New Trends in Software Methodologies, Tools and Techniques
H. Fujita and V. Mařík (Eds.)
IOS Press, 2009

doi:10.3233/978-1-60750-049-0-367

APM^3: A Methodology Metamodel for Agile Project Management

Mahsa Hasani Sadi[1] and Raman Ramsin
Department of Computer Engineering, Sharif University of Technology, Iran

Abstract. The advent of agile methodologies, though contributing much to software development processes, had a more profound impact on project management processes. Through supporting adaptability in their process frameworks, agile methodologies deviated from conventional project management approaches. This novel attitude has resulted in the emergence of an agile framework for project management. The Agile Project Management Framework (APMF) consists of fine-grained project management practices applied in agile methodologies, and is fast emerging as an alternative to the conventional project management framework. However, there are deficiencies in both frameworks that prevent developers from enhancing each framework to embrace the other. A logical solution is to merge the two frameworks into a Unified Project Management Framework (UPMF). With this objective in mind, we propose a project-management methodology metamodel as the common abstract substrate to fuse the standard framework with its agile counterpart. The proposed Agile Project Management Methodology Metamodel (APM^3) has been developed through applying abstraction to the fine-grained constituents of APMF. Based on the generic agile metamodel of APM^3, an analytical review of the project management processes of seven prominent agile methodologies has been conducted.

Keywords. project management methodology, unified project management framework, methodology metamodel, agile methodologies, process framework.

Introduction

The most significant characteristic of business environments is their exposure to permanent change. The volatility encountered in these environments results from changing business and organizational infrastructures such as resources, requirements, information, strategies and architectures. To survive in such an environment – where the only constant is change – the adoption of flexible and adaptable development processes is inevitable. In order to provide these features in their processes, agile methodologies incorporate evolutionary and iterative-incremental – *adaptive* – processes that support fast and frequent release of products.

The advent of agile methodologies contributed most to process and management frameworks, rather than to software development activities. By introducing adaptive process models, agile methodologies revolutionized the role of management framework activities in software development processes. Management framework activities, which were once considered secondary to development activities, became fundamental

[1] Mahsa Hasani Sadi: Sharif University of Technology, Department of Computer Engineering, Azadi Avenue, Tehran, Iran; E-mail: mhsadi@ce.sharif.edu.

components of the agile process model. This resulted in the extraction and compilation of agile process frameworks and activities from agile software development methodologies. Over time, agile project management activities have been consolidated into a de facto framework for agile project management. Although not formally defined, this framework is common knowledge among agile practitioners. For sake of brevity, we will call this framework APMF (Agile Project Management Framework).

Although the consolidation of fine-grained project management activities and practices into the APMF helps project managers and process engineers in applying the agile approach, the framework itself suffers from lack of coherence. This deficiency can be traced to the fact that abstraction has not been adequately applied to the fine-grained constituents of the APMF. A coherent agile project management framework should provide project managers and process engineers with reusable agile process components, along with guidelines for adapting/composing custom processes from those components. The need for such a framework is mainly due to the situation-dependent nature of project management practices and processes, which necessitates the existence of a comprehensive framework for engineering custom project management processes that are tailored to fit the project situation at hand. The scope of such a comprehensive framework goes far beyond agile project management, covering the entire field of project management – including conventional approaches.

On the other hand, since APMF is based on adaptive process models, it poses a great challenge to the conventional project management frameworks and standards that have so far been developed, especially the standard set by the Project Management Body of Knowledge (PMBOK) [1], [2]. These frameworks are based on linear process models, and therefore fail to support agile management frameworks; this and other shortcomings in current standard project management frameworks impede their enhancement to cover APMF. Moreover, due to the poor constitution of APMF in its current state – which only consists of fine-grained project management activities and some coarse-grained process frameworks – its promotion to a comprehensive framework is difficult. However, the maturity of APMF – which is a direct result of the maturity of the adaptive processes on which it is based – provides a promising prospect.

