

# Social Software Engineering

Development and Collaboration with Social Networking



**Jessica Keyes**



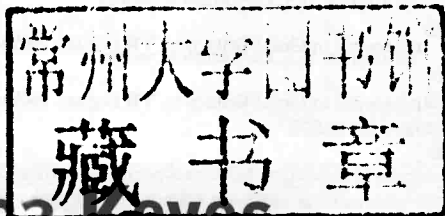
CRC Press  
Taylor & Francis Group

AN AUERBACH BOOK

# Social Software Engineering

Development and Collaboration with Social Networking

Jessica Keyes



**CRC Press**

Taylor & Francis Group  
Boca Raton London New York

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business  
AN AUERBACH BOOK

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2012 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed and bound in Great Britain by TJI Digital, Padstow, Cornwall  
Version Date: 20110803

International Standard Book Number: 978-1-4398-5375-7 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com/](http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at  
<http://www.taylorandfrancis.com>

and the CRC Press Web site at  
<http://www.crcpress.com>

# **Social Software Engineering**

Development and Collaboration with Social Networking

---

# Dedication

---

This book is dedicated to my family and friends.

I would especially like to thank those who assisted me in putting this book together. As always, my editor, John Wyzalek, was instrumental in getting my project approved and providing great encouragement.

---

# Preface

---

The world is a changed place. The collaborative web has caught our collective imagination and there is no turning back, particularly in the business world. Some have taken to calling this use of collaborative technologies in business Enterprise 2 (E 2.0). Wikipedia may have been the first company to popularize the phenomenon of user-generated knowledge, but this encyclopedia is just the tip of the iceberg. Companies far and wide are wiki-izing. Nokia hosts a number of wikis, some of which are used internally to coordinate technology research. Dresdner Kleinwort, an investment bank, operates the largest corporate wiki. About 50% of Dresdner staff use this wiki to make sure that all team members are on the same project management page.

E 2.0 is more than just wikis, of course. It constitutes the entirety of social networking applications including blogs, discussion boards, workspaces, and anything else that is sharable, and even combinable (i.e., mashups). IBM uses E 2.0 for everything from collaborative document production to internal project collaboration. Nokia uses it for all-purpose teamware. A whole host of companies use it for knowledge management. Honeywell was one of the first to use E 2.0 to perform knowledge discovery, research, and sharing across miles—regardless of whether users even know each other. It would appear, then, that E 2.0 using social networking technologies has wide applicability to all things business—including software engineering.

Software development projects are usually complex and often mission critical. Successful software development projects usually have something in common. Each of these projects, in some way, shape, or form, follows one or more principles of applied software engineering methodology. However, the precepts of software engineering have been around for decades. While advances in computer hardware and software have accelerated with breathtaking speed over these past decades, advances in software engineering have not quite kept pace. Modern systems that now integrate multiple platforms, multiple architectures, and usually seek a global reach via the Internet require modern software engineering methods.

Software engineering is an inherently collaborative set of socio-technical activities that permit distributed developers to discuss and share knowledge and artifacts using appropriate supportive processes and tools. This can be referred to as social software engineering.

The goal of this book is to examine the new field of software engineering through the spectrum of the social activities that compose it. Computer science must be fused with psychology, sociology, and mathematics, and wrapped around the principles of knowledge engineering, to develop an appropriate infrastructure (i.e., tools, applications, and environments) to support a software engineering discipline capable of developing twenty-first century applications.

---

# Author

---

**Jessica Keyes** is president of New Art Technologies, Inc., a high-technology and management consultancy and development firm started in New York in 1989.

Keyes has given seminars for such prestigious universities as Carnegie Mellon, Boston University, University of Illinois, James Madison University, and San Francisco State University. She is a frequent keynote speaker on the topics of competitive strategy and productivity and quality. She is former advisor for DataPro, McGraw-Hill's computer research arm, as well as a member of the Sprint Business Council. Keyes is also a founding board of director member of the New York Software Industry Association. She completed a two-year term on the Mayor of New York City's Small Business Advisory Council. She currently facilitates doctoral and other courses for the University of Phoenix and is a member of the Faculty Council for the College of Information Systems & Technology. She has been the editor for Warren, Gorham & Lamont's *Handbook of eBusiness* and CRC Press' *Systems Development Management* and *Information Management*.

