Book

Situational Method Engineering

HENDERSON-SELLERS, Brian, *et al*.

Abstract

While previously available methodologies for software – like those published in the early days of object technology – claimed to be appropriate for every conceivable project, situational method engineering (SME) acknowledges that most projects typically have individual characteristics and situations. Thus, finding the most effective methodology for a particular project needs specific tailoring to that situation. Such a tailored software development methodology needs to take into account all the bits and pieces needed for an organization to develop software, including the software process, the input and output work products, the people involved, the languages used to describe requirements, design, code, and eventually also measures of success or failure. The authors have structured the book into three parts. Part I deals with all the basic concepts, terminology and overall ideas underpinning situational method engineering. As a summary of this part, they present a formal meta-model that enables readers to create their own quality methods and supporting tools. In Part II, they explain how to implement SME in practice, [...]

Reference

HENDERSON-SELLERS, Brian, *et al*. *Situational Method Engineering*. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2014, 330

DOI : 10.1007/978-3-642-41467-1

Situational Method Engineering

Brian Henderson-Sellers • Jolita Ralyté •
Pär J. Ågerfalk • Matti Rossi

Situational Method Engineering

Springer

Brian Henderson-Sellers
School of Software
University of Technology, Sydney
Broadway
New South Wales
Australia

Jolita Ralyté
CUI, Battelle - bâtiment A
University of Geneva
Carouge
Switzerland

Pär J. Ågerfalk
Dept. of Informatics and Media
Uppsala University
Uppsala
Sweden

Matti Rossi
Aalto University
Aalto
Finland

ISBN 978-3-642-41466-4 ISBN 978-3-642-41467-1 (eBook)
DOI 10.1007/978-3-642-41467-1
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014941286

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Despite the vast amount of research undertaken over the years to find a single 'one-size-fits-all' methodological approach to software development projects, it is increasingly accepted that there will never be such a 'holy grail'.

Therefore, several research groups worldwide have adopted an alternative approach to software development. Under the generic banner of 'situational method engineering' or SME, software is developed specifically to fit the requirements of a single industry or a specific project.

SME encompasses all aspects of creating, using and adapting a software development method based on local requirements. Put simply, SME involves engineering a software development method for a particular situation—as opposed to buying an 'off-the-shelf' methodology and using it unchanged.

While each research group has adopted a different approach to SME, their leaders have come together in this book to provide a coherent synthesis.

The authors clearly and compellingly outline the components needed for an industry to put the SME approach to software development into practice. They assess the advantages and disadvantages of using method fragments, chunks or components, and discuss the social context in which method engineering best operates.

The first part of the book concludes with a more formal analysis of SME, using metamodelling techniques, which introduces all the necessary elements.

The second part of the book makes suggestions about the more practical aspects of SME and concludes with several case studies, emphasising how research can become best practice.

This is the first book to provide its readers all the tools they need to fully understand SME. It highlights future research and includes an extensive literature list, which provides an excellent starting point for potential research students undertaking their doctoral or postdoctoral work.

Practitioners will also find value, especially in the second part of the book.

The authors, Brian Henderson-Sellers, Jolity Ralyté, Pär J. Ågerfalk and Matti Rossi, are all well-respected and esteemed researchers in their chosen fields. They have not only undertaken the formal and theoretical research but put their ideas into practice within their local industries.

I commend Henderson-Sellers, Ralyté, Ågerfalk and Rossi on the hugely beneficial research they have undertaken in the area of SME. I am confident that this text will prove an invaluable resource for those interested in improving the standard of software development and the resulting software applications.

Mary O'Kane
NSW Chief Scientist & Engineer

Foreword

Turning research into practical, industry applicable knowledge, especially in information technology, is always a challenge. This book bridges the gap between research and industry applicability in the area of Situational Method Engineering (SME).

SME originated in the mid-1990s; although much of the early work did not label itself as such. In the early days of my own software development company, Object Consulting, we used these early SME ideas to create a toolset, Process Mentor, that has been extensively used in local industry. There are many challenges with introducing such ideas into organisations including competing against the 'not invented here' syndrome, dealing with entrenched 'standards teams' who often slavishly drove off-the-shelf packages or dealing with simple organisational apathy around process. Despite great advances in software development, software process remains a relatively immature area in most organisations.

With an SME approach as described in this book (and encapsulated in products like Process Mentor) the aim is to provide a robust yet flexible mechanism for constructing software development methods, resulting in high quality methods for each situation.

