FRAMEWORK AS SOFTWARE SERVICE (FASS)
An Agile e-Toolkit to Support Agile Method Tailoring

Asif Qumer and Brian Henderson-Sellers
Faculty of Engineering and IT, University of Technology, Boradway, Sydney, Australia

Keywords: Agile Method Tailoring, Method Engineering, Software as a Service.

Abstract: In a real software application development environment, a pre-defined or fixed methodology, whether plan-based or agile, is unlikely to be successfully adopted "off-the-shelf". Agile methods have recognised that a method should be tailored to each situation. The purpose of this paper is to present an agile e-toolkit software service to facilitate the tailoring of agile processes in the overall context of agile method adoption and improvement. The agile e-toolkit is a web-based tool to store and manage agile practices extracted from various agile methods and frameworks. The core component of the e-toolkit is the agile knowledge-base or repository. The agile knowledge-base contains agile process fragments. Agile consultants or teams can then use agile process fragments stored in the agile knowledge-base for the tailoring of situation-specific agile processes by using a situational method engineering approach. The e-toolkit software service has been implemented using a service-oriented cloud computing technology platform (Software as a Service – SaaS). The agile e-toolkit specifications and software application details have been summarized in this paper.

1 INTRODUCTION

It is well acknowledged that a pre-defined software development methodology, whether plan-based or agile, is unlikely to be able to be used or adopted off-the-shelf for any specific software project (Kumar and Welke, 1992; Lindvall and Rus, 2000; Keenan, 2004; Fitzgerald et al., 2006). It has been suggested (Mahanti, 2006) that "there is no agile methodology that can be universally applied and they all have to be tailored to integrate into existing processes". Cockburn (2002) suggests that "each situation calls for a different methodology". Keenan (2004) suggests that "projects differ in their scale, scope, and technical challenge, the same process will not suit all circumstances." All the agile methods (e.g. XP, Scrum, Crystal) have recognised this and suggest that a method should be tailored to each situation. Fitzgerald *et al.* (2006) suggested that "tailoring of methods is commonplace in the vast majority of software development projects and organizations. However, there is not much known about the tailoring and engineering of agile methods, or about how these methods can be used to complement each other". There are a number of conceptual frameworks or tools (e.g. Keenan 2004; Cockburn 2005; Sidky 2007) that discuss agile method tailoring. However, they lack the practical tool-based support for agile method tailoring. For instance, in order to support agile method tailoring, Keenan (2004) suggested a process knowledge base to hold the individual agile techniques but does not provide a full software tool-based support for agile method tailoring,

In our recent research, we have developed an agile software solution framework (Qumer and Henderson-Sellers 2008a,b) (ASSF – Figure 1) to support the assessment, tailoring, adoption and improvement of agile methods. One aspect of the ASSF is method tailoring. The ASSF includes a software component "framework as software service (FaSS)" that provides an agile e-toolkit service to store, tailor and manage agile processes and practices. The core component of the agile e-toolkit is an agile knowledge-base (repository), which contains practices or process fragments that had been identified and extracted from various agile methods. Here, it is anticipated that the FaSS can be used together with the well-known situational method engineering approach (e.g. Kumar and Welke, 1992; Brinkkemper, 1996; Cockburn, 2002a, b, 2005) to facilitate the tailoring of agile software development methods.

This paper is organized as follows: Section 2

Qumer A. and Henderson-Sellers B. (2010).
FRAMEWORK AS SOFTWARE SERVICE (FASS) - An Agile e-Toolkit to Support Agile Method Tailoring.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 167-172
DOI: 10.5220/0003006401670172