Prior to founding New Art, Keyes was the managing director of R&D for the New York Stock Exchange and has been an officer with Swiss Bank Co. and Banker's Trust, both in New York City. She earned a master's in business administration from New York University and a doctorate in management. A noted columnist and correspondent with more than 200 articles published, Keyes is the author of the following books:

*The New Intelligence: AI in Financial Services*, HarperBusiness, 1990  
*The Handbook of Expert Systems in Manufacturing*, McGraw-Hill, 1991  
*Infotrends: The Competitive Use of Information*, McGraw-Hill, 1992  
*The Software Engineering Productivity Handbook*, McGraw-Hill, 1993  
*The Handbook of Multimedia*, McGraw-Hill, 1994  
*The Productivity Paradox*, McGraw-Hill, 1994  
*Technology Trendlines*, Van Nostrand Reinhold, 1995  
*How to be a Successful Internet Consultant*, McGraw-Hill, 1997  
*Webcasting*, McGraw-Hill, 1997  
*Datacasting*, McGraw-Hill, 1997

- The Handbook of Technology in Financial Services*, Auerbach, 1998  
*The Handbook of Internet Management*, Auerbach, 1999  
*The Handbook of eBusiness*, Warren, Gorham & Lamont, 2000  
*The Ultimate Internet Sourcebook*, Amacom, 2001  
*How to Be a Successful Internet Consultant, 2nd ed.*, Amacom, 2002  
*Software Engineering Handbook*, Auerbach, 2002  
*Real World Configuration Management*, Auerbach, 2003  
*Balanced Scorecard*, Auerbach, 2005  
*Knowledge Management, Business Intelligence, and Content Management: The IT Practitioner's Guide*, Auerbach, 2006  
*X Internet: The Executable and Extendable Internet*, Auerbach, 2007  
*Leading IT Projects: The IT Manager's Guide*, Auerbach, 2008  
*Marketing IT Products and Services*, Auerbach, 2009  
*Implementing the Project Management Balanced Scorecard*, Auerbach, 2010