Considering the above problems, it seems that the logical way to obtain a unified basis for project management standards and frameworks is to fuse current standards with APMF. This is expected to result in the emergence of a *Unified Project Management Framework* (UPMF). UPMF will not only contribute to the establishment of a comprehensive project management framework, but will also lead to the production of fused project management processes that combine agile and conventional features. This may well contribute to resolving the shortcomings in project management processes, whether agile or conventional. However, in order to set the foundation for UPMF, several obstacles should be removed. The main obstacle hindering the development of UMPF is the lack of a common abstract substrate based on which the fused comprehensive framework can be built. Furthermore, the shortcomings of standard frameworks in supporting agility – not to mention the deficiencies of APMF itself – make fusion very problematic.

When considering the above issues, and the situation-dependent nature of project management processes, Situational Method Engineering (SME) approaches [3] come to mind as viable solutions to the fusion problem. We have therefore converged project management standards and APMF towards the definition of a project management methodology. To this aim, we have used method-engineering approaches to develop an

abstract substrate for project management activities, resulting in an outline for a *Project Management Methodology Metamodel* (PM^3). Based on the PM^3 thus produced, we provide an analytical review of standard project management frameworks and APMF. The main focus of our study is on enhancing APMF through applying abstraction to its fine-grained activities and practices. We thus outline an *Agile Project Management Methodology Metamodel* (APM^3) as an abstract agile project management framework based on PM^3. We then show how the resulting project management framework is manifest in seven prominent agile methodologies.

The rest of this paper is organized as follows: Section 1 provides a concise overview of related works; an abstract outline of PM^3 is devised and presented in Section 2; Section 3 contains a brief review of standard project management frameworks that has been conducted based on PM^3; in Section 4, we introduce APM^3 and provide an analytical review of seven prominent agile methodologies; and Section 5 contains the conclusions, as well as strategies for furthering this research.

1. Related Research

Most research on agile methodologies and their project management frameworks has been concentrated on extracting and compiling project management processes and related fine-grained activities from prominent agile methodologies. Such research efforts are mostly conducted in the context of the APMF [4], [5], [6], [7], [8]. APMF deviates from standard project management frameworks [1], [2], and therefore poses a great challenge to the field of project management. To address these challenges, much work has been concentrated on comparing APMF with standard project management frameworks, mainly to shape a rational background for both through depicting their similarities and differences [9], [10], [11], [12], [13], [14]. However, comparison has been very difficult due to the lack of an abstract project management framework to provide a set of criteria for analyzing project management processes.

In this paper, as a solution to this problem, we propose PM^3 as a project management methodology metamodel. In order to obtain a common substrate, we use methodology-engineering approaches [3], thus regarding the target project management *framework* as a project management *methodology*. This results in a common attitude towards APMF and standard project management, regarding them as general processes that must be tailored to fit the project situation at hand. The idea of a project management methodology has been applied in previous research conducted on APMF and standard project management frameworks [15], [16], [17], [18]. Although our proposed approach in the development of PM^3 has been inspired by [17], [19], it also uses ideas from methodology metamodels – especially SPEM-2 [20] and OPF [21] – which are based on viewing processes as consisting of process components.

2. Project Management Methodology Metamodel (PM^3)

In compliance with the approach adopted in prominent software development methodology metamodels, especially SPEM-2 [20]and OPF [21], we sketch the outline of the Project Management Methodology Metamodel (PM^3) through decomposing

methodology processes into their constituent elements – as proposed in [17], [19]. Although project management processes have been addressed in OPF and SPEM-2, they are considered as umbrella activities that are secondary to software development activities; excessive attention to development activities has resulted in an incomplete set of project-management process chunks. We consider a project management framework as a project management methodology that is independent of the software development process used. By applying abstraction to project management processes, we extract the project management framework in the form of a methodology metamodel that contains the software development methodology as a sub-domain.

We highlight the high-level features of project management processes by decomposing project management processes into basic components. This results in the decomposition of conventional project management practices and APMF into common elements, thus providing the basis for a uniform analysis of both frameworks. To this aim, as the first step, we have decomposed a project environment into three domains: *Target Environment*, *Project Management Environment*, and *Development Environment*; each of the latter domains is a sub-domain of its predecessor. A sub-domain and its super-domain may overlap, in that some process components may reside in both. Each super-domain imposes constraints on its sub-domain, and the processes in each sub-domain should be configured, tailored and adapted to fit best into the super-domain environment, thus satisfying its constraints. Moreover, each sub-domain process is monitored and controlled by its immediate super-domain. Any change in the constraints and components of a super-domain can affect its sub-domain.

