Level	Agile Levels 1–2 Values Agile Levels 1–2 Principles Agile Levels 1–2	Agile Levels 1–3 Values Agile Levels 1–3 Principles Agile Levels 1–3
6	Abstraction	Object Oriented Service Oriented	Object Class Object Interaction Object State and Event -	Object Class Component Service Interface Service
7	Business Value	Time Scope Cost Quality Training	- - - - Student Training	- - - - Student Training
8	Policy	Product Development Policy Project Development Policy	- Workload Policy	- Workload Policy
9	Rule	Communication Cooperation Decision	Face-to-face lectures (1 h) and tutorial sessions (2 h) per week Consulting (2 h) per week Email communication rules Collaboration log books Assessment Rules	face-to-face lectures (1.5 h) and tutorial sessions (1.5) hours Consulting (2 h) per week Email communication rules Collaboration log books Assessment Rules
10	Legal	Contract	Subject Outline	Subject Outline
11	Context	Nature Industry Domain Structure Culture	Academic Project Education Recruitment Management Hybrid Collaborative	Academic Project Education Finance Hybrid Collaborative
12	Facility	Spatial	UTS Lab (Physical) Home (Physical) Internet (Virtual)	UTS Lab (Physical) Home (Physical) Internet (Virtual)

The adaptive BA methodology incorporates the agile levels 1–2 values, principles and relevant agile practices as techniques (see Appendix C). The adaptive BA methodology includes the basic

levels of agility since it is a first year undergraduate subject. The adaptive PD methodology incorporates agile levels 1–3 values, principles and relevant practices. The adaptive PD methodology

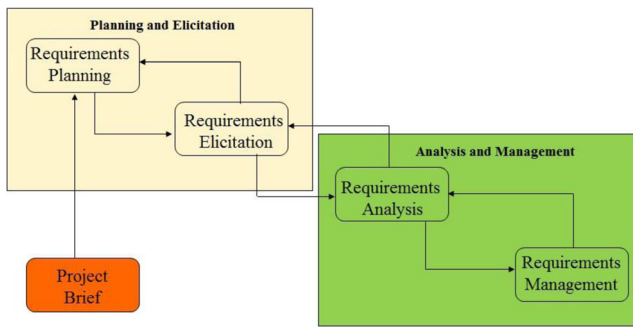


Fig. 5 Hybrid adaptive BA methodology architecture – stage and activity view

includes the mid-levels of agile (better than BA) since it is a second year undergraduate subject. The adaptive methodologies incorporating advanced agile levels, values, principles and relevant practices can be architected for advanced adaptive BA and PD subjects.

The abstraction element classes describe the abstraction dimension of the adaptive BA (object oriented) and PD (e.g. object and service oriented) methodologies. It is important to note here that although agile practices originated in the context of object-oriented development, agile is not in fact tied to any specific abstraction mechanisms (e.g. object or service). It seems to be abstraction neutral. However, it is important to understand the abstraction dimension from the abstraction-specific work product, technology and technique perspective.

The business value element classes are mapped to both adaptive BA and PD methodologies. Since it is an academic project, only the training business value is relevant and mapped in this context. Other business value element classes can be mapped to commercial project methodologies.

The adaptive BA and PD methodologies were tailored for the academic projects; therefore, only the project development policy

class of the policy element was relevant. The project development policy states the UTS student workload policy for the subjects in the context, which is 9–12 h per student per week (including 3 h face-to-face teaching time per week). The tasks in the adaptive methodologies should be compliant to this policy. Similarly, rule class elements such as communication, cooperation and decision rules (e.g. UTS assessment rules) guide the BA and PD methodologies. For instance, in the hybrid adaptive BA, agile rules (e.g. rules of face-to-face human communication) and non-agile rules (e.g. requirements document driven communication) have been integrated. The legal contract element class refers to the subject outlines, which is a legal contract between students and UTS. This contract guides the assessment tasks in the adaptive BA and PD methodologies. Similarly, the context element classes describe the academic context for the adaptive BA and PD methodologies. Finally, the facility element class ‘spatial’ refers to the UTS lab, student home and internet facilities supporting the execution of adaptive BA and PD methodologies.

This section has presented the application of the final version of HAMRA. It is clear from the analysis of HAMRA that the HAMRA is not another agile or non-agile methodology. HAMRA is a reference architecture or guide; a hybrid adaptive methodology can then be established or represented by using the HAMRA elements and classes for a particular situation. The HAMRA construct (e.g. a vocabulary and conceptualization of agile and non-agile environment) has laid down a foundation for possible future investigation (and future research) of a standard metamodel for hybrid adaptive methodologies.

6 Discussion

HAMRA should perhaps be considered as a living organism (living systems theory of Miller 1995), which is continually

