

Pro Visual Studio Team System Application Lifecycle Management



Joachim Rossberg

Pro Visual Studio Team System Application Lifecycle Management

Copyright © 2008 by Joachim Rossberg

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13: 978-1-4302-1080-1

ISBN-13 (electronic): 978-1-4302-1079-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Tony Campbell

Technical Reviewers: Norman Guadagno, Dan Massey

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Beth Christmas

Copy Editor: Sharon Wilkey

Associate Production Director: Kari Brooks-Copony

Production Editor: Laura Esterman

Compositor: Patrick Cunningham

Proofreader: Linda Seifert

Indexer: Becky Hornyak

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

Amelie: You swept me away when you entered my life. I love you.

Opus: I miss the furry, warm shape of your body on the keyboard.

I'll see you again sometime, I'm sure about that.

Karin: Thanks for all the support and for giving me such a wonderful daughter. Love you.

Gaston: Best (living) cat in the world. Thanks for staying away from the keyboard.

Contents at a Glance

Foreword	xiii
About the Author	xv
About the Technical Reviewers	xvii
Acknowledgments	xix
Introduction	xxi
■ CHAPTER 1 Why Do We Develop Business Software and Systems?	1
■ CHAPTER 2 Application Lifecycle Management	23
■ CHAPTER 3 Development Processes and Frameworks	43
■ CHAPTER 4 The Agile Enterprise—Or How We Could Bridge the Gap	87
■ CHAPTER 5 ALM Assessments	127
■ CHAPTER 6 Application Lifecycle Management Using Visual Studio Team System	157
■ CHAPTER 7 Working with VSTS Customization to Further Enhance the ALM Process	203
■ CHAPTER 8 Real-Life Example	279
■ INDEX	303

Contents

Foreword	xiii
About the Author	xv
About the Technical Reviewers	xvii
Acknowledgments	xix
Introduction	xxi
■ CHAPTER 1 Why Do We Develop Business Software and Systems? ...	1
Understanding the Cornerstones of Business	2
Processes	2
Business Rules	3
Information	4
Understanding the Need for Business Software	4
Today's Business Environment and the Problems We Face	5
Project Health Today: Three Criteria for Success	6
Factors Influencing Projects and Their Success	8
Project Success in Research	11
IT Budget Spending	17
Summary	21
■ CHAPTER 2 Application Lifecycle Management	23
Roles in the ALM Process	23
Four Ways of Looking at Application Lifecycle Management	26
The SDLC View	27
The Service Management or Operations View	28
The Application Portfolio Management View	30
The Unified View	30
Three Pillars of Application Lifecycle Management	31
Traceability of Relationships Between Artifacts	32
Automation of High-Level Processes	32
Visibility into the Progress of Development Efforts	32

A Brief History of ALM Tools	32
Application Lifecycle Management 1.0	34
Application Lifecycle Management 2.0	36
ALM and PPM	39
Summary	41

■ CHAPTER 3 **Development Processes and Frameworks** 43

Waterfall Model	44
Spiral Model	46
Rational Unified Process (RUP)	48
The Principles of RUP	49
The RUP Lifecycle	51
Disciplines	54
Work Products, Roles, and Tasks	57
Manifesto for Agile Software Development	57
Extreme Programming (XP)	59
Scrum	60
Empirical Process Control	61
Complexity in Projects	62
What Scrum Is	63
The Roles in Scrum	64
The Scrum Process	66
Essential Unified Process (EssUP)	69
The Practices of EssUP	69
Implementing EssUP	75
Capability Maturity Model Integration for Development (CMMI-DEV)	75
Microsoft Solutions Framework (MSF)	77
Key Elements in MSF	79
Roles in MSF	81
The MSF Lifecycle	83
MSF for Agile and MSF for CMMI	84
Summary	85