Based on the schema of PM³ (depicted in Figure 1), we define a Project Management Methodology (PMM) as a process consisting of project framework components that are organized to fit best to the project situation at hand. A process framework is used as the basis for constructing the PMM; the process framework specifies the overall strategy of organizing project components in a PMM against the Target Environment (and its constraints) to achieve the goals.

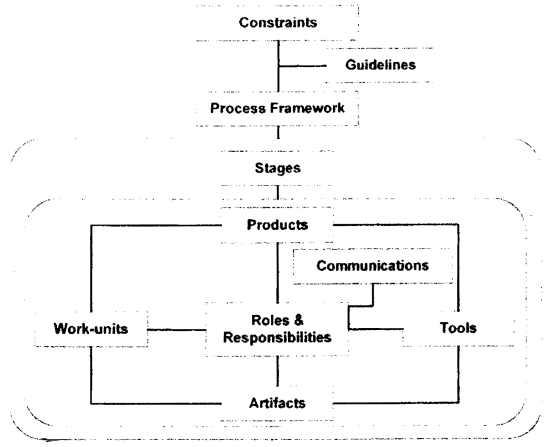

Figure 1. Components of Project Management Methodology Metamodel (PM³)

The process framework navigates the project framework components through the constraints imposed by the Target Environment, and maintains the focus on the goals that it should achieve. As each project typically has its own set of Target Environment, constraints, goals, and Development Environment, a tailored project management methodology is required that fits the project situation at hand. Guidelines should be provided as to how to select and assemble the best configuration of project framework

components. Thus, project management is not restricted to configuring, managing and controlling a project, but it also involves navigating the project so that it fits best into the Target Environment, considering the goals. This means that if one component changes, others should be modified accordingly in order to keep the project framework in line with the goals. Descriptions for PM^3 components have been provided in Table 1.

As PM^3 provides a generic outline for decomposition and categorization of project-management process elements, it can be used for reviewing project management processes. However, a detailed specification of PM^3 – providing a reusable set of project management processes together with specifications of the elements – remains an open issue that should be taken up in future studies.

Table 1. Descriptions for PM^3 Components

Process Framework	**Stages**	Duration or point in time that provides a high level organization to project management work units (e.g., lifecycle or milestone)
	Work-units	Any activity performed by people involved in projects at different levels (process models, patterns, activities, tasks, methods techniques, standards), which may result in the production of artifacts or products
	Roles & Responsibilities	Any active entity taking part in the project relating to the three domains specified at different levels of structural granularity (teams, individuals, managers, developers, stakeholders), and their related characteristics and skills
	Communications	Any structural, functional and behavioral relationship between roles and responsibilities
	Artifacts	Any project-management-related artifact and document produced during the project (e.g., plans, and vision document)
	Tools	Any CASE tool needed for performing work units, developing artifacts, and managing/monitoring roles and communications
	Products	Any deliverable of the project
Goals	The reasons based on which a project has been defined, and the outcomes that should be achieved (scope)	
Constraints	Limitations imposed by the target environment on the process framework (resources, costs, time, requirements, quality, risks, criticality, size, culture, technology); Constraints are either static or dynamic during the project lifecycle.	
Guidelines	Suggestions as to the best selection of components, and guidelines on how to organize them into processes	

3. PM^3-Based Review of Conventional Project Management Frameworks

In this section, standard project management frameworks, their structures and their shortcomings are scrutinized. Through applying PM^3, we provide an analytical review of PMBOK [1], [2], and its architecture in terms of reusable elements. PMBOK was chosen because it is the only comprehensive project management framework available.