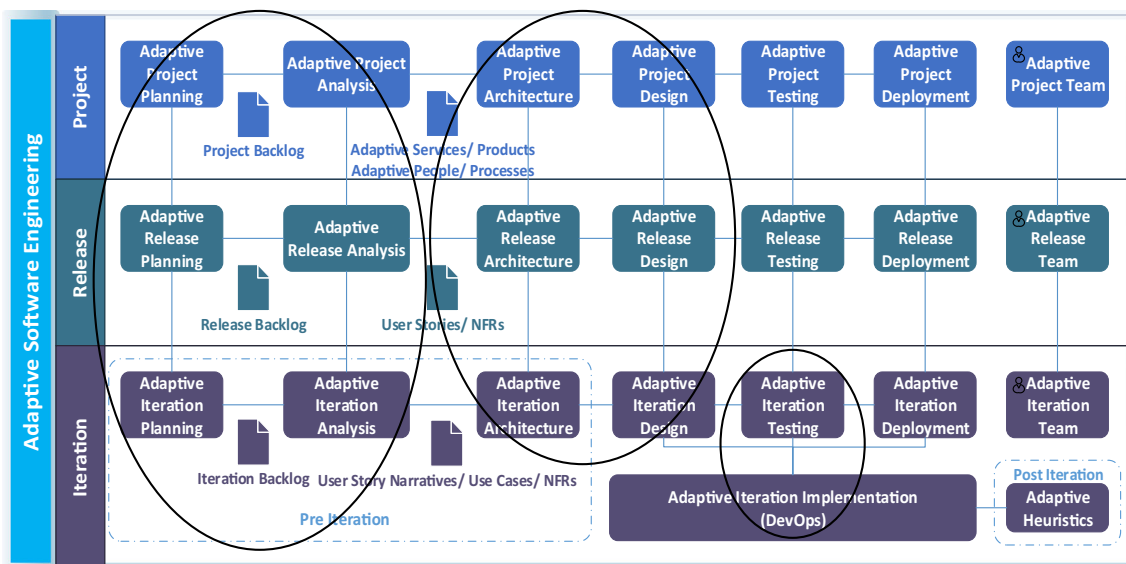


Fig. 6 Hybrid adaptive PD methodology architecture – stage and activity view

evolving and adapting. Living systems are created, used, maintained, transformed and retired or expired. Living system organisms are composed of nested parts or cells. According to the living systems metaphor, each living system at the higher level is made of other low-level parts or cells. In the HAMRA, an adaptive methodology is a living system, where each element is thus perceived as a “Cell”. These cells can be combined to architect an agile or hybrid adaptive methodology. The advantage of using the “Cell” metaphor of the living system theory is that it presents the holistic and dynamic nature of the HAMRA and the subsequently created methodologies. It indicates that these should both be architected and managed as a whole of interdependent and connected elements or cells instead of a collection of separately architected and managed elements (see multidimensional viewpoint – Fig. 4). It thus takes into consideration the inter-dependency and interactions of methodology cells because changes in one part or cell of a methodology can impact other parts or cells of that methodology. This is critical because changes made in one part while ignoring the whole may channel the methodology energy in the wrong direction and could be a recipe for future methodology malfunction or failure. The HAMRA is flexible and can be extended, localized and modified by the development team(s), using situational method engineering. The elements or dimensions or cells embedded in the HAMRA can be integrated to generate various situationally-specific configurations of hybrid adaptive methodologies for project development. The concreteness and contribution of the proposed HAMRA are discussed from both practice and research perspectives within the overall context of its scope and limitations.

6.1 Practice

The HAMRA construct has been incrementally developed over a period of time (see Section 3). Firstly, the initial version of the HAMRA was developed based on the literature review (as discussed earlier). This initial HAMRA construct was applied to two industrial empirical cases: A and B (see Table 1) to get early feedback and directions for further development. We conducted process workshops in industry and created hybrid agile product-enhancement process (APEP) and agile service oriented process (ASOP) by using the initial HAMRA construct for industrial cases A and B for two different organisations (see Appendix A and B for workshop photos). It was done to determine the industrial applicability and soundness of the HAMRA elements to ensure that the construct in hand represents the domain of interest and is fit for the purpose of software process or method creation. The results of these cases highlighted that a step-by-step method tailoring approach along with a structured framework, such as HAMRA, is appropriate for a gradual and successful adoption of hybrid and agile practices for large and complex projects. The results of these empirical cases were described in (Qumer and

Hnderson-Sellers 2008b), which laid a foundation for further development of the HAMRA in its current form. Secondly, these empirical study results were analysed and integrated with the detailed theoretical analysis work (see Table 1) to further identify the hybrid methodology element classes, relationships and also additional elements (see Figs. 3 and 4) for the final and refined version of the HAMRA construct. Finally, the practical applicability and concreteness of the refined version of the HAMRA was further tested by successfully using it in creating two hybrid adaptive methodologies for practice-oriented undergraduate software projects (cases C and D) in academic settings (see Section 5). This rigorous three-fold evaluation of the practice and theory based HAMRA construct clearly indicates its practical applicability and relevance.

6.2 Research

The final version of the HAMRA was compared with three related frameworks such as ISO/IEC 24744 SEMDM International Standard (2007), Agile Scrum Framework (Schwaber 2007) and the Four-Tiered Framework of Iivari et al. (2000). This was done to ensure that the important elements are not overlooked in the proposed HAMRA construct. The comparative analysis clearly highlighted the concreteness and research contribution of the HAMRA (see Table 3).

Firstly, HAMRA was compared with the ISO/IEC 24744 SEMDM. It is clear from the comparative analysis (Table 3) that the HAMRA construct incorporates both agile and traditional methodology elements (e.g. light grey coloured column) in addition to the already well-established ISO/IEC 24744 SEMDM traditional methodology elements. As noted earlier, ISO/IEC 24744 SEMDM seem to support only non-agile traditional methodology elements. Secondly, HAMRA was compared with the well-known and widely used agile industry framework of Scrum (Schwaber 2007). It has been found (see Table 3) that the contemporary Scrum framework discusses most of the elements of the HAMRA except methodology abstraction, legal, context and facility elements. Scrum seems to be an abstraction-neutral framework and claims to be used with a range of abstraction mechanisms. It also does not explicitly discuss the policy element. However, it seems that the policy element is embedded in its agile principles. Finally, HAMRA was compared with the generic Four-Tiered Framework (Iivari et al. 2000). This framework provides a four-layered architecture approach to classifying the ISD: ISD Paradigms (ISDP), ISD Approaches (ISDA), ISD Methodologies (ISDM) and ISD Techniques (ISDT). The scope of the HAMRA is limited to software and ISD Methodologies; therefore, we compared it to the ISDM layer of the Four-Tiered Framework with the HAMRA. The Four-Tiered Framework ISDM layer discusses the process element (e.g. relationship between techniques) and, surprisingly, it

Table 3 Comparison and contribution

Ref.	HAMRA	ISO/ IEC 24744 SEMDM (2007)	Schwaber (2007) Scrum	Iivari et al. (2000) The Four-Tiered Framework
1	Process	Process	Process	Process
2	Technology	Tool	Tool	Tool
3	People	People	People	X
4	Work Product	Work Product	Work Product	Work Product
5	Agility	X	Agility	X
6	Abstraction	X	X	Abstraction
7	Business Value	X	Business Value	Business Value
8	Policy	X	Policy (Agile Principles)	Policy
9	Rule	X	Rule	X
10	Legal	X	X	X
11	Context	X	X	X
12	Facility	X	X	X

Columns in grey colour show the contribution of this paper, whereas the last column indicates the elements of the ISO/IEC 24744 SEMDM, Scrum and the generic Four-Tiered Framework. ‘X’ symbol indicates the absence of the methodology elements

does not discuss the most important people element in the ISDM layer. The ISDM layer also inherits attributes from the ISDA layer (e.g. abstraction, goal, principles). Based on our detailed analysis, we found that ‘goal’ element seems to refer to business value element of the HAMRA, whereas ‘principles’ maps to the policy element of the HAMRA. The Four-Tiered Framework ISDT layer, below the ISDM layer, seems to discuss the technology and work product elements under the category of techniques. Overall, the Four-Tiered Framework seems to incorporate most of the elements of the HAMRA except the methodological people, agility, rule, legal, context and facility elements. This comparative analysis clearly indicates the concreteness and comprehensiveness of the HAMRA. It provides the elements, as a research contribution, which have not been discussed earlier (see Table 3).