---

# Contents

---

**Preface.....xi**

**Author ..... xiii**

**1 Why Social Networking?.....1**

The Social Network .....3

The Software Engineering Social Network .....7

Collaborative Applications .....8

We've Reached the End of Chapter 1 .....11

References.....11

**2 Social Networking Tools at Work.....13**

Tools that Provide Networking Capabilities .....13

Wikis in Action .....17

Meaning-Based Computing.....18

Semantic Web.....19

Virtual Worlds.....21

Knowledge Management Tools.....22

Mashups .....25

We've Reached the End of Chapter 2 .....27

References.....27

**3 Preparing Team to Collaborate.....29**

Nontechnological Issues in Software Engineering.....29

Creativity Development .....31

Communications and Group Productivity.....34

Productivity through Shared Information Technology .....34

Promoting Collaboration .....35

Collaboration and Knowledge Management.....37

Best Practices .....39

Expert Interviews.....40

Job Aids .....41

Knowledge Audits and Knowledge Fairs.....42

	Knowledge Maps and Inventories .....	42
	Learning Games .....	44
	Lessons-Learned Debriefings .....	46
	Storytelling .....	49
	We've Reached the End of Chapter 3 .....	51
	References .....	51
<b>4</b>	<b>Knowledge Sharing and Software Engineering Teams: A Study .....</b>	<b>53</b>
	Knowledge Management .....	53
	Knowledge Management Issues .....	55
	Knowledge Management and Knowledge Sharing .....	56
	Organizational Culture's Effect on Knowledge Sharing .....	59
	IT Support's Effect on Knowledge Sharing .....	60
	Techniques for Promoting Knowledge Sharing .....	62
	IT Worker Study .....	63
	Question 1: What are the cultural reasons that cause employees to resist the sharing of knowledge? .....	65
	Question 2: What are the organizational reasons that cause employees to resist the sharing of knowledge? .....	67
	Question 3: What are the key reasons employees list for not wanting to share their expertise? .....	71
	Recommendations for Making It Work .....	79
	We've Reached the End of Chapter 4 .....	82
	References .....	82
<b>5</b>	<b>Action Learning Teams .....</b>	<b>87</b>
	Action Learning .....	87
	Leaderful Practice .....	89
	Action Projects .....	92
	We've Reached the End of Chapter 5 .....	93
	References .....	93
<b>6</b>	<b>Knowledge across Social Networks .....</b>	<b>95</b>
	Heterogeneity of Information .....	96
	Intelligent Systems .....	96
	Semantic Standards .....	99
	Enterprise Information Management .....	101
	Content Management System (CMS) .....	102
	Document Management and Electronic Document Management Systems .....	104
	Digital Asset Management .....	105
	Data Mining .....	105
	We've Reached the End of Chapter 6 .....	107
	References .....	107

<b>7</b>	<b>Measuring Social Software Engineering .....</b>	<b>109</b>
	Effective Project Management .....	109
	Balanced Scorecard for Project Management .....	110
	Adopting a Balanced Scorecard.....	111
	Example: FedEx.....	117
	Attributes of Successful Project Management Measurement Systems.....	119
	Project Management Office .....	120
	Project Management Process Maturity Model (PM <sup>2</sup> ) and Collaboration .....	123
	We've Reached the End of Chapter 7 .....	129
	References.....	129
<b>8</b>	<b>Social Engineering Paradigm Remixed.....</b>	<b>131</b>
	Scrum .....	132
	Extreme Programming (XP) .....	134
	Collaborative Development Environment (CDE) .....	135
	Traditional Software Methodologies and Social Networking.....	137
	We've Reached the End of Chapter 8.....	141
	References.....	141
<b>9</b>	<b>Mobile Social Software Engineering.....</b>	<b>143</b>
	Pervasive (Ubiquitous) Computing.....	144
	Mobile Social Software Engineering.....	148
	Software Platforms.....	148
	Security.....	150
	We've Reached the End of Chapter 9 .....	151
	References.....	151
<b>10</b>	<b>Legal, Privacy, and Security Issues.....</b>	<b>153</b>
	Website Legal Issues.....	154
	Developing Your ePolicy .....	155
	Security Issues.....	156
	Web Server Security.....	161
	Protecting Mobile Devices.....	163
	We've Reached the End . . . of This Book.....	165
<b>A</b>	<b>Social Software Engineering Tools.....</b>	<b>167</b>
<b>B</b>	<b>Community of Practice Practitioner's Guide .....</b>	<b>187</b>
<b>C</b>	<b>Knowledge Discovery Techniques.....</b>	<b>255</b>
<b>D</b>	<b>Staff Competency Survey .....</b>	<b>271</b>
<b>E</b>	<b>Behavioral Competencies .....</b>	<b>277</b>

**F**    **Balanced Scorecard Metrics .....283**

**G**    **Glossary.....293**

**H**    **Brief EMMML Reference.....303**

**I**    **Computer Use Policy .....309**

**J**    **Best Practices Security Checklist ..... 311**

**K**    **User Interface Design Guide .....395**

**Index .....459**

## Chapter 1

---

# Why Social Networking?

---

The 2009 *Standish Group Chaos Report* painted a dismal picture of the state of software development. The Boston, Massachusetts, research firm surveyed 400 organizations and found a decrease in IT project success rates and an increase in IT project failure rates in a short two-year period. Only 32% of surveyed projects were considered successful (i.e., on time, on budget, and with required functionality and feature sets). Nearly one quarter of IT projects were considered failures. The rest were considered to be challenged—a euphemism for late, over budget, or implemented without the full set of promised functions and features. It should be noted, however, that many dispute the *Chaos Report* findings. What's undisputed is that a large number of projects do fail.