PMBOK presents a process-centered analysis of project management tasks and activities taking place during the project lifecycle, and provides a functional model of the project management framework in an IPO (Input- Process- Output) format. It decomposes the activities and tasks taking place in the process of project management

into three categories of *Process Groups*, *Processes* and *Techniques*. *Process Groups* are coarse-grained activities that form the project management stages, including *initiating*, *planning*, *executing*, *monitoring and control*, and *closing*. Process Groups are composed of finer-grained activities called *processes*. Processes depict the standard activities necessary to accomplish a project management stage. Since there exist various ways of enacting a process, *Techniques* of applying a process are introduced as alternatives for performing the process. PMBOK also sketches the features of the *Tools* needed for applying project management processes and techniques. The overall interaction model of process groups during the project lifecycle is shown in Figure 2.

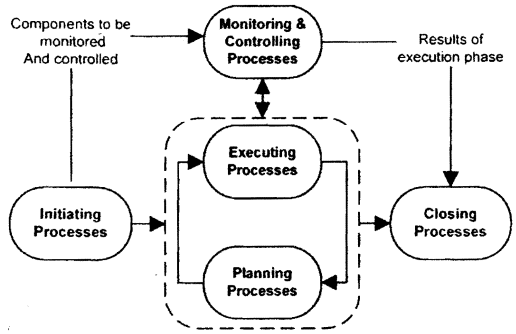

Figure 2. PMBOK Project management process model – adapted from [1], [2]

As shown in Figure 2, the overall project management process begins with *initiating* processes, continues with iterations of *planning* and *executing* processes that run in parallel with *monitoring and controlling* processes, and ends in *closing* processes. PMBOK also applies another orthogonal categorization on project management processes, called *knowledge areas*. PMBOK introduces nine knowledge areas: *Project Integration Management*, *Project Scope Management*, *Project Time Management*, *Project Cost Management*, *Project Quality Management*, *Project Human Resource Management*, *Project Communication Management*, *Project Risk Management*, and *Project Procurement Management*. Knowledge areas categorize processes based on the constraints imposed on the project management framework. The mapping between PMBOK and PM^3 components is shown in Table 2.

As revealed in Table 2, PMBOK suffers from several shortcomings in covering PM^3 components, some of which are explained below:

- PMBOK does not incorporate a process framework to depict the overall outline of its strategy for project management.
- Specification and configuration of communications (both functional and structural, among roles and responsibilities) has been excluded from project management components.
- As a framework introducing reusable method chunks, PMBOK does not provide any facilities or guidelines on selecting and assembling its components to configure and tailor a project management methodology that fits the Target Environment and its constraints.

Based on the above, and by focusing on the process framework, we abstract the overall strategy of PMBOK by organizing its components as a project management methodology based on the project's Target Environment and its constraints. Although PMBOK does not explicitly include a process framework, the organization of its processes points to a process framework. Applying abstraction on PMBOK's process groups and processes exposes a linear process framework (Figure 3).

Table 2. Mapping of PMBOK components to PM^3 components

PM^3 Project Management Components	PMBOK Project Management Components
Process framework	----------
Stages (life cycle, milestones, phases)	project lifecycle, project management lifecycle, project phases.
Work-units (process models, patterns, activities, tasks, methods techniques, standards)	Process model, process groups, processes, techniques
Roles & Responsibilities (teams, individuals, managers, developers, stakeholders)	Defining the Project Manager as the central role
Communications (structural and functional)	----------
Artifacts (plans, schedules, visions)	Plans, schedules, vision documents
Tools	Outline of tools and models (no specifications)
Goals	Scope management knowledge area
Constraints (resources, costs, time, quality, requirements, criticality, size, culture, technology)	Knowledge areas; dynamicity in constraints is considered a risk, and is managed by the *risk management* knowledge area.
Guidelines	----------

The PMBOK process framework consists of five stages, as shown in Figure 3. In order to instantiate a project management methodology, all PM^3 components should pass through these stages. The interactions among the stages are detailed in Table 3. The Configuration Deviation Control Loop in the PMBOK process framework implies that the main goal of the PMBOK strategy is to keep the project on the defined track, preventing any deviations from the plan. Therefore, PMBOK results in project management processes that are referred to as *plan-driven* or *predictive*. In target environments with a high degree of volatility in goals and constraints, applying predictive project management methodologies does not produce the desired results.