The final version of the HAMRA was compared with three related frameworks such as ISO/IEC. The measurable benefits of agile approaches such as improved time-to-market, productivity and quality of software have already been demonstrated, studied and reported in the literature (Reifer 2002). The challenge for organisations is how to adopt agile at the large scale rather why do we need to adopt agile? This paper addresses this important concern of “how” and proposed the HAMRA construct. The analysis of the HAMRA, both from practice and research perspective, clearly highlights that it provides a comprehensive and concrete set of the agile and non-agile methodology reference elements and associated

classes that can be used as a lens to create an architecture of a situation-specific hybrid adaptive methodology for adopting agile at the large scale (as demonstrated earlier). The HAMRA is intended to be used by agile and non-agile teams as a blueprint for understanding the overall picture of the agile and non-agile methodology elements.

6.3 Scope and limitations

The HAMRA should not be viewed as providing fine-grained details and exhaustive index of methodology elements and classes. It is an evolving construct and should be viewed with its scope and limitations. It is also not another agile, non-agile or hybrid methodology either. There are a number of agile [e.g. XP (Beck and Andres 2004), Scrum (Schwaber 2007)] and non-agile methodologies [e.g. Waterfall (Royce 1970), Spiral (Boehm 1988)]. These well-known agile methodologies, together with the Agile Manifesto (2001), originated in the context of small to medium project development environments, offering a collection of agile values, principles and industry best practices. However, they lack empirical evidence and theoretical underpinning. Traditional methodologies offer practices for large project development environments that lack agility whereas an agile response is preferable for tackling unclear or changing project requirements. Agile and non-agile traditional methodologies have their own individual advantages over each other (Qumer and Henderson-Sellers

2008a). Therefore, it seems appropriate to have a hybrid methodology (e.g. a combined set of agile and non-agile practices) tailored for a specific situation by mixing both agile and plan-driven approaches to support the large-scale environment. However, a hybrid adaptive methodology reference architecture is required to represent and tailor a hybrid adaptive methodology. The proposed HAMRA construct developed here attempts to fill a small part of this gap. It builds on the empirical studies and also has theoretical underpinning and rigor, and integrates both agile and non-agile elements, which have not been previously analysed as we discussed in this paper.

HAMRA is a theoretical and practical construct, it is intended to capture and convey the high level view of the hybrid methodology elements and their relationships that can be detailed as needed for architecting a situationally-specific hybrid adaptive methodology. The HAMRA construct that we have developed and extended here needs to be considered with a view of its limitations since the body of literature and practice are both dynamic in nature. It should be considered as an ongoing work to be revised and extended by future studies. Given the project scope and time constraints, this study is limited to the number of case studies, qualitative analysis and finite number of theoretical concepts. However, based on the initial results, we are fully confident that the proposed HAMRA construct elements provide sufficient coverage and guidance for developing hybrid methodologies. It is important to mention that there was no relationship bias between the researchers and empirical study organisations. The analysis, coding and labelling of HAMRA elements are subject to human error and mistakes, which may lead to inconsistencies. The elements and their interconnections were identified by the first author independently, and the second and third were continuously involved in critically reviewing each element. This review was performed iteratively to get feedback and minimize any possible omissions, errors or analysis bias. The review feedback, and disagreements were discussed and resolved during review meetings involving the second and third authors, who have more experience in IS modelling and methodologies.

7 Conclusion and future work

Agile approaches are based on industry best principles and practices; and have been criticized for lacking empirical evidence and theoretical underpinning. The research reported here attempts to fill a small part of this gap (between theory and practice). Here, the HAMRA conceptual construct has been presented based on an