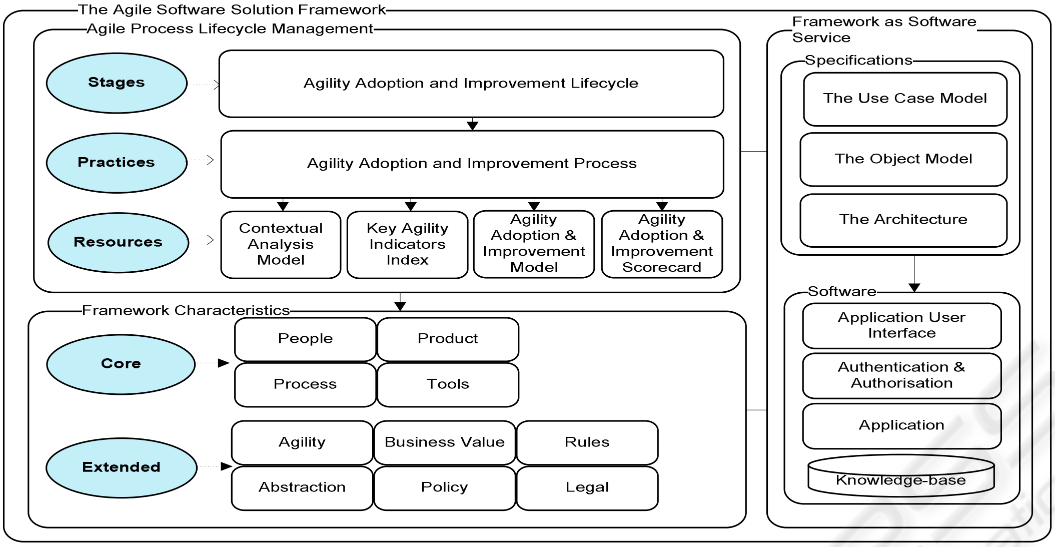
The Agile Software Solution Framework
Agile Process Lifecycle Management
Stages
Practices
Resources
Agility Adoption and Improvement Lifecycle
Agility Adoption and Improvement Process
Contextual Analysis Model
Key Agility Indicators Index
Agility Adoption & Improvement Model
Agility Adoption & Improvement Scorecard
Framework Characteristics
Core
Extended
People
Product
Process
Tools
Agility
Business Value
Rules
Abstraction
Policy
Legal
Framework as Software Service
Specifications
The Use Case Model
The Object Model
The Architecture
Software
Application User Interface
Authentication & Authorisation
Application
Knowledge-base

Figure 1: The Agile Software Solution Framework.

provides an overview of the agile software solution framework. Section 3 presents the agile e-toolkit Use Case Model. Section 4 presents the agile knowledge-base or repository object model. Section 5 presents the agile e-toolkit architecture and software application, respectively. Finally, Section 6 presents the conclusions.

2 AGILE SOFTWARE SOLUTION FRAMEWORK (ASSF)

The ASSF (Figure 1) has three main parts: framework characteristics (FCs), agile process lifecycle management (APLM) and framework as software service (FaSS).

2.1 Framework Characteristics (FCs)

The framework characteristics includes a set of key elements or attributes (Figure 1) that could be related in an agile or hybrid software development methodology (SDM). The framework characteristics set can be classified in two categories: core and extended. The core characteristics are: people, process, product and tools; whereas the extended characteristics are agility, abstraction, business value, policy, rules and legal. These framework characteristics have been initially identified based on the analysis of both agile and non-agile extant approaches as well as meta-models (e.g. Agile Manifesto, 2001; Firesmith and Henderson-Sellers, 2002; Australian Standards, 2004; ISO/IEC, 2007). Further feedback from industry as well as from researchers on these characteristics have been obtained with the aim of eliciting, by means of a field survey, the most relevant and important SDM characteristics for practitioners (Qumer and Henderson-Sellers, 2009). An agile or hybrid SDM can be built and then tailored on these framework characteristics for a particular situation when using a situational method engineering approach (Firesmith and Henderson-Sellers, 2002).

2.2 Agile Process Lifecycle Management (APLM)

This section presents the second part of the ASSF (Qumer *et al.*, 2007; Qumer and Henderson-Sellers, 2006, 2008a, b) framework (Figure 1), which contains six key components to support the adoption and improvement of agility. Firstly, this presents a generic high-level agility assessment, adoption and improvement lifecycle (AAIL): (1) initiation, (2) development, (3) deployment, (4) administration, (5) management and (6) governance. Secondly, it specifies the agility adoption and improvement process (AAIP), which specifies a step-by-step process or practices at each stage of the AAIL. The AAIP provides a set of twenty-two key practices to assist agile consultants, coaches and organizations in the overall context of agility adoption and improvement. Thirdly, it presents the essential models and templates that can be used at the different stages of the AAIL during the execution of the AAIP: contextual analysis model (CAM), key agility indicators index (KAII), agility adoption and improvement model (AAIM) and adoption and improvement scorecard. The CAM can help in understanding the current agility adoption and improvement capability and the readiness of the organization – a basic prerequisite to the adoption of a new methodology/process. The KAII is a measurement index that can be used by agile consultants and coaches to assess the degree of agility of an individual practice and the agile level of an organization or team. The AAIM can be used as a roadmap for introducing agile practices into both agile and non-agile software development environments. The agility adoption and improvement scorecard (AAIS) can be used to capture and highlight the status and progress of agility adoption and improvement at each stage.