CHAPTER 4	The Agile Enterprise— Or How We Could Bridge the Gap	87
	Radically Simple IT	89
	Enterprise Architecture	91
	Service-Oriented Architecture	92
	Background to SOA	92
	What Is a Service?	94
	What Is This Thing Called SOA?	95
	Why Use SOA?	96
	Characteristics of Services: The Four Tenets of Don Box	98
	More on Characteristics: Sundblad and Sundblad	100
	Services and Business Processes	106
	Other Benefits of Automating Business Processes	109
	A Redefined View of the Architect Role(s)	109
	Architect Roles According to IASA Sweden	111
	The Sundblads' Architect Roles	116
	Conclusions on Roles So Far	125
	Summary	125
CHAPTER 5	ALM Assessments	127
	Microsoft Application Platform Optimization (APO) Model	128
	Infrastructure Optimization Model	129
	Business Productivity Infrastructure Model	130
	APO Maturity Levels	130
	APO Capabilities	132
	The APO Assessment	134
	Starting the Assessment	134
	Example 1: Assessing the Development Process	136
	Example 2: Assessing the User Experience	136
	Viewing the Results	137
	Application Platform Capability Assessment	140
	Starting the Assessment	142
	Example 1: Assessing Requirements Handling	144
	Example 2: Assessing Code Analysis	144
	Viewing the Results	145
	The Assessment Builder	147

How to Use These Tools	151
Using the APO Assessment	152
Using the Application Platform Capability Assessment	152
Using the Assessment Builder	153
Following a Schedule	153
Why Do an Assessment?	156
Summary	156

CHAPTER 6	Application Lifecycle Management Using Visual Studio Team System	157
	Application Lifecycle Management Overview	157
	Visual Studio Team System Overview	158
	Team Foundation Server	159
	Process Template	160
	VSTS Editions	161
	Web Access	162
	Microsoft Office	163
	Add-Ins	164
	Traceability	165
	The TFS Work Item Tracking System	166
	Configuration Management Using VSTS	171
	Automation of High-Level Processes	181
	Visibility	182
	Collaboration	188
	The Project Portal and the VSTS Web Access	189
	Work Items for Collaboration	191
	System Designers in the Architect Edition for Collaboration	191
	Service-Oriented Architecture	192
	IT and Business Gap	193
	Office/MS Project Integration	193
	Requirements Analysis in VSTS	195
	Use of One Tool/Role Based	198
	Extensibility	199
	Summary	200

CHAPTER 7	Working with VSTS Customization to Further Enhance the ALM Process	203
	The Overall Architecture of Visual Studio Team System	204
	Architecture of Team Foundation Server	204
	Visual Studio Team System Deployment	207
	Team Explorer	209
	Visual Studio Professional	211
	Editions of Visual Studio Team System	212
	Features That All Editions Have in Common	212
	Visual Studio Team System Architecture Edition	213
	Visual Studio Team System Database Edition	214
	Visual Studio Team System Development Edition	216
	Visual Studio Team System Test Edition	218
	Automation of High-Level Processes	225
	Project Process Template	225
	Microsoft Solutions Framework	228
	Scrum for Team System	230
	eScrum	235
	Other Process Templates	237
	Visibility	237
	Code Metrics	237
	Code Analysis	240
	Code Profiling	242
	Code Coverage	243
	Traceability	244
	Test-Driven Development	245
	Unit Testing in VSTS	247
	Automated Build/Continuous Integration	250
	Check-In Policies	254
	Extensibility	257
	Modifying the Process Template	257
	Common Adaptations of the Process Template	259
	Modifying Work Items	264
	Collaboration	267
	Architecture with Visual Studio Team System	267
	Operations with VSTS	276
	Requirements Specifications	277
	Extensibility	278
	Summary	278

■ CHAPTER 8	Real-Life Example	279
	Customer Case.....	279
	ALM Assessment.....	280
	Meeting 1	281
	Meeting 2	283
	Meeting 3	288
	Workshop with Priorities (Lowest-Hanging Fruits).....	288
	Implementation	290
	Startup Project	291
	Pilot of MSF for Agile and Scrum for Team System.....	295
	Follow-Up	299
	Summary.....	300
■ INDEX		303

Foreword

Developing software is hard—really hard. Even more so when you think about the fact that as an industry we do not typically examine software development in a holistic approach. That is, we don't ask ourselves questions such as “What are the business drivers for this project?” “What benefits will this project provide us?” and “How do we determine if the project is successful?” In many cases, the software development industry is concerned with simply developing the software.