Industry best practice relies on proven techniques and approaches, some of which are formal and some more informal in the form of heuristics. This book provides an excellent and comprehensive review of the research in the SME field in Part 1, and then in Part 2 provides a detailed framework for reviewing and developing an SME approach together with a range of heuristics to construct development methods. As such the book is useful for both researchers as a summary of the latest thinking in the field, as well as the practitioner looking to understand the breadth and depth of material available to them when looking at developing an SME approach. This book deftly balances the advantages of such an approach in terms of practical application, underpinned by the solid theory from worldwide research.

It is an excellent and comprehensive SME book with no rival—and I heartily recommend it for both researchers and practitioners.

Julian Edwards
Chief Operating Officer, Object Consulting

Preface

Most people we know don't read a book's Preface. So, we'll keep it brief.

Why read this book? Why did we write it? The answer to both these questions is straightforward. We, the four authors of this book, have all, independently and more recently collaboratively, been working with Situational Method Engineering for almost two decades. Yet, all our published efforts are in conferences and journals so that when someone wants to join our several SME teams, there is no single source to which we can refer them in order that they can 'get up to speed'.

Now there is. This is the first book-length summary of everything we know about situational method engineering (SME) at the present time.

In this book, we present an overview and introduction to the topic of SME. SME provides a solution to the problem of the selection, creation or modification of existing methodological information in order to provide the 'most appropriate' methodology for an organisation and/or its projects. By a (software development) methodology, we mean all the bits and pieces that are needed for an organisation to undertake a software development. That means understanding how the process works; what the input and output work products are; who are the people involved; what languages are used to describe requirements, design, code, etc.; and when all this happens in terms of scheduling and other project management issues. It also includes evaluation of quality, productivity and other measures of success (or failure). The problem is that previously available methodologies for software—like those published in the early 1990s in the early days of object technology—claim to be appropriate for every conceivable project. This is clearly an ambit claim that is not supported in practice. Rather, SME acknowledges that all (or most) projects are individualistic such that the most efficacious methodology for that particular project needs individual tuning. This is where method construction (using SME) followed by method tailoring comes into its own.

We have structured the book into three parts. Part I deals with all the basic concepts, terminology and overall ideas underpinning SME. In Part II, we explain how you do SME in practice; how to find method parts and put them together and how to evaluate the resulting method. Part III is much shorter and summarises some of the more recent (and futuristic) ideas being discussed in the SME community.

SME's origins, as we shall explain in detail in Chap. 1, resulted from the frustration of finding (or developing) a single method for all situations. The alternative to the one-size-fits-all methodology of the 1980s and 1990s was our recognition that a

constructed method, suitably tailored to a specific context or situation that exists within a specific industry sector and/or project could be more efficacious. Early work originated in the Netherlands and then in Finland, Sweden, France, Switzerland and Australia. The authors of this book reflect these trail-blazing centres of SME.

There are several kinds of method parts used in SME. These have arisen from our different projects and are called method fragments, method chunks and method components; there are also 'larger' parts such as patterns that we discuss in Chap. 2. Following this detailed comparison of these method parts, we then introduce the overall social context, in particular the notion of method rationale which, in turn, leads to method-user-centred method engineering (Chap. 3).

These basic ideas, technical and sociological, are then combined in the subsequent chapters. Chapter 4 introduces some of the underlying theory and formal representations for SME, in particular metamodels for method construction and the current ISO standard relevant (ISO/IEC 24744). This chapter also introduces some basic ideas from ontology engineering relevant to our discourse.

In Chaps. 5–9 we examine SME in practice. Chapters 5 and 6 analyse *how* a method can be constructed from method parts—how to identify and locate the parts, approaches for method construction and the importance of reuse. Method configuration and method tailoring are the focus of Chap. 7, including a discussion on supporting tools for construction and customisation.

In Chap. 8, we focus on the more difficult issue of quality assessment—the quality of the method parts, the method base and the constructed method (both on paper and in action). In Chap. 9, we present examples in several domains/contexts of SME-constructed methods.

Chapters 10 and 11 form Part III and address more futuristic ideas within SME. In particular, we look at how recent ideas in services can be usefully addressed from an SME perspective and how large metamodels can themselves be tailored to create project-specific metamodels.

We have thus gathered together these originally disparate strands of SME into a coherent whole so that the 'SME novice' has a single point of entry into this fascinating and highly industry-relevant research topic. Although most SME published work has been in the research area, industry today is moving towards its adoption—as seen in the case studies in Chap. 9—sometimes under a name other than method engineering.

We need to include in this Preface some appreciation of copyright holders and other support. In particular, we note that a number of paragraphs in this book have been included from our previously published research papers.