2.3 Framework as Software Service

The ASSF framework as software service includes an agile e-toolkit service, providing a "tool-based" assistant to store and manage agile practices for the tailoring of situation-specific agile processes. The following sections present the e-toolkit (Framework as Software Service – FaSS) – its specification and the details of its software application.

3 THE USE CASE MODEL

The two primary actors and five main or essential use cases (detailed in Table 1) of the e-toolkit (independent of other conceptual components of the ASSF framework) are shown in Figure 2.

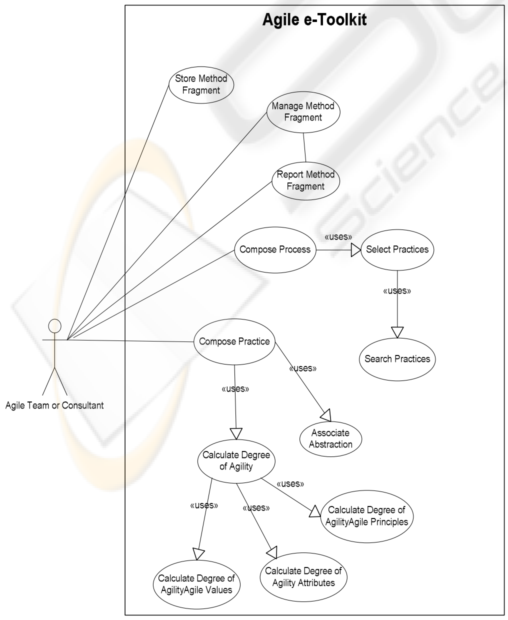

Figure 2: Agile e-Toolkit Essential Use Cases.

Table 1: Agile e-Toolkit Use Case Specification.

Actors & Use Cases	**Description**
Primary Actors	Users: Agile Team, Agile Consultant
UC01: Store Method Fragment	A user would be able to store the SDM fragments such as process, practices, agility (agile principles, agile values, agility attributes and agile levels), business value and abstraction.
UC02: Manage Method Fragment	A user would be able to add, delete, and update the SDM fragments such as process, practices, agility (agile principles, agile values, agility attributes and agile levels), business value and abstraction.
UC03: Report Method Fragment	A user would be able to perform search and list the stored SDM fragments such as process, practices, agility (agile principles, agile values, agility attributes and agile levels), business value and abstraction.
UC04: Compose Process	A user would be able to define a situation-specific process and relevant process areas (e.g. Development, Testing, and Requirements Engineering) within a process. A user would be able to search, select and then include agile practices in the relevant process area suiting to the specific situation-in hand (e.g. Process Composing or Tailoring Workshop). The included practices may be referred to as process line items.
UC05: Compose Practice	A user would be able to store an agile practice extracted from well known published agile methods and frameworks or customised agile practice (team's or consultant's experience-based best or customised practices). A user would be able to associate supported specific abstraction mechanisms (e.g. agent, service, object and neutral) to an agile practice. A user would be able to calculate degree of agility of an agile practice in terms of agility attributes, agile values and principles.

The agile e-toolkit allows users to access the system functionality, via essential system Use Cases and web interface, over the internet.