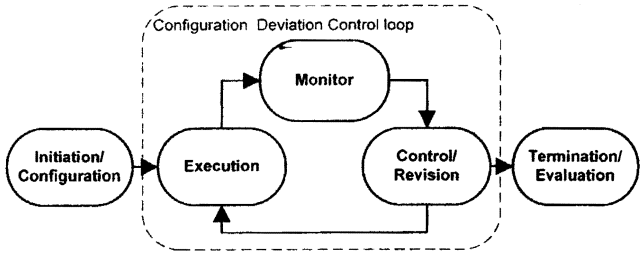

Figure 3. Linear Project Management Process framework of PMBOK

4. Agile Project Management Methodology Metamodel (APM^3)

Due to the dynamic nature of Project- and Target Environments in development processes – in which not only constraints and goals, but also project environment components may change continuously – the application of linear and predictive project management processes is not always effective. Agile methodologies, on the other hand, are based on an adaptive process framework that assumes dynamicity in constraints and goals.

Table 3. Stages of the linear project management process framework

Initiation/ Configuration		Goals and constraints imposed by the Target Environment are identified. Based on the constraints, the PM³ components (stages, work-units, roles and responsibilities, artifacts and tools) are configured and tailored to fit the target environment and constraints.
Configuration Deviation Control Loop	**Execution**	Project execution begins based on the configured components; the focus is on Development Environment activities.
	Monitoring	Execution and all Project Environment components are monitored in order to keep them in line with their initial configurations. The progress of the system of components is also monitored.
	Control/Revision	If any component deviates from its initial configuration, the process framework enters the control/revision stage in order to put everything back on track.
Termination/ Evaluation		If project goals are achieved completely, or the progress of the configured framework is not acceptable, the process framework enters this stage. In this stage, all project processes are closed and components are released. Moreover, all the outputs and components of the project are evaluated, necessary post-mortem analysis is conducted, and lessons learnt during the project are documented to be used in future projects.

By having an adaptive process framework in the background, and through paying special attention to Roles-and-Responsibilities and Communication components, agile methodologies can instantiate concrete processes incorporating fine-grained agile practices and principles. However, the lack of a generic and coherent project management paradigm and template results in ambiguities that hinder the introduction of agility in project management processes. Although agile project management practices have been extracted and compiled as the APMF, the framework is confined to activities and practices applied in agile methodologies. Examples include project management activities and practices such as *self-organizing teams* in XP, *daily stand-up meetings* in Scrum, *quality reviews* in ASD; and *reflection workshops* in Crystal.

To resolve this issue, we propose the Agile Project Management Methodology Metamodel (APM³). APM³ defines a generic coherent template for APMF based on PM³, developed through applying abstraction to agile processes. We use APM³ for reviewing seven prominent agile methodologies: DSDM, Scrum, XP, ASD, Crystal Clear, FDD [22], and AUP [23] are analyzed as concrete instances of APM³.

4.1. Agile Project Management Methodology Metamodel (APM³)

In this section, based on PM³ components and inspired by [4], we introduce the APM³ process framework component, which specifies the overall approach of agile project management in organizing project framework components, and governs the organization of other components in a project management process framework.

The agile process framework deviates from the linear process framework in that it uses an Adaptation/Learning Loop instead of the traditional Configuration Deviation Control Loop. Based on the assumption that the constraints of the Target Environment and the goals of the project framework are volatile, APM³ adds three stages to the linear process framework – *Initializing/Tuning*, *Finalization/Integration*, and *Evaluation/ Revision* – thus forming an Adaptation/Learning Loop (Figure 4).

Instead of scrutinizing the Target Environment obsessively in order to predict the path of the project, planning the overall project, and monitoring and controlling the project through its execution, APM³ uses incremental learning (as described in Table 4).