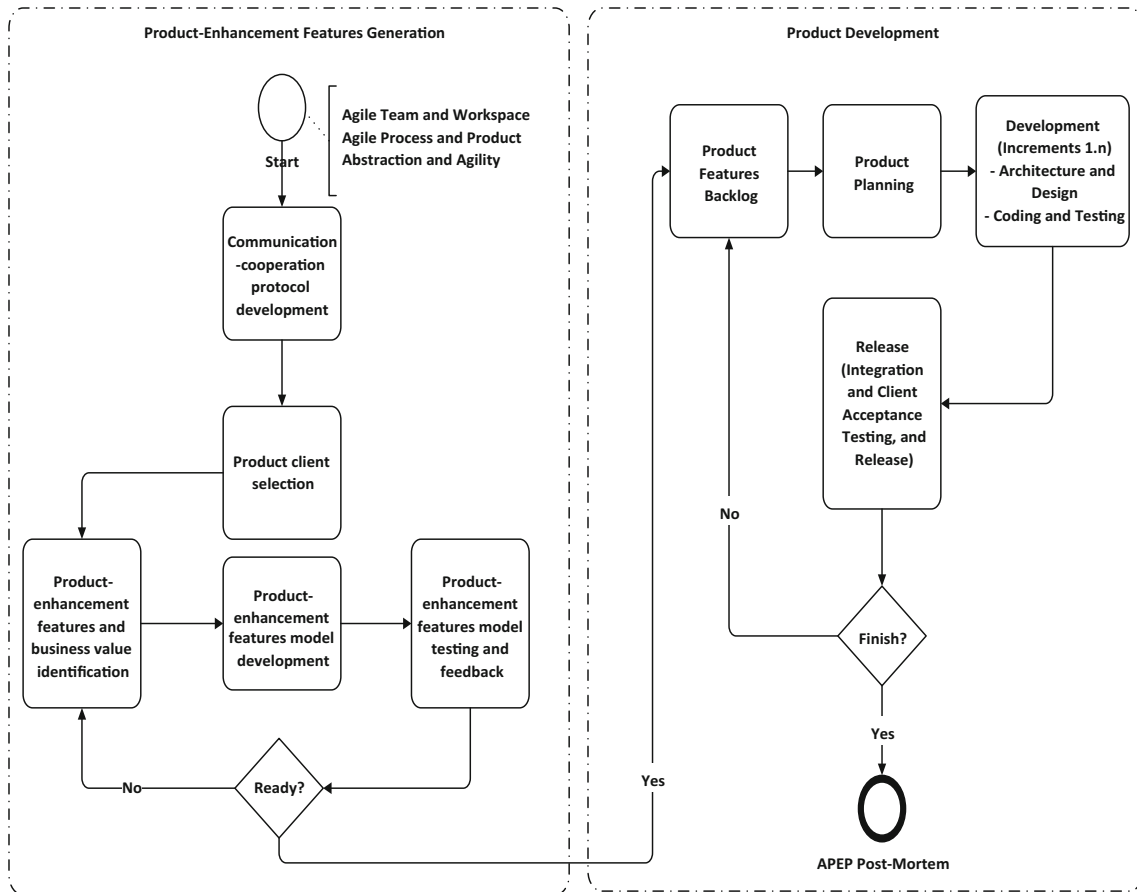
analysis of both literature and qualitative empirical studies. A number of existing well-established (people, process, product and technology) and new (agility, abstraction, business value, rules, policy, legal, context and facility) elements have been discussed. These new elements and HAMRA as a whole construct initiate research into possible extensions to the ISO/IEC 24744 SEMDM International Standard that will both support agility concerns and also modify the standard in a seamless way – by using the extension procedure detailed in Henderson-Sellers and Gonzalez-Perez (2006).

The HAMRA construct is thus a first step towards the possible future investigation of a standard metamodel for hybrid adaptive development methodologies. Based on analysis presented in this paper, the HAMRA elements can be used as a checklist or vision-guiding reference architecture when constructing various situation-specific agile and hybrid methodologies. Further research may investigate whether there are other elements that can be included in HAMRA. An empirical study can be conducted to understand the impact of identified HAMRA elements on the project success factors such as quality, time, scope, budget, and customer satisfaction. Similarly, a design-oriented and action-oriented study can be conducted for building, intervening and evaluating (BIE) the details of different types of HAMRA artefacts for different industrial contexts (e.g. health, financial services, manufacturing). This research is further implementing the proposed HAMRA and developing a full scale context-aware hybrid adaptive methodology information system. The initial prototype has been developed and deployed as an agile e-toolkit in the public cloud environment to support the teams to self-tailor agile and hybrid methodologies for their local context (see Qumer and Henderson-Sellers 2010). The discussion of the agile e-toolkit is beyond the scope of this paper due to space limitations. This paper concluded with the development of a theoretical and practical HAMRA construct as a prelude to likely further research directions in this important area.

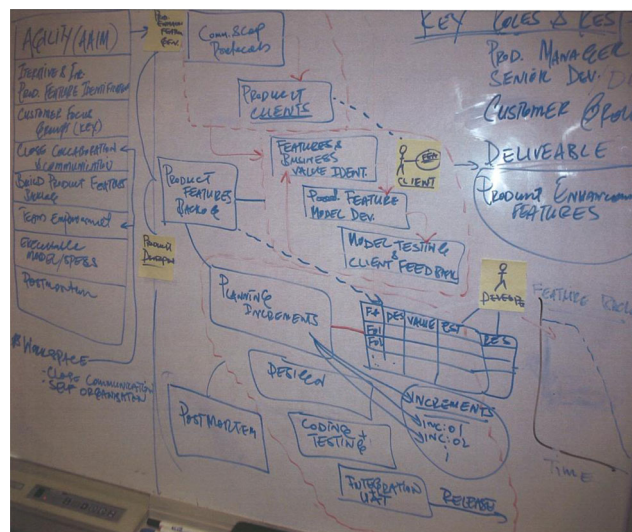
Appendix 1: Hybrid agile product-enhancement process model (Qumer and Henderson-Sellers 2008b)

The hybrid agile product-enhancement process has two main parts: feature generation and product development. The feature generation focuses on the identification of new features for enhancing the product. The product development focuses on incremental product architecture, design, coding, testing,

integration and release. This combines both the agile and traditional approaches.