4 THE AGILE KNOWLEDGE-BASE OBJECT MODEL

This section presents the basic agile knowledge-base metadata or object model of the agile e-toolkit (Figure 3). The current version of the agile knowledge-base metadata or object model has been built on the ASSF framework characteristics for storing and managing the key SDM fragments such

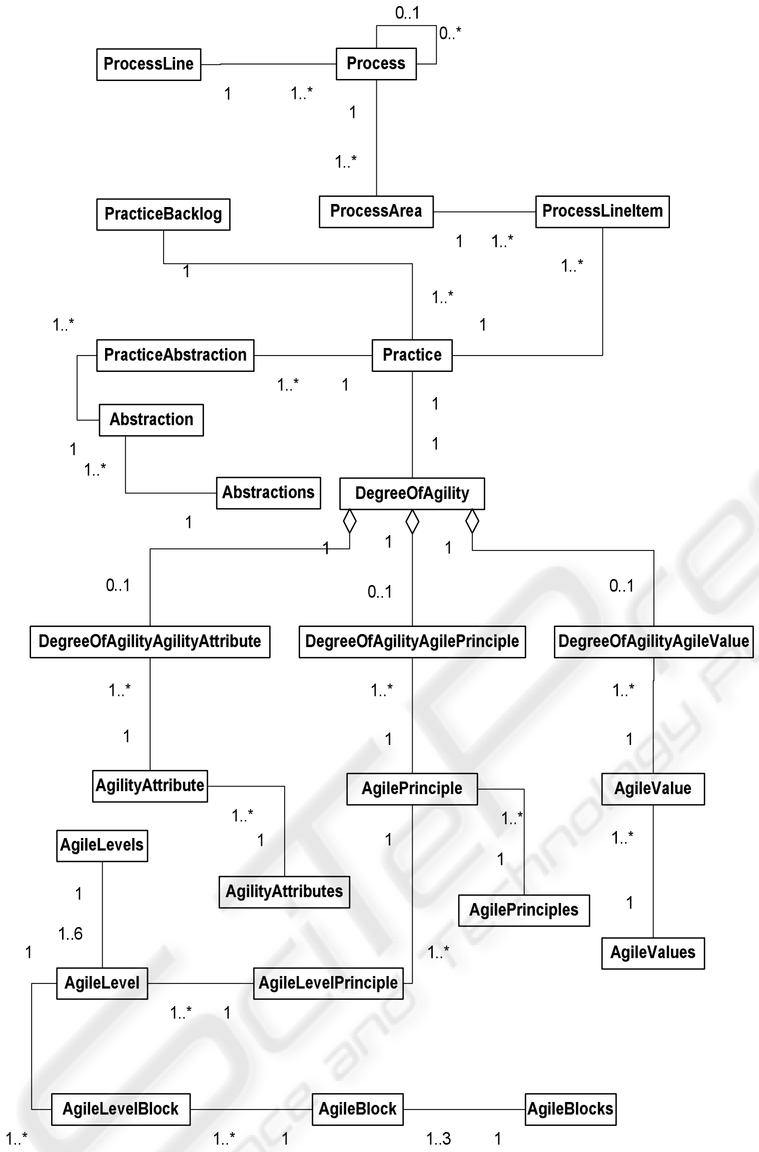

Figure 3: Agile e-Toolkit Object or Metadata Model.

as process, practices, agility (agile principles, agile values, agility attributes, agile levels and agile blocks) and abstraction, which are outlined below. Process Line: A list to store and manage processes in the agile knowledge-base of the e-toolkit.

Process: Process metadata to store and manage off-the-shelf or situational-specific process instance. A process can be linked to a parent process.

Process Area: Process area metadata to store and manage off-the-shelf or situational-specific set of selected agile practices for a team or organization.

Process Line Item: Process line item metadata to store and manage off-the-shelf or situation-specific selected agile practices.

Practice Backlog: A list to store and manage a set of agile practices (e.g. distilled from various agile methods and frameworks) in the agile knowledge-base of the e-toolkit. Practice metadata to store and manage an individual agile practice.

Practice Abstraction: To associate abstraction mechanisms to an individual agile practice.

Abstractions: A list to store and manage a set of SDM abstraction mechanisms in he agile knowledge-base of the e-toolkit. Abstraction metadata is to store and manage an individual abstraction mechanism.