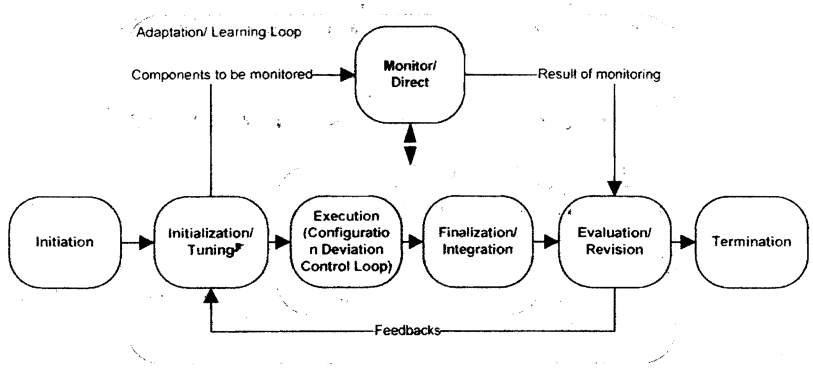

Figure 4. Agile Project Management Methodology Metamodel

As explained above, APM^3 models project management processes as learning agents in search of dynamic goals in a dynamic environment. This is in contrast to the approach of linear project management, which models project management processes as rigid and fixed solutions in which change is considered a liability that should be controlled and prevented. Based on this view, the outline schema of APM^3 components and interactions has been illustrated in Figure 5. Project environment components pass through the filters of the Adaptation/Learning loop continuously, thus being refined and revised in order to better fit the constraints imposed by the Target Environment. APM^3 depicts the outline of a generic metamodel, and should therefore be instantiated with detailed task, activities and management patterns in order to form a comprehensive template; this demands further research.

Table 4. Stages of the Agile Project Management Methodology Metamodel

Initiation	The project begins by envisioning outlines of the environment, constraints and goals.	
Adaptation/ Learning Loop	**Initialization/ Tuning**	Project environment components are organized/ reorganized depending on the outcome of the previous stage. If the previous stage is Initiation, components are organized with an initial configuration based on the outlines elicited. If the previous stage is Evaluation/Revision, component organization will be revised and tailored based on the feedbacks obtained.
	Execution (Configuration Deviation Loop)	The development process begins based on the configuration produced in the previous stage. The Configuration Deviation Loop will be performed as described in the linear process framework. Parallel instances of this stage can be concurrently performed.
	Finalization/ Integration	Outputs of the execution stage are refined, integrated with other increments, and released into the Target Environment.
	Monitor/Direct	This stage would be run in parallel with Execution and Finalization/Integration stages, mainly to direct these two stages towards the goals specified, monitor project framework components, and control their configuration.
	Evaluation/ Revision	The results of the Execution and Finalization stages, and the project environment framework and project progress are evaluated according to the results of the Monitor/Direct stage. Necessary revisions are made in order to tune the project to better fit the Target Environment. If evaluation results show that project goals are achieved completely, or that the advancement of the configured framework is not possible, the Termination Stage begins. This stage can proceed in parallel with the Finalization/Integration and Initializing/Tuning stages.
Termination	This stage is the same as its equivalent in the linear process framework (Section 3).	