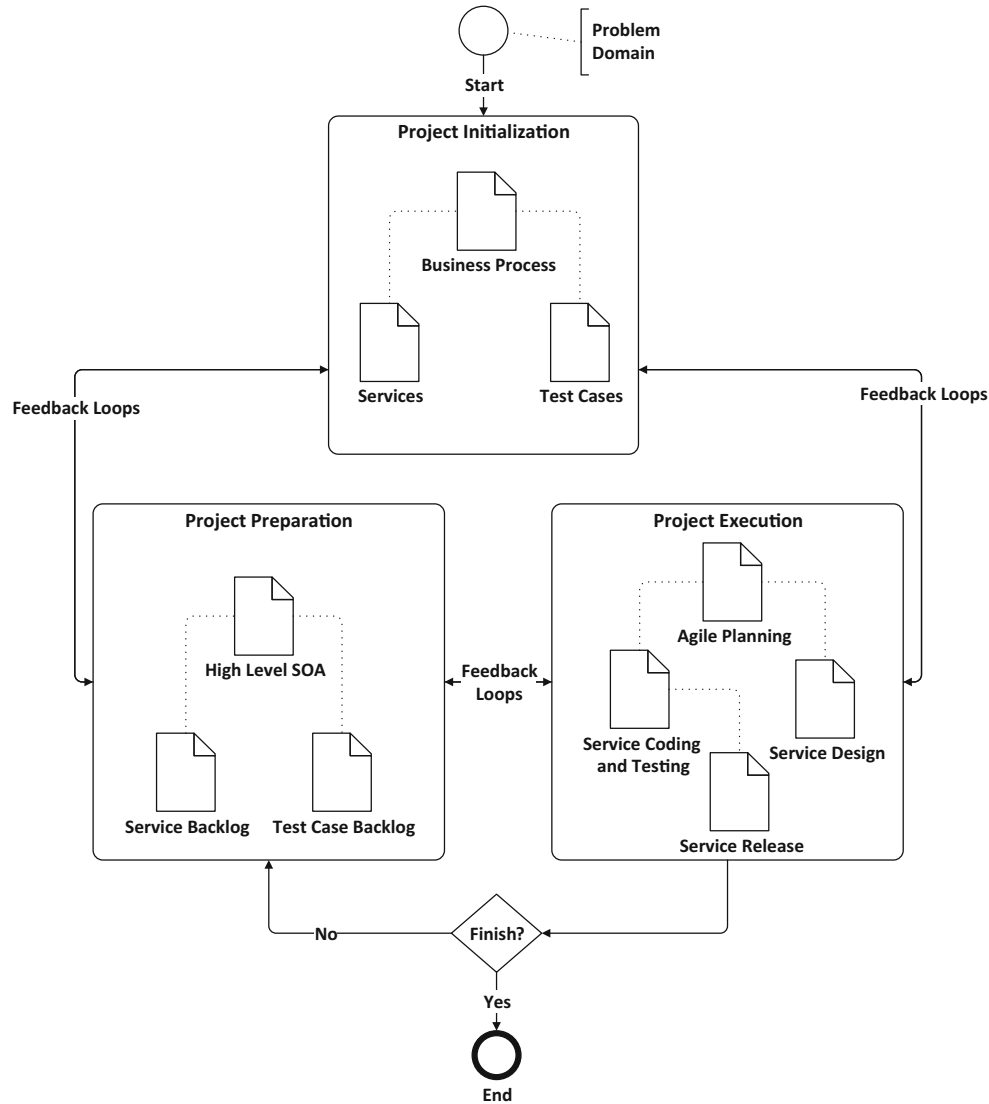
The hybrid agile product-enhancement process was tailored in a workshop conducted in industry in a case study organisation. The following figure is from the process tailoring workshop screenshot.



Appendix 2: Agile service oriented process model (Qumer and Henderson-Sellers 2008b)

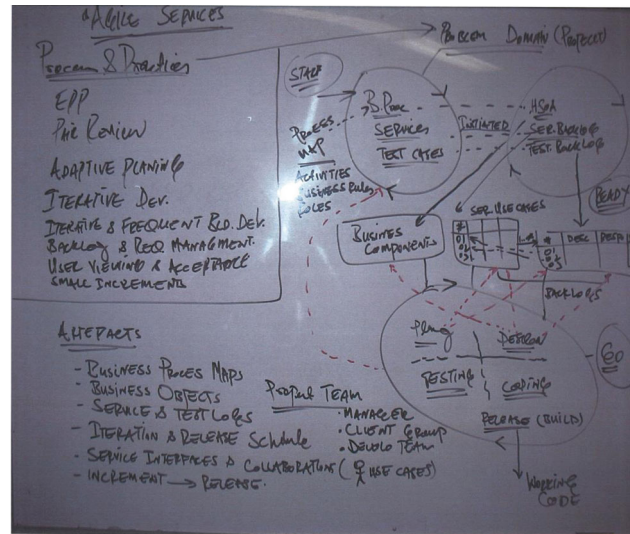
The agile service-oriented process has three main parts: project initialization, project preparation and project execution. The project initialization focuses on the identification of

services for the project in hand. The project preparation focuses developing the service backlog and architecture. The project execution focuses on iterative planning, design, coding, testing and release of services. This is a case of abstraction-specific (service oriented) process that combines both the agile and service oriented approaches.



The agile service oriented process, similar to the first process, was tailored in a workshop conducted in industry in a

case study organisation. The following figure is from the process tailoring workshop screenshot.



Appendix 3: Agile values, principles and levels

The following tables show the mapping of agile values, principles and corresponding agile levels that form the agile adoption and improvement model.

Agile Principles

- Early and Continuous Delivery
- Welcome Change
- Frequently Delivery
- Motivate Individuals
- Collaborative Work
- Face to Face Conversation
- Working Software
- Simplicity
- Self-Organizing Team
- Reflection and Tuning
- Sustainable Development
- Enhance Agility

Agile Values

- Customer collaboration over contract negotiation
- Responding to change over following a plan
- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Keeping the process cost effective

- Keeping the process agile
- Keeping the process agile
- Keeping the process cost effective

Agile Levels	Name	Encourage & Focus	Agile Principles
Level 1	Agile Infancy	Evolutionary	Early and Continuous Delivery Welcome Change Frequently Delivery Motivate Individuals
Level 2	Agile Initial	Collaborative	Collaborative Work Face to Face Conversation
Level 3	Agile Realization	Simple Results Focused	Working Software Simplicity
Level 4	Agile Value	People Focused	Self Organizing Team
Level 5	Agile Smart	Knowledge Focused	Reflection and Tuning
Level 6	Agile Progress	Lean Agile	Sustainable Development Enhance Agility

Appendix 4: Agile values and practices mapping

The following table shows the mapping of agile values and corresponding agile practices distilled from six well-known agile methods.