Software failures cost companies billions (yes, that's not a typo—*billions* with a *b*) of dollars every year. Organizations can no longer afford to ignore these failures. They have to understand how software helps them reach their goals and how to prioritize investments in software. Organizations must also understand the difference between software that provides value and successful software projects (which may not be the same thing). Unfortunately, this has been a difficult transition.

I have worked with numerous customers who do not know the benefit of building a particular piece of software. Likewise, I have worked with numerous developers who did not understand the importance of the software they were building. Situations like these are unacceptable in the current marketplace. Everyone must understand the context in which applications are being built and the fact that the context can change during the development process. Companies that cannot respond to change effectively are no longer in business.

Application Lifecycle Management (ALM) is a relatively new term describing the overall process of prioritizing, developing, and managing software development projects to provide maximum benefit to the business organization. ALM principles have been around for a number of years now, but the focus on software development organizations is just starting to come around. The “What cool new thing can we build now?” philosophy that was so popular during the Internet boom (and subsequent bust) is now giving way to “What cool new thing can we build now *that will help improve our business?*” This is a fundamental shift and an acknowledgment that the role of IT organizations has to change.

To make this shift, you have to understand how your organization develops software right now. Do you have a governance process? Do you have a standard Software Development Lifecycle? Do you have a standard operations framework? And more important, how well do these things work together? Microsoft started the ALM assessment program as a way to help companies understand these three major areas of ALM. These assessments give companies insight into what they are doing that works well and what areas they need to target for improvement. It also enables companies to prioritize their improvements and introduce changes rapidly but in a controlled manner.

Up to this point, I haven't mentioned tools. Tools can certainly help organizations accomplish their jobs better—but without understanding the underlying process, tools only exacerbate the current situation. But how do you deal with a process that needs to change over time? You need a tool that can change with you. Microsoft Visual Studio Team System (VSTS) is the premier tool to help you manage your software development projects with an eye on the business. It brings the business perspective to the developers and enables the business to view important software project metrics. VSTS also provides you the necessary metrics to understand which areas need improvement and enable you to track that improvement over the course of numerous software development projects.

By combining the process of ALM with the technical capabilities of VSTS, most organizations can realize significant benefits with a carefully thought-out plan to improve the process. *Pro Visual Studio Team System Application Lifecycle Management* gives you the context and detailed understanding of how to use these tools to the benefit of your development team *and* your business.

Jeff Levinson
ALM practice lead, Northwest Cadence
Microsoft Team System MVP

About the Author



JOACHIM ROSSBERG is a project manager working and living in Sweden. These days he works for Know IT, one of Sweden's most renowned IT consulting firms, and he is also a member of the Visual Studio Team System Inner Circle program. Before working for Know IT, he was employed by Capgemini for several years, working with IT architecture, project management, and much more.

At Know IT, he works mainly as a project manager or scrum master (if you are unfamiliar with this term, Chapter 3 will tell you more about it). He is, together with his colleague in Stockholm, Jonas Samuelsson, responsible for Know IT's alliance with Microsoft, focusing mainly on Visual Studio Team System (VSTS).

Development processes are a favorite topic for Joachim, and he is a member of the Microsoft Customer Advisory Council for VSTS. He has cowritten two previous books with a colleague, Rickard Redler: *Designing Scalable .NET Applications* (Apress, 2003) and *Pro Scalable .NET 2.0 Application Designs* (Apress, 2005).

Before starting in the IT business in 1998, Joachim worked for 10 years as an assistant air-traffic controller in Halmstad, on the west coast of Sweden. He then earned a bachelors degree in psychology from the University of Gothenburg, including studies in criminology and pedagogy. Rather than join the academic research world, he started studying informatics and working for Capgemini.

On the personal side, Joachim was born in 1967 on the east coast of Sweden in a town called Kalmar. He lived in Halmstad, on the other side of Sweden, for 10 years while working there. Halmstad is also the town where he met his wife, Karin. In 1996, Joachim moved to Gothenburg a bit further up on the Swedish west coast (or best coast as the people there say) where he has lived ever since.