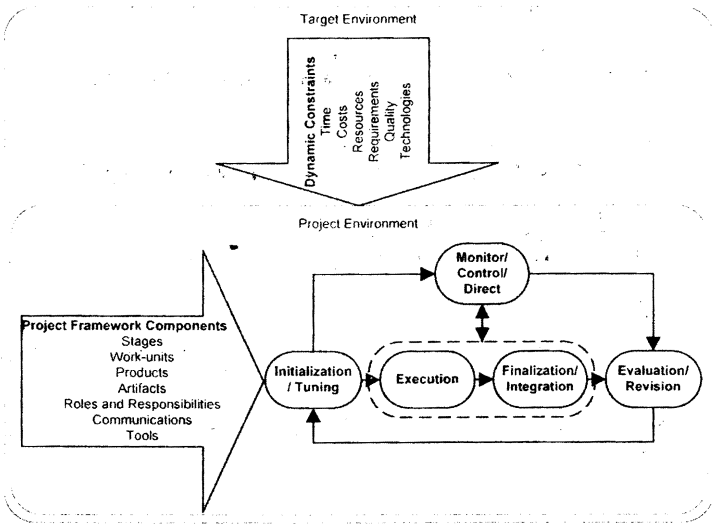

Figure 5. Schema of APM³ components and interactions

4.2. APM³-Based Review of Project Management in Prominent Agile Methodologies

Based on APM³, components of the project framework (including stages, work-units, products, artifacts, roles and responsibilities, communications and tools) are adapted in order to accommodate the dynamicity of the Target Environment constraints (including time, costs, resources, requirements, quality, technology and criticality). APM³ can also act as a benchmark that defines the maximum degree of agility in project management, based on which an Agile Process Maturity Model (APMM) can be defined. To this aim, we have categorized the seven agile methodologies under study based on their level of conformance to APM³. Four categories have thus been defined: APMM *level 0 (linear)*, *level 1*, *level 2, and level 3*, each providing a measure of the level of agility present in the methodology's project management process. We provide a review of the agile methodologies belonging to each category. By using the approach applied in [24], we have presented the significant APM³–related features that recur in the methodologies as *project-management process patterns*, presented at the *stage*, *task* and *technique* levels:

- **APMM level 0** (FDD, AUP) – FDD and AUP incorporate agility into their project management frameworks by adding iterations and parallelism to the execution stage of the linear process framework. The Configuration Deviation Control Loop is performed through configuring the relevant project framework components at the beginning of the project. Augmented FDD and AUP also cover the Finalization/Integration stage of the APM³. The relevant process patterns include task process patterns such as *feature-based planning*.
- **APMM level 1** (DSDM and ASD) – Processes at this level show the characteristics of the previous level; in addition, they apply the APM³ process framework for organizing their components into a project framework. In fact, by assuming dynamicity in constraints/quality/requirements in DSDM, and quality/requirements/time in ASD, they adopt the adaptation/learning loop of the APM³ and adapt it as project framework components (stages, products, and plan/schedule artifacts). This forces the project framework to tune the components in order to attain a better fit with the Target Environment and its constraints. Therefore, these methodologies incorporate instances of tuning/review process patterns. These include stage process patterns such as *quality reviews*, and task process patterns such as *reviewing and validating*.

- **APMM level 2** (XP, Scrum) – Processes at this level enhance the adoption of APM³ in their processes by adding two components to their project frameworks: roles-and-responsibilities, and communications. The relevant process patterns include task process patterns such as *self-organizing teams*.
- **APMM level 3** (Crystal Clear) – Processes at this level provide a high-level *flexible* project management process through reviewing and revising the process itself. In addition to other components, work-units are tuned continuously to fit the Target Environment. The relevant process patterns include task process patterns such as *reflection workshops*.

The different categories introduced, their coverage of APM³, and the relevant methodologies have been summarized in Table 5.

Table 5. APM³–based Comparison of Agile Methodologies

		Project Framework Components coverage by APM³ process framework ✓: follows APM³ metamodel ×:follows linear metamodel							Constraints F:fixed, A: adaptive, N: not addressed					
APMM Level	Methodology	stages	work-units	products	artifacts	roles and responsibilities	communications	tools	time	costs	resources	requirements	quality	Technology
Level 0	FDD	×	×	×	×	×	×	×	F	F	F	A	A	N
	AUP	×	×	×	×	×	×	×	F	F	F	A	A	N
Level 1	DSDM	✓	×	✓	✓	×	×	×	F	A	F	A	A	N
	ASD	✓	×	✓	✓	×	×	×	A	A	A	A	A	N
Level 2	XP	✓	×	✓	✓	✓	✓	×	A	A	A	A	A	N
	Scrum	✓	×	✓	✓	✓	✓	×	A	A	A	A	A	N
Level 3	Crystal Clear	✓	✓	✓	✓	✓	✓	×	A	F	F	A	A	N

5. Conclusions and Future Work

In the course of developing the APM³, several secondary research contributions were achieved. As the first step to developing the APM³, we introduced the Project Management Methodology Metamodel (PM³), which decomposes project management processes into a set of reusable components. PM³ also provides a set of analysis criteria, based on which we have reviewed the standard project management framework (as delineated by PMBOK) to point out its differences with APMF. Since each framework puts emphasis on different project management components, the idea of fusing these frameworks into a Unified Project Management Framework (UPMF) was proposed. The agile features of APM³, and its emphasis on roles and responsibilities and their communications, can be combined with detailed work-units and artifacts provided by PMBOK to thereby provide a comprehensive project management framework.