Agile Values	XP	Scrum	FDD	ASD	DSDM	Crystal
Individuals and interactions over processes and tools	<ol style="list-style-type: none"> 1. Sit together 2. Whole team 3. Informative workspace 4. Energized work 5. Pair programming 	<ol style="list-style-type: none"> 1. Scrum teams 2. Sprint planning meeting 3. Daily scrum meeting 4. Collocation 	<ol style="list-style-type: none"> 1. Domain object modelling 2. Individual class ownership 3. Feature teams 4. Inspection 	<ol style="list-style-type: none"> 1. Adaptive management model 2. Collaborative teams 3. Joint application development by independent agents 4. Customer focus group reviews 	<ol style="list-style-type: none"> 1. Active user involvement 2. Empowered teams. 3. Collaboration and cooperation among stakeholders 	<ol style="list-style-type: none"> 1. Staging. 2. Parallelism and Flux. 3. Holistic diversity and strategy 3. User viewings. 4. Personal safety 5. Daily stand-ups
Working software over comprehensive documentation	<ol style="list-style-type: none"> 1. Ten-Minutes build 2. Continuous integration. 3. Test-First programming 4. Incremental design 5. Pair programming 	<ol style="list-style-type: none"> 1. Sprint 2. Sprint review 	<ol style="list-style-type: none"> 1. Developing by Feature 2. Inspection 3. Regular Builds 4. Reporting / visibility of results 	<ol style="list-style-type: none"> 1. Developing by components 2. Software inspection 3. Project postmortem 	<ol style="list-style-type: none"> 1. Frequent product delivery 2. Iterative and incremental development 3. Agile interaction design testing 	<ol style="list-style-type: none"> 1. Revision and Review 2. Side-by-side programming 3. Agile interaction design
Customer collaboration over contract negotiation	<ol style="list-style-type: none"> 1. Weekly cycle 2. Quarterly cycle 	<ol style="list-style-type: none"> 1. Product backlog 2. Sprint planning meeting 	<ol style="list-style-type: none"> 1. Domain object modelling. 	<ol style="list-style-type: none"> 1. Adaptive management model 2. Joint application development 	<ol style="list-style-type: none"> 1. Collaboration and cooperation among stakeholders 2. Requirements are baselined at a high level. 	<ol style="list-style-type: none"> 1. Staging 2. Revisions and review 3. User viewing 4. Agile interaction design
Responding to change over following a plan	<ol style="list-style-type: none"> 1. Stories 2. Weekly cycle 3. Quarterly cycle 4. Slack 	<ol style="list-style-type: none"> 1. Sprint review 2. Sprint planning meeting 	<ol style="list-style-type: none"> 1. Domain object modelling 2. Configuration Management 	<ol style="list-style-type: none"> 1. Adaptive cycle planning 2. Customer focus group reviews 	<ol style="list-style-type: none"> 1. Reversible changes 	<ol style="list-style-type: none"> 1. Reflection Workshops 2. Methodology Tuning 3. Blitz planning 4. Delphi estimation 5. Monitoring of a progress 6. Burn charts
Keeping the process agile	-	<ol style="list-style-type: none"> 1. Sprint review 2. Daily scrum meeting 	<ol style="list-style-type: none"> 1. Reporting / visibility of results 2. Inspection 	<ol style="list-style-type: none"> 1. Software inspection 2. Project Postmortem 	<ol style="list-style-type: none"> 1. Integrated testing 	<ol style="list-style-type: none"> 1. Reflection workshops 2. Monitoring of a progress
Keeping the process cost effective	-	-	-	-	-	<ol style="list-style-type: none"> 1. Reflection workshops 2. Process miniature 3. Methodology shaping

References

- AgileManifesto (2001) Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>. 2001, Accessed 14 March 2005.
- Asan, E., & Bilgen, S. (2013). Agility problems in traditional systems engineering - a case study. *Complex Systems Design & Management*, 2013, 53–71.
- Beck, K., & Andres, C. (2004). *Extreme programming explained: Change*. Boston: Addison-Wesley.
- Berzins, V., Gray, M., & Naumann, D. (1986). Abstraction-based software development. *Communications of the ACM*, 29(5), 402–415.
- Beznosov, K., and Kruchten, P (2004) Towards agile security assurance, In Proceedings of the 2004 Workshop on New Security Paradigms (Nova Scotia, Canada, September 20–23, 2004). NSPW '04, ACM, New York, NY, pp. 47–54.
- Boehm, B. W. (1988). A spiral model of software development and enhancement, *Computer. IEEE JNL*, 21(5), 61–72.
- Boehm, B., & Turner, R. (2003). *Observations on balancing discipline and agility, proceedings of the agile development conference*. Salt Lake City, UT, USA: IEEE Computer Society.
- Boehm, B., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Boston: Pearson Education, Inc..
- Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools, *Inf. Software Technol.*, 38(4), 275–280.
- Brinkkemper, S., Saeki, M., & Harmsen, F. (1998). *Assembly techniques for method engineering, advanced information systems engineering, 10th international conference, CAiSE'98* (pp. 381–400). Italy: Pisa.
- Bucher, T. Klesse, M (2006) Contextual Method Engineering. University of St. Gallen (HSG). Report No.: BE HSG/EIW/03
- Chapman, J.R (1997) Software Development Methodology, http://www.hyperhot.com/pm_sdm.htm.
- Chance, K. (2011). *User stories in practice: A distributed cognition perspective*. MCIS Thesis: Auckland University of Technology.
- Conboy, K. (2009). Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329–354.
- Conboy, K and Fitzgerald, B (2010) Method and developer characteristics for effective agile method tailoring: a study of expert opinion, ACM Transactions on Software Engineering Methodology (TOSEM), Vol. 20, No 1.
- Cockburn, A. (2002). *Agile software development*. Boston: Addison-Wesley.
- Cockburn, A (2003). People and Methodologies in Software Development, PhD. Thesis, University of Oslo, http://alistair.cockburn.us/index.php/People_and_methodologies_in_software_development.
- Cockburn, A. (2005). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley.
- Cockburn, A. (2006). *Agile software development: The cooperative game*. Addison-Wesley.
- Conboy, K., & Fitzgerald, B. (2004). *Toward a conceptual framework of agile methods: A study of agility in different disciplines, proceedings of the 2004 ACM workshop on interdisciplinary software engineering research* (pp. 37–44). NY, USA, Newport Beach, CA, USA: ACM Press New York.
- Cohn, M., & Ford, D. (2003). Introducing an agile process to an organisation. *Computer*, 36(6), 74–78.
- Elssamadisy, A (2007) *Patterns of Agile Practice Adoption*, InfoQ, Lulu.com (publisher), 2007, pages 184.
- Feiler, P and Humphrey, W.S (1993) Software process development and enactment, International Conference on the Software Process Improvement, Berlin, Germany, 1993, pp. 28–40.
- Feuerlicht, G. (2006). *System development life-cycle support for service-oriented applications, new trend in software methodologies, tools and techniques (SoMeT2006)* (pp. 116–126). Quebec, Canada: IOS Press.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2), 200–213.
- FOI ACT (1982) Commonwealth Consolidated Acts: http://www.austlii.edu.au/au/legis/cth/consol_act/foia1982222/
- Fowler, M (2003) The New Methodology, <http://www.martinfowler.com/articles/newMethodology.html>.
- Gains, M., & Hawkins, T. (2007). Keeping the peace: mixing agile and waterfall methods. *Cutter IT Journal*, pp., 25–29.
- Harrison, R. (2011) TOGAF foundation. The Open Group.
- Harmsen, A. F. (1997). *Situational method engineering*. Doctoral dissertation. University of Twente.
- Harmsen, F., I. Lubbers, G. Wijers. (1995) Success-driven Selection of Fragments for Situational Methods: The S3 model. In: Pohl, K., and P. Peters (Eds.), Proceedings of the Second International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ'95), Aachener Beiträge zur Informatik, Band 13, pp. 104–115, Aachen, 1995.
- Henderson-Sellers, B. (2003). Method engineering for OO systems development. *Communications of the ACM*, 46(10), 73–78.
- Henderson-Sellers, B., & Serour, M. K. (2005). Creating a dual-agility method: the value of method engineering. *Journal of Database Management*, 16.4(2005), 1.
- Henderson-Sellers, B., & Gonzalez-Perez, C. (2006). "on the ease of extending a powertype-based methodology metamodel." *meta-modelling and ontologies. WoMM, 2006*(2006), 11–25.
- Henderson-Sellers, B., & Ralyté, J. (2010). Situational Method Engineering: State-of-the-Art Review. J. UCS, 16(3), 424–478. submitted 31.3.09 revised version submitted 12.3.10
- Henderson-Sellers, B., Ralyté, J., & Rossi, M. (2014). *Situational method engineering*. Springer.
- Gill, A. Q. (2014). Hybrid adaptive software development capability: an empirical study. *Journal of Software*, 9(10), 2614–2621.
- Glaser, B. G. (1998). *Doing grounded theory: Issues and discussions*. Mill Valley, CA: Sociology Press.
- Highsmith, J. A. I. (2000). *Adaptive software development: A collaborative approach to managing complex systems*. New York: Dorset House Publishing.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- ISO/IEC (2007) Software Engineering - Metamodel for Development Methodologies (SEMDM). ISO/IEC Standard 24744.: 2007, ISO, Geneva
- ISO/IEC (2014) Software Engineering - Metamodel for Development Methodologies (SEMDM). ISO/IEC Standard 24744.: 2014, ISO, Geneva
- ISO/IEC/IEEE 42010 (2011) Systems and software engineering — Architecture description. <http://www.iso-architecture.org/ieee-1471/faq.html#wharch>.
- IIBA (2009) A guide to the business analysis body of knowledge, BABOK Version 2.0.
- Iivari, J., Hirschheim, R., & Klein, H. K. (2000). A dynamic framework for classifying information system development methodologies and approaches. *JMIS*, 2000, 17(3), 179–218.
- Järvinen, P. (2001). *On research methods*. Tampere: Juvenes-Print.
- Janssen, M., van der Voort, H., and van Veenstra, A.F. (2015) Failure of large transformation projects from the viewpoint of complex adaptive systems: management principles for dealing with project dynamics. *Information Systems Frontiers*, Volume 17, Issue 1, pp 15–29