Many things can and do go wrong with software development efforts. McConnell (1996) neatly categorized the issues as shown in Table 1.1. Hyvari (2006) provides an updated view, as shown in Table 1.2. The names may have changed but the problems remain more or less the same. As you can see, a host of reasons can negatively impact project success and high on the list is the human element. Ewusi-Mensah (2003) states simply that, “the software development enterprise is a purely abstract and conceptual endeavor, and as such places an undue burden on all the stakeholders to collaborate with a clear vision of what is to be achieved, how it is to be achieved, and at what cost and in what time frame.”

Having the right people on a project team is certainly key to the success of a project. In a large pharmaceutical company, the lead designer walked off a very important project. Obviously, that set the team back to a large extent because no one else had enough experience to do what he did. Even if the IT staff stays put, it is always possible that a “people” issue will negatively affect a project. For example, a change in senior management may mean that the project you are working on gets canned or moved to a lower priority. A project manager working for America

**Table 1.1 Classic Software Development Project Problems**

<i>People-Related Mistakes</i>	<i>Process-Related Mistakes</i>	<i>Product-Related Mistakes</i>	<i>Technology-Related Mistakes</i>
Undermined motivation	Overly optimistic schedules	Requirement gold-plating, i.e., too many product features	Silver-bullet syndrome, i.e., latching onto new technology or method that is unproven for project
Weak personnel	Insufficient risk management	Feature creep	Overestimated savings from tools or methods
Uncontrolled problem employees	Contractor failure	Developer gold-plating, i.e., use of technology for the sake of using that technology	Switching tools in middle of project
Heroics	Insufficient planning	Push me–pull me negotiation, i.e., constantly changing schedule	Lack of automated source code control
Adding people to late project	Abandonment of planning under pressure	Research-oriented development, i.e., stretching limits of technology	
Noisy crowded offices	Wasted time before project starts, i.e., approval and budgeting processes		
Friction between developers and customers	Shortchanged upstream activities, e.g., requirements analysis, etc.		

**Table 1.1 (continued) Classic Software Development Project Problems**

Unrealistic expectations	Inadequate design		
Lack of effective project sponsorship	Short-changed quality assurance		
Lack of stakeholder buy-in	Insufficient management controls		
Lack of user input	Premature or too frequent convergence, i.e., product released too early		
Politics over substance	Omitting necessary tasks from estimates		
Wishful thinking	Planning to catch up later		
	Code-like-hell programming		

Online-Time Warner had just started an important new project when a new president was installed. The new president did what all new presidents do. He engaged in a little housecleaning. Projects—and some people—were swept away. When the dust settled, the project manager faced a whole new set of priorities and a bunch of new team members. As you can see, today's dynamically changing, and very volatile, business landscape can play havoc with software engineering efforts and going global adds an entirely new dimension to the mix. What we need, then, is a whole new paradigm of software development that places the human aspect at the center of software engineering.

## The Social Network

Social networking is a hot topic. More than 30 billion pieces of content are shared on Facebook each month and Nielsen researchers say that consumers spend more than 5½ hours on social networking sites per day. So I am sure it doesn't come as a surprise that social networking has made its way into the workplace.

Table 1.2 Success and Failure Factors

<b>Factors related to project</b>
Size and value
Having clear boundary
Urgency
Uniqueness of project activities
Density of project network (in dependencies between activities)
Project life cycle
End-user commitment
Adequate funds and resources
Realistic schedule
Clear goals and objectives
<b>Factors related to project manager and leadership</b>
Ability to delegate authority
Ability to trade off
Ability to coordinate
Perception of role and responsibilities
Effective leadership
Effective conflict resolution
Relevant past experience
Management of changes
Contract management
Situational management
Competence
Commitment
Trust
Other communication
<b>Factors related to project team members</b>
Technical background
Communication
Troubleshooting
Effective monitoring and feedback
Commitment