In particular, we draw heavily on a publication by the first two authors (BH-S and JR) in the *Journal of Universal Computer Science*, 16(3), 424–478 ('Situational Method Engineering: state-of-the-art review').

We also acknowledge the original publications for some text as follows:

Chapter 2 contains some text taken from Henderson-Sellers, B., Gonzalez-Perez, C. and Ralyté, J., 2008, Comparison of method chunks and method fragments for situational method engineering, *Proceedings 19th Australian Software Engineering Conference. ASWEC2008*, IEEE Computer Society, Los Alamitos, CA, USA,

479–488. It also contains some material from Karlsson F and Ågerfalk P J (2009) Towards Structured Flexibility in Information Systems Development: Devising a Method for Method Configuration, *Journal of Database Management*, 20(3), pp. 51–75.

Part of Chap. 3 is based on previous publications by Ågerfalk and Fitzgerald: Ågerfalk P J (2006) Towards Better Understanding of Agile Values in Global Software Development. Proceedings of Eleventh International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD'06), Luxembourg, 5–6 June 2006; Ågerfalk P J and Fitzgerald B (2006) Exploring the Concept of Method Rationale: A Conceptual Tool for Method Tailoring, In Advanced Topics in Database Research, Vol. 5, pp. 63–78, (Ed, Siau K) Hershey, PA: Idea Group.

Chapter 4 contains some text taken from Henderson-Sellers, B., 2007, On the challenges of correctly using metamodels in method engineering, keynote paper in *New Trends in Software Methodologies, Tools and Techniques. Proceedings of the sixth SoMeT_07* (eds. H. Fujita and D. Pisanelli), IOS Press, Frontiers in Artificial Intelligence and Applications, vol. 161, 3–35. Some ideas and parts of text of Sect. 7.3.1 are from Kelly, S., Rossi, M., & Tolvanen, J.-P, (2005), What is Needed in a MetaCASE Environment? Journal of Enterprise Modelling and Information Systems Architectures, 1(1), pp. 1–11.

Chapter 6 contains some text taken from Gonzalez-Perez, C., Giorgini, P. and Henderson-Sellers, B., 2009, Method construction by goal analysis, in *Information Systems Development. Challenges in Practice, Theory, and Education* (eds. C. Barry, K. Conboy, M. Lang, G. Wojtkowski and W. Wojtkowski), Springer-Verlag, New York, USA, 79–92. It also contains some material from Karlsson F and Ågerfalk P J (2009) Towards Structured Flexibility in Information Systems Development: Devising a Method for Method Configuration, *Journal of Database Management*, 20(3), pp. 51–75.

Chapter 7 uses a paragraph from Hug, C., Front, A., Rieu, D. and Henderson-Sellers, B., 2009, A method to build information systems engineering process metamodels, *J. Systems Software*, 82(10), 1730–1742.

It also contains ideas and examples originally published by Karlsson and Ågerfalk, MC Sandbox: Devising a Tool for Method-User-Centred Method Configuration, *Information and Software Technology*, 54(5), pp. 501–516.

Chapter 8 uses text originally published in Henderson-Sellers, B., 2011a, Random thoughts on multi-level conceptual modelling, chapter in *The Evolution of Conceptual Modeling* (eds. L. Delcambre and R. Kaschek), LNCS 6520, Springer-Verlag, Berlin, 93–116.

and from two papers presented at the ME 2011 conference in Paris:

Henderson-Sellers, B. and Gonzalez-Perez, C., 2011, Towards the use of granularity theory for determining the size of atomic method fragments for use in situational method engineering, *Engineering Methods in the Service-Oriented Context. 4th IFIP WG8.1 Working Conference on Method Engineering, ME 2011, Paris France, April 2011, Proceedings*, (eds. J. Ralyté, I. Mirbel and R. Deneckère), Springer, Heidelberg, 49–63.

McBride, T. and Henderson-Sellers, B., 2011, A method assessment framework, *Engineering Methods in the Service-Oriented Context. 4th IFIP WG8.1 Working Conference on Method Engineering*, *ME 2011*, *Paris France*, *April 2011*, *Proceedings*, (eds. J. Ralyté, I. Mirbel and R. Deneckère), Springer, Heidelberg, 64–76.

Section 9.1 utilises the examples from Henderson-Sellers, B., Serour, M., McBride, T., Gonzalez-Perez, C. and Dagher, L. 2004b. Process construction and customization. *Journal of Universal Computer Science*. 10(4), 326–358.