- Kasanen, E., Lukka, K., & Siitonen, A. (1993). The constructive approach in management accounting research. *Journal of Management Accounting Research*, 243–264.
- Kenens, P., Michels, S., Matthijs, F., Robben, B., Truyen, E., Vanhaute, B., Wouter, J., & Verbaeten, P. (1998). *An AOP case with static and dynamic aspects, European conference on object-oriented programming (ECOOP'1998)*. Belgium: Brussels.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2nd Edition, Wiley.
- Knolmayer, G., Endl, R., & Pfahrer, M. (2000). Modelling processes and workflows by business rules. *Business Process Management, LNCS, 1806*, 16–29.
- Kornyshova, E., Rébecca Deneckère, R., and Claudepierre, B. (2011). Towards Method Component Contextualization. *International Journal of Information System Modeling and Design (IJISMD) 2*, 4 (2011) 49–81"
- Kumar, K., & Welke, R. J. (1992). *Methodology engineering: A proposal for situation-specific methodology construction, in Challenges and Strategies for Research in Systems Development, J* (pp. 257–269). Chichester: Wiley.
- Laanti, M. (2008). *Implementing program model with agile principles in a large software development organisation* (pp. 1383–1391). Turku, Finland: IEEE COMPSAC.
- Laanti, M. (2013). *Agile Methods in Large-Scale Software Development Organizations. Applicability and model for adoption*. Diss. Dissertation. University of Oulu, 2013.
- Lee, S. and Young, H. (2013) Agile Software Development Framework in a Small Project Environment, *J Inf Process Syst*, Vol.9, No.1.
- Leite, J. C., & Leonardi, M. C. (1998). Business rules as organisational policies, proceedings of the 9th international workshop on software specification and design. *IEEE Computer Society, Washington, DC, 1998*, 68.
- Leonardi, M.C. and Leite, J.C. (2002) Using business rules in extreme requirements, 14th international conference, CAiSE, Toronto, Canada, 2002, pp.420–435.
- Mahanti, A. (2006) Challenges in Enterprise adoption of agile methods – a survey. *Journal of Computing and Information Technology - CIT*, pp. 197–206.
- Maharmeh, M. and Unhelkar, B. (2008) Investigation into the creation and application of a composite application software development process framework (CASDPF), fifth international conference on Information technology: New generations, ITNG, IEEE, pp.1286.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266.
- Miller, J. G. (1995). *Living systems*. University Press of Colorado.
- Nasir, M. H., Ahmad, R., & Hassan, N. H. (2008). Resistance factors in the implementation of software process improvement project in Malaysia. *Journal of Computer Science*, 4(3), 211–219.
- Odell, J. (2002). Objects and agents compared. *Journal of Object Technology*, 1(1), 41–53.
- Ojala, L., & Hilmola, O. P. (2003). Case study research in logistics'. *Turku School of Economics and Business Administration, Series B/ 1*.
- Oriens, B., & Yang, J. (2006). A rule driven approach for developing adaptive service oriented business collaboration. *IEEE International Conference on Service Computing, Chicago, USA, 2006*, 182–189.
- Osterle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., and Sinz, E.J. (2010) Memorandum on design-oriented information systems research. *European Journal of Information Systems*, 20(1), pp. 7–10.
- Palmer, S. R., & Felsing, J. M. (2002). *A practical guide to feature-driven development*. Upper Saddle River: Prentice-Hall Inc.
- Peffer, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hiu, W., Virtanen, V., & Bragge, J. (2006). *The design science research process: A model for producing and presenting information systems research*. Claremont, CA, USA: DESRIST.
- Qumer, A., & Henderson-Sellers, B. (2008a). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4), 280–295.
- Qumer, A., & Henderson-Sellers, B. (2008b). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11), 1899–1919.
- Qumer, A., & Henderson-Sellers, B. (2009). Agile software solution framework: An analysis of practitioners' perspectives. In *Information Systems: Modeling, Development, and Integration* (pp. 41–52). Springer Berlin Heidelberg.
- Qumer, A., & Henderson-Sellers, B. (2010). *Framework as software service (fass) an agile e-toolkit to support agile method tailoring*. Athens, Greece: CSOFT.
- Racheva, Z., and Daneva, M., and Sikkil, K. (2009) Value creation by agile projects: methodology or mystery? *PROFES*, pp. 141–155.
- Ralyté, J. (1999). *Reusing scenario based approaches in requirement engineering methods: CREWS method base, proceedings of the 10th international workshop on database and expert systems applications (DEXA'99)* (pp. 305–309). Italy, IEEE Computer Society, Los Alamitos, CA, USA: Florence.
- Randell, B., & Zurcher, F. W. (1968). *Iterative multi-level modelling: A methodology for computer system design* (pp. 867–871). IFIP: IEEE CS Press.
- Reifer, D. (2002). How good are agile methods? *IEEE Software*, 19, 16.
- Reis, C. A. L., Reis, R. Q., Schlebbe, H., & Nunes, D. J. (2002). *Resource instantiation policies for software process environments, 26th annual international computer software and applications conference* (pp. 53–58). Oxford: England.
- Rolland, C., Prakash, N. (1996) A Proposal for Context-Specific Method Engineering, in: Brinkkemper, S., Lytinen, K., Welke, R.J. (Hrsg.), *Method Engineering, Principles of Method Construction and Tool Support*, Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, Springer, Berlin pp. 191–207.
- Royce, W.W. (1970) Managing the Development of Large Software Systems: Concepts and Techniques. In *WESCON Technical Papers, Western Electronic Show and Convention, Los Angeles, Aug. 25–28, number 14*. Reprinted in proceedings of the ninth international conference on software engineering, Pittsburgh, ACM Press, PA, USA, 1989, pp. 328–338.
- Rodríguez, P., Markkula, J., Oivo, M., & Turula, K. (2012). Survey on agile and lean usage in finnish software industry. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 139–148). ACM.
- Schwaber, K. (2007). *The Enterprise and scrum*. Washington: Microsoft Press.
- Smith, G., & Sidky, A. (2009). *Becoming agile in an imperfect world*. Manning: Publications.
- Sommer, A.F., Dukovska-Popovska, I., Steger-Jensen, K. (2014). *Agile Product Development Governance – On Governing the Emerging Scrum/Stage-Gate Hybrids*. 2014. IFIP Advances in Information and Communication Technology pp 184–191.
- Stapleton, J. (1997) 'DSDM: The method in practice', Addison-Wesley, Inc
- Theodorakis, M., Analyti, A., Constantopoulos, P., & Spyrtatos, N. (1999) Contextualization as an Abstraction Mechanism for Conceptual Modelling, *International conference on conceptual modeling, Paris, France, Springer Berlin / Heidelberg*, vol. 1728.