This research can be furthered in several aspects. By focusing on PM^3 and producing detailed specifications for the project management components outlined herein, a comprehensive project management methodology metamodel can be produced. The application of PM^3 to project management processes can produce a rich repository of reusable method chunks based on which project management methodologies can be built, leading to the development of CAME (Computer Aided Method Engineering) Tools [25] for assembly-based engineering of project management methodologies [3].

References

[1] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, Fourth Edition Exposure Draft, PMI, 2008.

[2] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, Third Edition, PMI, 2004.

[3] J. Ralyté, S. Brinkkemper, B. Henderson-Sellers, (Eds.), *Situational Method Engineering: Fundamentals and Experiences*, Springer, 2007.

[4] R. Owen, L J. Koskela, An Agile Step Forward in Project Management, *Proceedings of ASCE-CIB Leadership in Construction & Engineering Conference*, (2006), 216–224.

[5] S. Augustine, B. Payne, F. Sencindiver, S. Woodcock, Agile Project Management: Steering from the Edges, *Communications- ACM*, 48(12), (2005), 85–90.

[6] M. Coram, S. Bohner, The Impact of Agile Methods on Software Project Management, *Proceedings of ECBS'05*, 2005, 363–370.

[7] K. Schwaber, *Agile project management with Scrum*, Microsoft Press Redmond, USA, 2004.

[8] J. Sutherland, Future of Scrum: Parallel Pipelining of Sprints in Complex Projects, *Proceedings of AGILE'05*, (2005), 90–102.

[9] A. C. Marcal, F. S. F. Soares, A. D. Belchior, Mapping CMMI Project Management Process Areas to SCRUM Practices, *Proceedings of SEW'07*, (2007), 13–22.

[10] X. Wang, R. Vidgen, Order and Chaos in Software Development: A comparison of two software development teams in a major IT company, *Proceedings of ECIS'07*, (2007), 807–818.

[11] A. Mahanti, Challenges in enterprise adoption of agile methods - A survey, *Journal of Computing and Information Technology* **14**(3), (2006), 197–206.

[12] B. Boehm, R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*, Boston, MA: Addison-Wesley, 2004.

[13] S. Kopensteiner, How Agile Software Methodologies Influence The role of the Project Manager, *Proceedings of IPMA'04*, (2004).

[14] P. Abrahamsson, J. Warsta, M.T. Siponen & J. Ronkainen, New Directions on Agile Methods: A Comparative Analysis. *Proceedings of ICSE'03*, (2003), 244–254.

[15] R. Kliem, I. Ludin, K. Robertson, *Project management methodology: a practical guide for the next millennium*, 1997.

[16] C. Bentley, *Practical Prince2*, Stationary Office, 2005.

[17] C. N. Bodea, Agile Software Project Management Methodologies, *Economy Informatics*, (2005), 1–4.

[18] H. V. Loon, A Management Methodology to Reduce Risk and Improve Quality, *IEEE IT Professional*, **9** (2007), 30–35.

[19] A. Cockburn, Selecting a Project's Methodology, *IEEE Software*, **17**(4), 64–71.

[20] OMG, Software and Systems Process Engineering Metamodel Specification (v2.0). Object Management Group (OMG), 2007.

[21] D. Firesmith, B. Henderson-Sellers, *The OPEN Process Framework: An Introduction*, Addison-Wesley, 2001.

[22] R. Ramsin, R. F Paige, Process-Centered Review of Object-Oriented Software Development Methodologies, *ACM Computing Surveys*, **40** (1), (2008), 89 pages.

[23] S. W. Ambler, Agile Unified Process, published on the web at: www.ambysoft.com/unifiedprocess/agileUP.html (2006).

[24] S. W. Ambler, *Process Patterns: Building Large-Scale Systems Using Object Technology*, Cambridge University Press, 1998.

[25] A. Niknafs, R. Ramsin, Computer-Aided Method Engineering: An analysis of existing environments, *Proceedings of CAiSE'08*, (2008), 525–540.