Degree of Agility (DA): DA to calculate and store overall degree of agility of an agile practice in

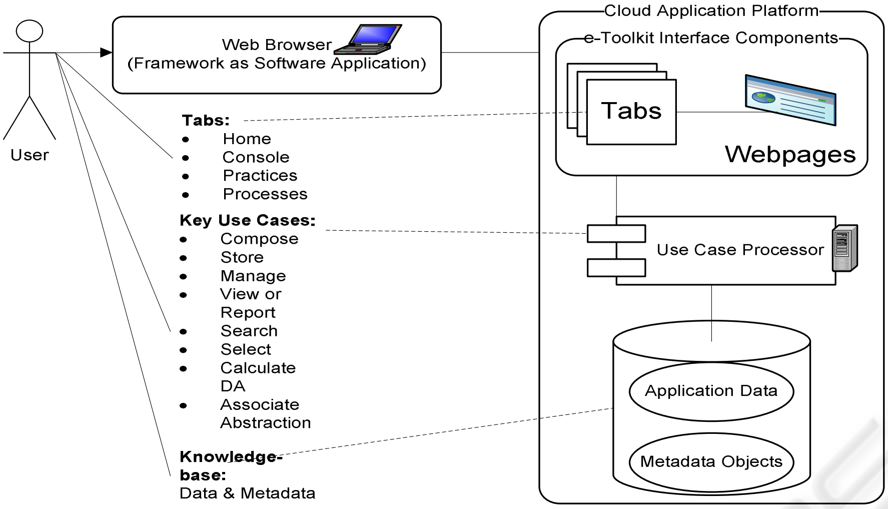

Figure 4: Agile e-Toolkit Architecture.

terms of key agility indicators: agility attributes, agile values and agile principles.

DA Agility Attributes are used to calculate and store the agility attributes (Key Agility Indicators) supported by an agile practice.

DA Agile Values are used to calculate and store the agile values (Key Agility Indicators) supported by an agile practice.

DA Agile Principles are used to calculate and store the agile principles (Key Agility Indicators) supported by an agile practice.

Agility Attributes: A list to store and manage a set of agility attributes in the agile knowledge-base of the e-toolkit. Agility Attribute metadata are used to store and manage an individual agility attribute.

Agile Values: A list to store and manage a set of agile values in the agile knowledge-base of the e-toolkit. Agile Value metadata are used to store and manage an individual agile value.

Agile Principles: A list to store and manage a set of agile principles in the agile knowledge-base of the e-toolkit. Agile Principle metadata are used to store and manage an individual agile principle.

Agile Level Principles: A list to store and manage a set of agile principles linked to the specific agile level.

Agile Levels: A list to store and manage a set of agile levels in the agile knowledge-base of the e-toolkit. Agile Level metadata are used to store and manage an individual agile level.

Agile Level Block: A list to store and manage a set of agile levels linked to the specific agile block (e.g. white, green, and black).

Agile Blocks: A list to store and manage a set of agile blocks in the agile knowledge-base of the e-toolkit. Agile Block metadata are used to store and manage an individual agile block.

5 AGILE E-TOOLKIT ARCHITECTURE AND SOFTWARE APPLICATION

The e-toolkit software service has been implemented by using the Force.com cloud computing application development, testing and deployment platform (Salesforce 2000). Figure 4 describes the architecture of this agile e-toolkit.

The Force.com cloud computing platforms allows the development, deployment and access of software applications over the internet as a service (SasS – Software as Service) (McGuire et al. 2008; SalesForce 2008). The e-toolkit runs in the internet cloud rather than as a desktop application running on the local machine or server. Agile teams and agile consultant can access e-toolkit service to search, select, compose and share agile processes and practices (e.g. during the agile process tailoring and adoption workshops) stored in the agile knowledge-base via internet cloud. The e-toolkit can be used to share the best practices within the team and across the organization. The e-toolkit service has four key components: user interface, use case processor, application data and metadata objects.

Tabs & WebPages: Home, Console, Practices and Processes to display e-toolkit functions (Use Cases) and contents of the knowledge-base.

Use Case Processor: Process the SDM's aspects or fragments stored in the knowledge-base via e-toolkit service Use Cases e.g. Store, Compose, Edit, Delete, Search, View, Calculate Degree of Agility

Application Data: Actual data stored in the knowledge bases – instances of the metadata objects

Metadata Object: The metadata objects to store and manage the application data.