Karin and Joachim have a wonderful daughter, Amelie, born in 2007. They also have a cat called Gaston, who goes (on a leash) for long walks in the neighborhood. And yes, they do get the occasional stare from people. ☺ Opus, Joachim's cat and companion for 16 years who helped write Joachim's first two books for Apress by sleeping on the keyboard or monitor, passed away in 2006.

In Joachim's spare time, when he is not writing books, he listens to a lot of music. Some of his favorite artists are Arch Enemy, Neil Young, Bob Dylan, The Hives, Sahara Hotnights, Ulf Lundell, Bruce Springsteen, LCD Soundsystem, and Underworld. He also watches a lot of movies, an interest from his early working life as a cinema machinist. Joachim also tries to go to the gym and out for a jog a couple of times a week.

About the Technical Reviewers



■ **NORMAN GUADAGNO** is currently director of product marketing for Visual Studio Team System at Microsoft. In his previous role at Microsoft, he was responsible for the company's marketing efforts to the architect audience worldwide, including owning the Microsoft Strategic Architect Forum and the *Microsoft Architecture Journal*.

Norman has more than 10 years of experience in the software industry. Prior to Microsoft, he was VP of marketing at Qpass and VP of marketing and business development at Primus Knowledge Solutions. Earlier in his career, he was a senior director at Oracle on the Oracle Application Server team. Norman began his career as a usability engineer and in technical product management. He has a BA degree from the University of Rochester and an MA from Rice University. Norman and his family live and work in Redmond, Washington.

■ **DAN MASSEY** is a software architect in the Microsoft Visual Studio Team System group. Previously, he worked on Application Lifecycle Management (ALM) products at Borland and was a mentor at TogetherSoft. Dan began leading agile teams in 1999, starting with feature-driven development. Since then, Dan has managed agile delivery organizations and consulted on large-scale agile implementations using ICONIX, Extreme Programming, and Scrum. His areas of interest include agile architecture, domain modeling, SOA, and team-centric process improvement.

Acknowledgments

Sam Guckenheimer, Microsoft. Thanks for giving me valuable information and contacts at Microsoft.

Dan Massey, Microsoft. Dan, thanks for giving priceless input for the book. As a technical reviewer, you have been great. The book is so much better now after all your input.

Norman Guadagno, Microsoft. Norman, your feedback as a technical reviewer has been invaluable. Together with Dan, the two of you have improved the book very much.

Tony Campbell, Apress. Thanks for all the feedback, Tony. Having three technical reviewers scared me at first, but the improvements to the book were great.

Dag König, Microsoft. Dag, I missed giving you credit in my last book. I'm sorry about that. Thanks for the continuing support.

Jeff Levinson. Thanks for writing the foreword and giving feedback on my work.

Ewan Buckingham, Apress. Ewan, thanks for pushing me in deciding to write this book.

Ameya Bhatawdekar, Microsoft. Thanks for taking your time to discuss the Project Connector with me.

Andreas Nordenadler, Microsoft. Thanks for your support.

Colin Bird, Conchango. Thanks a lot.

Thomas Gustavsson, Know IT. Thanks for helping me out.

Mattias Olausson, Sweden. Thanks for helping me with some code samples and discussions.

Per Wallentin, Know IT. Thanks for all the support and the nice MacBook Air I have used to write this book.

Thanks to all the other people at Apress working on this book: Beth Christmas, Sharon Wilkey, Laura Esterman, and anyone else who was involved. It couldn't happen without you.

I'm sure I forgot someone in this list, but I thank everybody who has helped me and supported me along the line.

I also thank you, the reader, for purchasing this book. Without readers, there would be no books.

Introduction

At the end of 2006, Know IT was asked by Microsoft Sweden if we were interested in putting together a seminar tour introducing Visual Studio Team System (VSTS) in Sweden. This was too interesting for me to pass up, so a colleague and I set of to work. Microsoft gave us free rein in filling the seminars with topics, which was great because we then did not feel like we had to please them but instead could be realistic about the set of tool(s).

We were invited to visit important customers and talk about the benefits (and of course the concerns) of using VSTS in an enterprise. We traveled from the north of Sweden to the south and had a pretty interesting time. Some customers had already started adopting VSTS, but most of them were curious and had not taken the step yet.