Section 9.4 is based on Rossi, M. and Tuunanen, T., 2010, A method and tool for rapid consumer application development, *International Journal of Organisational Design and Engineering*, 1(1/2), 109–125.

Having said our thanks to publishers of our original research, we also wish to make some personal thanks—to those of our colleagues and students who read our earlier drafts of these chapters, in particular Marko Bajec, Rebecca Deneckère, Sergio Espana, Mahdi Fahmideh Gholami, Akhtar Ali Jalbani, Fredrik Karlsson, Elena Kornyshova, Graham Low, Ben Rogers, Colette Rolland, Motoshi Saeki, Juha-Pekka Tolvanen and Kai Wistrand.

Finally, BH-S acknowledges the continuing support of Ann; JR Colette, Isabelle, Michel; PJA Kajsa, Amanda, Algot, MR Tuuli, Saana, Sippo and Samu.

Glossary of Acronyms

AD	Activity Diagram
ATL	ATLAS Transformation Language
BPM	Business Process Modelling
BPMN	Business Process Modeling Notation
BWW	Bunge-Wand-Weber
B2C	Business to Consumer
CAME	Computer-Aided Method Engineering
CASE	Computer-Aided Software Engineering
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
COBIT	Control Objectives for Information and Related Technology
COMMA	Common Object Methodology Metamodel Architecture
COTS	Commercial Off-The-Shelf
CRC	Class Responsibility Collaborator (cards)
DSDM	Dynamic Systems Development Method
ER	Entity Relationship
ERP	Enterprise Resource Planning
FDT	Formal Description Technique
FIPA	Federation for Intelligent Physical Agents
GOPPRR	Graph-Object-Ports-Property-Relationship-Role
GOPRR	Graph-Object-Property-Relationship-Role
GQM	Goal Question Metric
GUI	Graphical User Interface
IAG	Intention Achievement Guideline
IBIS	Issue-Based Information Systems
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFIP	The International Federation for Information Processing
ISD	Information Systems Development
ISDM	Information Systems Development Method
ISE	Information Systems Engineering
ISG	Intention Selection Guideline
ISO	International Organization for Standardization

IT	Information Technology
JTC1	Joint Technical Committee 1 (between ISO and IEC)
LOC	Lines of Code
MaaS	Method as a Service
MC	Method Configuration
MDA	Model-Driven Architecture
MDD	Model-Driven Development
MDE	Method-Driven Engineering
ME	Method Engineering
MEL	Method Engineering Language
MIS	Management Information Systems
MMC	Method for Method Configuration
MOA	Method-Oriented Architecture
MOBMAS	Methodology for Ontology-Based Multi-agent Systems
MOF	Meta-Object Facility
MOSES	Methodology for Object-oriented Software Engineering of Systems
NIAM	Nijssen's Information Analysis Methodology (later renamed Natural language Information Analysis Method)
OLMS	Object Library Management System
OMG	Object Management Group
OMT	Object Modeling Technique
OO	Object-Oriented or Object Orientation
OOSE	Object-Oriented Software Engineering
OOSPICE	Object-oriented Software Process Improvement and Capability dEtermination
OPEN	Object-oriented Process, Environment and Notation
OPF	OPEN Process Framework
OPM3	Organisational Project Management Maturity Model
OPRR	Object-Property-Relationship-Role
PDD	Process-Data Diagram (later renamed Process Deliverable Diagram)
PMUC	Process Metamodel Under Construction
REMAP	REpresentation and MAintenance of Process knowledge
RUP	Rational Unified Process
SC7	Subcommittee 7 (a committee of ISO/JTC1 dealing with software engineering standards)
SEI	Software Engineering Institute (at Carnegie Mellon University)
SEM	Systems Engineering Method
SEMDM	Software Engineering Metamodel for Development Methodologies
SEP	Software Engineering Process
SIMM	Service Integration Maturity Model
SME	Situational Method Engineering
SMME	Situational Metamodel Engineering
SMSDM	Standard Metamodel for Software Development Methodologies
SOA	Service-Oriented Architecture
SOMA	Semantic Object Modeling Approach

SPC	Software Process Control
SPEM	(version 1) Software Process Engineering Metamodel (version 2) Software & Systems Process Engineering Metamodel
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination
SPLE	Software Product Line Engineering
SPM	Software Product Management
SSG	Strategy Selection Guideline
SUS	System Under Study
TAME	Tailoring A Measurement Environment
UML	Unified Modeling Language
VIBA	Versatile Information and Business Analysis
XP	eXtreme Programming

Contents