6 CONCLUSIONS

The e-toolkit service is presented here as a software tool-based support that may prove useful during agile process tailoring and adoption workshops. Agile teams can inspect the contents of the e-toolkit agile knowledge-base (e.g. agile practices) and then may chose a set of agile practices in the context of the specific situation in hand – project or product development. The set of selected agile practices can then be configured into a process appropriate to the skills of the development team and the constraints of the target project and development environment, according to SME principles and practice, and then applied (adopted) in a real time by the team (enactment). In future, we are planning to do further comparative and empirical evaluation of the e-toolkit (in particular with non-agile process environments and also intending to develop other important services of the ASSF framework, such as additional, standard SME constraints such as criticality of the system to the business, capability (e.g. CMM, SPICE) of the organizational team members and quality criteria for the final product (e.g. Nguyen and Henderson-Sellers, 2003).

REFERENCES

AgileManifesto, 2001. Manifesto for Agile Software Development. http://www.agilemanifesto.org/

Australian Standards, 2004. Standard metamodel for software development methodologies. AS 4651-2004.

Brinkkemper, S., 1996. Method engineering: engineering of information systems development methods and tools. Inf. Software Technol., 38(4), 275-280.

Cockburn, A., 2002a. Agile Software Development. Addison-Wesley, Boston.

Cockburn, A., 2002b. Agile Software Development Joins the “Would-Be” Crowd. Cutter IT J., 15(1), 6-12.

Cockburn, A., 2005. Crystal Clear: a Human-Powered Methodology for Small Teams. Addison-Wesley.

Fitzgerald, B., Hartnett, G. and Conboy, K., 2006. Customising agile methods to software practices at Intel Shannon. J. Information Systems, 15, 200-213.

Firesmith, D. G. and Henderson-Sellers, B., 2002. *The OPEN Process Framework. Pearson Education*, UK.

ISO/IEC, 2007. Software Engineering - Metamodel for Development Methodologies. ISO/IEC 24744.

Keenan, F., 2004. Agile Process Tailoring and probLem analYsis (APTLY). In: *Procs. 26th Int. Conf. Software Eng*. (ICSE 2004), *IEEE, Edinburgh*, Scotland , 42-44.

Kumar, K. and Welke, R. J., 1992. Method Engineering: A Proposal for Situation-specific Methodology Construction. Systems Analysis and Design: A Research Agenda, John Wiley and Sons.

Lindvall M. and Rus, I., 2000. Process Diversity in Software Development, *IEEE Software*, 17(4), 14-18.

Mahanti, A., 2006. Challenges in Enterprise Adoption of Agile Methods – A Survey. J. Computing and Information Technology - CIT, 197–206.

McGuire, C., Roth, C., Carroll, D. and Tran, N., 2008. Force Platform Fundamentals: An Introduction to Custom Application Development in Cloud, Salesforce.com.

Nguyen, V. P. and Henderson-Sellers, B., 2003, Towards automated support for method engineering with the OPEN Process Framework, Procs. Seventh IASTED Int. Conf. Software Eng.& Applications (ed. M.H Hamza), ACTA Press, Anaheim, CA, USA, 691-696

Qumer, A. and Henderson-Sellers, B., 2006. Measuring agility and adoptability of agile methods: A 4-Dimensional Analytical Tool. IADIS Int. Conf. Applied Computing 2006, IADIS Press.

Qumer, A., Henderson-Sellers, B., and McBride, T., 2007. Agile adoption and improvement model. In: *Procs. EMCIS* 2007.

Qumer, A., and Henderson-Sellers, B., 2008a. An evaluation of degree of agility in six agile methods and its applicability for method engineering. J. Systems and Software, 81, 1899-1999.

Qumer, A., and Henderson-Sellers, B., 2008b. A framework to support the evaluation, adoption, adoption and improvement of agile methods in practice. Inf. Software Technol., 50(4): 280-295.

Qumer, A., and Henderson-Sellers, B., 2009. Agile Software Solution Framework: An analysis of Practitioners’ Perspectives. *Procs. UNISCON* 2009. LNBIP 20, Springer-Verlag, Berlin, 41-52.

SalesForce, 2000. www.salesforce.com.

SalesForce, 2008. Agile development meets cloud computing for extraordinary results at salesforce.com.

Sidky, A. 2007. A structured approach to adopting agile practices: The agile adoption framework, Ph.D. Thesis, Virginia Polytechnic Institute & State Univ., USA.