A framework to assist in the assessment and tailoring of agile software development methods

Asif Qumer

A dissertation submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy in Computing Sciences

School of Software
Faculty of Engineering and Information Technology
University of Technology, Sydney
July 2010

CERTIFICATE OF AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text. I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Production Note:
Signature removed prior to publication.

Acknowledgements

I have been privileged to have Prof. Brian Henderson-Sellers and Prof. John Debenham as my supervisors. I would like to express my sincere gratitude to my supervisor Prof. Brian Henderson-Sellers for providing me an opportunity to work with him on this research project. His highly valuable support, coaching, encouragement, quick feedback and guidance helped me to finish this research project.

This study, off course, would not have been possible without the support of my family. I am very grateful to my parents Mr. Riaz Ahmad Gill and Asghari Baigum (late) for their precious love, blessings, support, encouragement and care since childhood. I am thankful to my lovely wife Saja and her family, my dearest sisters Fakhra, Fozia and Zobia, and my brother Kashif.

I would like to express my sincere thanks to my all colleagues and friends for their valuable feedback and support, especially Prof. Hung Nguyen, Dr. Tom McBride and my friends Ommar Sabbagh and Adam Alami.

At last, but most dearest, I am really grateful to my late mother for her support and encouragement to do the PhD. May God bless her soul and a place in heaven. I want to name this thesis to my late mother who would not be able to see me graduating.

I also wish to thank the Australian Research Council for the financial support. I am thankful to the people from both the research community and the software industry experts who helped me with their valuable feedback and experience throughout this research project.

I am thankful to all the reviewers for the valuable feedback and comments.

Thank you all.

Research Contribution and Publications

During this PhD. research project, I collaborated with my supervisor and other colleagues, and published the components of this research work (ASSF) in a number of rigorously reviewed international conference papers and scientific journals. The writing of papers, for publication, was an opportunity to present my work for review before including in my thesis. Please find the list of the publications, which have been included in this thesis, in Appendix F.

Table of Contents

List of Figures

List of Tables

Abbreviations, Acronyms

AAIL	Agility Adoption and Improvement Lifecycle
AAIP	Agility Adoption and Improvement Process
AAIM	Agility Adoption and Improvement Model
AAIS	Agility Adoption and Improvement Scorecard
ASD	Adaptive Software Development
APEP	Agile Product-Enhancement Process
ASOP	Agile Service Oriented Process
ASSF	Agile Software Solution Framework
APLM	Agile Process Lifecycle Management
ASL	Application Service Library
BDI	Belief, Desire and Intention
CAM	Contextual Analysis Model
CAME	Computer Assisted Method Engineering
CMMI	Capability Maturity Model Integration
DSDM	Dynamic Software Development Method
FCs	Framework Characteristics
FaSS	Framework as Software Service
FDD	Feature Driven Development
FOI	Freedom of Information
FY	Flexibility
IEC	International Electrotechnical Commission
ISD	Information Systems Development
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
KAII	Key Agility Indicators Index
LG	Learning
LS	Leanness
ME	Method Engineering
OPF	OPEN Process Framework
OMG	Object Management Group
OOSPICE	Object-Oriented Software Process Improvement and Capability

	dEtermination
PRINCE	Projects IN Control Environment
PMO	Project Management Office
RS	Responsiveness
SD	Speed
SDLC	Software Development Lifecycle
SDM	Software Development Methodology
SME	Situational Method Engineering
SE	Software Engineering
SEMDM	Software Engineering Metamodel for Development Methodologies
SOA	Service Oriented Architecture
SPEM	Software Process Engineering Metamodel
SPI	Software Process Improvement
XP	eXtreme Programming
4-DAT	4-Dimensional Analytical Tool

Glossary

Agility	Agility refers to agility attributes such as speed, flexibility, learning, leanness and responsiveness as well as agile values and principles.
Agility Adoption	Agile process adoption in a non-agile environment.
Agility Improvement	Agile process improvement in an existing agile environment.
Abstraction	It refers to a logical view of an entity such as object, service, agent or component.
Agile Knowledge	It refers to the knowledge that a person or group knows about agile software development.
Framework	It refers to a set of software development element and components that can be combined to produce a tailored software process/method.
Multi-Abstraction	It refers to a situation/project which involves more than one abstraction such as object and agent or agent and service.
Method Engineering/ Tailoring	The method engineering and method tailoring have been used interchangeably in this thesis. [Although, it is acknowledged that in other areas of software engineering these terms are used differently – these two terms have been used interchangeably in order to avoid any arguments.]
Process/Method	The process and method have been used interchangeably in this thesis. [Although, it is acknowledged that in other areas of software engineering these terms are used differently. Each forum seems to have its own individual usage – these two terms have been used interchangeably in order to avoid these arguments.]
Process/Method Fragment	A smallest unit of software process or method.
Traditional/ Plan-Driven Methods	The traditional and plan-driven methods have been used interchangeably in this thesis. [Although, it is acknowledged that in other areas of software engineering these terms are used differently – these two terms have been used interchangeably in order to avoid any arguments.]
Practice	It refers to an activity or fragment in a software process/method.

Abstract

The innovative well-known agile methods offer many powerful agile software development practices and have received considerable attention from both practitioners as well as the research community. While many organizations are interested in adopting agile methods suitable to their local circumstances, there is little guidance available on how to do so. Organizations, especially on the large-scale, currently lack systematic support for adopting agile methods in their complex software development settings. To address this important issue, this research proposes an agile software solution framework (ASSF) to both assistance in the assessment of the capability of the organization or team and tailoring of agile method in order to support the systematic adoption and improvement of agility in both agile and, incidentally, non-agile software development environments - especially formal and large environments. The ASSF has been incrementally developed by the iterative application of build, review and adjust research activities, which is called here a "qualitative empirical" research method. The ASSF is intended for use by agile coaches and consultants as a comprehensive information guide. The ASSF has two main components: framework characteristics and lifecycle management. The framework characteristics component incorporates 10 main elements or attributes to describe the agile-hybrid software development methodologies: (1) people (2) process, (3) product, (4) tools, (5) agility, (6) abstraction, (7) business value, (8) policy (9) rules and (10) legal. The framework lifecycle management component specifies the stages, practices and resources in order to support the systematic adoption and improvement of agility. The framework stages refer to an agility adoption and improvement lifecycle, its practices refer to an agility adoption and improvement process, and its resources refer to models, templates and toolkit that can be used during the agility adoption and improvement process such as the contextual analysis model, a key agility indicators index, an agility adoption and improvement model, an agility adoption and improvement scorecard, and an agile toolkit. The components of this framework have been empirically analysed and reviewed by experts from industry as well as the research community, and updated based on the feedback received. The results of this research indicated that the proposed ASSF framework may be considered reasonable for a gradual successful transition or adoption of agile practices in formal and large software development environments.