- Tolfo, C., & Wazlawick, R. S. (2008). The influence of organisational culture on the adoption of extreme programming. *Journal of Systems and Software*, 81(11), 1955–1967.
- Tyndale-Biscoe, S., Sims, O., Wood, B., & Sluman, C. (2002). *Business modelling for component systems with UML, in proceedings of the sixth international enterprise distributed object computing conference*. Washington, USA: IEEE Computer Society.
- Vähäniitty, J., & Rautiainen, K. T. (2008). Towards a conceptual framework and tool support for linking long-term product and business planning with agile software development. In *Proceedings of the 1st international workshop on software development governance* (pp. 25–28). Leipzig: Germany.
- Vijayarathy, L.R., and Turk, D (2008) Agile software development: a survey of early adopters. *Journal of Information Technology Management* Volume XIX, Number 2.
- Wallin, C., Ekdahl, F., & Larsson, S. (2002). Integrating business and software development models. *IEEE Software*, 19(6), 28–33.
- Wong, S. P., & Whitman, L. (1999). *Attaining agility at the Enterprise level, proceedings of the 4th annual international conference on industrial engineering theory*. San Antonio, Texas, USA: Applications and Practice.
- Yin, R. K. (2003). *Case study research: Design and methods*. Thousand Oaks, CA, USA: SAGE Publications.
- Dr. Asif Q. Gill** is a senior lecturer in the Faculty of Engineering and Information Technology, School of Software at the University of Technology, Sydney. He has substantial experience in both industry and academia. He practices and investigates how to design and implement adaptive strategy and architecture for continuous efficiency and innovation. His work focuses on “adaptive thinking” that includes agility, analytics, design thinking, resiliency, service science and systems thinking.
- Brian Henderson-Sellers** received the doctor of science (DSc) degree from the University of London in 2001 for his research contributions in object-oriented methodologies. Until his retirement in 2015, has was the director of the Centre for Object Technology Applications and Research and a professor of information systems at the University of Technology, Sydney (UTS). He is the author or editor of 34 books and over 600 other publications.
- Dr. Mahmood Niazi** is working as an associate professor at King Fahd University of Petroleum and Minerals Saudi Arabia. Dr Niazi has 18 years of international experience, 8 years in software industry and 10 years in academia. Previously, he worked for Keele University UK, National ICT Australia, the University of Technology Sydney Australia, the University of Sydney Australia, the University of Manchester, UK. He holds a PhD from University of Technology Sydney Australia and MPhil from University of Manchester, UK.