During this tour, my interest in Application Lifecycle Management (ALM) started for real. I had of course heard the term before, but now I had to dive deep inside it to see what place VSTS had in all this. Before this, my focus had been mostly on the tool itself and never about where it fits in an organization. I was impressed by what I saw, and by what I felt such a tool could do in my own projects. I knew that the tool at that stage was not entirely ready for everything in ALM, but I felt it had great potential. And so far it has lived up to my expectations.

What This Book Will Cover

I will try to cover a great deal in this book. The primary focus of the book is to help organizations invest in IT projects that return business value to the customer. This can be done in many ways, and by investing in the Application Lifecycle Management process and tools supporting this process, I hope to give you my vision on this.

These are the chapters and a short overview of each.

Chapter 1

The first chapter focuses mainly on the IT project world of today. Far too many IT projects seem to fail or at least have significant overruns when it comes to time and budget. More important, many of them seem to not deliver the business value that the customer wants. And this is a very bad thing. You will look at some of the difficult questions that challenge IT organizations. What makes it so hard to calculate the outcome of an IT project? What are the challenges they face? Why don't IT systems align better with the business processes? Why don't IT projects seem to deliver the business value the customer wants? What can be done to eliminate these problems?

Chapter 2

In Chapter 2, we discuss what Application Lifecycle Management (ALM) really is. I define ALM so we will have a common ground to stand on for the rest of this book. You will also see how the ALM process can be supported by tools and what requirements we can have on such tools.

Chapter 3

All organizations have processes for how work is carried out. They could have a sales process, a procurement process, and so on. Chapter 3 focuses on the development process and gives a few examples of different popular processes, such as Microsoft Solutions Framework and Scrum.

Chapter 4

One of the biggest issues in IT development today is that there is a gap between the business and IT sides of many organizations. This chapter shows two topics that can help bridge this gap and that are important to address in the ALM process. The first is service-oriented architecture (SOA), which is a popular software system architecture intended to provide flexibility in our IT systems so we can better align these with the business process in the organization. This is extremely important for the ALM process to be successful.

The second topic is IT architecture and the roles of IT architects, big topics in ALM. Traditionally, an IT architect has a very technical position, holding deep technical knowledge. By redefining the architect role and extending it to cover business areas as well, we can make sure the IT systems really support the business and the business processes.

Chapter 5

I have seen many organizations implementing Visual Studio Team System (VSTS) over the last few years. Some of them have truly used the potential of the product while others have implemented only parts of it. Most companies I have seen, however, belong to the latter group and use mostly the VSTS version-control system.

By doing a thorough assessment of our ALM process (including the Software Development Lifecycle, or SDLC, process) by using a couple of tools from Microsoft, we can better see which parts of VSTS will help us most in improving our ALM process.

Chapter 6

Here you will see how to use VSTS to fulfill the ALM vision. In Chapter 6, I give an overview of what Visual Studio Team System is and what tools are included in it. This set of tools is Microsoft's answer to how we can improve our ALM process, including such things as a flexible development process foundation, collaboration platform, version-control system, development environment, work item workflow system, and much, much more.

After you read this chapter, I want you to have a good understanding of the benefits and concerns of using VSTS as an ALM tool.

Chapter 7

This chapter wraps everything up. Here you will see how to use VSTS to fulfill the ALM vision. You will have a look at customizations and deployments scenarios for VSTS, and how you can customize VSTS to achieve even better business value.

Chapter 8

The final chapter shows how I have used the ALM assessment I described in Chapter 5 to help organizations with their ALM process. I think this is extremely important to do before implementing any ALM tool in an organization. By assessing the ALM process, we can implement a process that will give business value to the organization. You will also see how to adopt the changes that an assessment has found necessary and implement these in the organization.

Who Should Read This Book

The main audiences for this book are IT managers and business managers with an IT interest as well as portfolio management decision makers. I aim to give project managers, scrum masters, and architects something to think about as well.

If you do not feel like you belong to these groups, don't hesitate to read this book anyway. I hope that I will have some treats for you as well. Developers, for example, can benefit from getting the bigger picture for their efforts. Where does your work fit into the ALM process? This is important to know because everybody involved in a development project is also an important participant in the ALM process.

