IBM PRESS

A Practical Guide to Distributed Scrum

Elizabeth Woodward, Steffan Surdek, and Matthew Ganis

Forewords by Ken Schwaber, Scott Ambler, Roman Pichler, and Matthew Wang

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer
Cover design: IBM Corporation

Editor-in-Chief: Karen Gettman
Marketing Manager: Kourtnaye Sturgeon
Publicist: Heather Fox
Executive Editor: Chris Guzikowski
Senior Development Editor: Chris Zahn
Managing Editor: Kristy Hart
Designer: Alan Clements
Senior Project Editor: Lori Lyons
Copy Editor: Water Crest Publishing, Inc.
Indexer: Lisa Stumpf
Compositor: Jake McFarland/Nonie Ratcliff
Proofreader: Kathy Ruiz
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com.

For sales outside the U.S., please contact:

International Sales
international@pearson.com.

Foreword by Ken Schwaber

Agility is the word of the new millennium. As the world around us grows more complex, we strive to build more complex products. These products often consist of many components that must interact precisely through sophisticated interfaces. At the same time, these products are being used in more sophisticated, critical applications, including life-critical products such as pacemakers and nano-robots, and society-critical applications such as an intelligent energy grid. In parallel with the growth of complexity, there has been a need for increased safety, predictability, risk management, and control—both of the development process itself and the resultant products. At the same time, our need to be nimble, flexible, and adaptable has increased. Enter the era of agility.

Agility first formally entered the product development arena with the publishing of the *Agile Manifesto* in 2001. As of 2008, more organizations are employing agile techniques and processes to develop and sustain complex products than those that continue to employ more traditional techniques. Of those using agile techniques, 84% of them employ an agile framework process, Scrum.

A complexity faced by almost all large organizations is distribution of teams (many teams working in different locations) and dispersed teams (team members within a team are dispersed to different locations). Among the compelling reasons for distribution and dispersion are scarcity of skills, flexibility of forming teams rapidly, and sustaining adequate workforces. One of the original reasons for distribution and dispersion that has proven to be elusive, however, was lowered costs. By the time the development infrastructure was in place, the teams and team members had familiarized themselves with each other, and integration issues were addressed, cost savings were no longer the compelling reason.

When organizations begin to use Scrum, they often run into the difficulties inherent to distributed and dispersed teams. Because Scrum uses frequent inspection and adaption of transparent artifacts to control risk and create predictability, distribution and dispersion make these

techniques difficult. For instance, if I have never worked closely with someone on my team from Asia and I am in Boston, how do I interpret what he or she means in the Daily Scrum? If we have many teams scattered throughout the world, what techniques can we use to resolve our dependencies and frequently integrate our work, so that transparency is retained? There are no optimal solutions, only best possible solutions.

I have known Elizabeth since she and a small group within IBM took on the challenge of making the IBM culture agile, and doing this with Scrum as its framework. When IBM began its path to use Scrum and other agile techniques, it arose from a grass-roots initiative. As success occurred and became visible, senior management provided needed support and leadership to realize the benefits at an enterprise level.

This book is about how to use Scrum in a very large, heterogeneous, globally distributed organization, such as IBM. This book contains many of the best practices that have emerged during IBM's transition to agile. Each practice is well stated with tips and alternatives. These techniques are not ethereal, but have emerged from hard, empirical experiences that the teams in IBM have faced.

I have found that change only happens one person at a time. If a person doesn't understand what the benefits of the new way of doing things are, both to his or her organization and to him or her personally, passive and/or outright resistance will occur—singly, then in groups, and then in political action. The authors have carefully and subtly demonstrated the advantages of agile, not by selling, but by creating insight everywhere and in everyone, and by having the natural leaders within the organization lead the change.

I encourage you to read of the changes to agility that those contributing to this book have been able to initiate, support, and sustain. Read this book and gain insights that may assist you in your efforts to gain agility in your organization.

Elizabeth told me the community would be honored if I wrote this Foreword. I am honored to write this Foreword for them, and the cunningly thoughtful change they have caused.

Ken Schwaber
Co-creator of Scrum
www.scrum.org

Foreword by Scott Ambler

If you're reading this foreword, you're probably trying to answer one or more of the following questions: "What will I learn?," "Should I spend my hard-earned money on this book?," "Will it be worth my valuable time to read it?," and "Is this a book that I'll refer to again and again?" To help you answer these questions, I thought I'd list a few user stories that I believe this book clearly fulfills.

As a reader, I want:

- A book that is well-written and understandable.
- Real-world examples that I can relate to.
- Quotes from actual people doing this in the field.
- To understand the challenges that I'll face with distributed agile development.

As someone new to agile, I want to:

- Learn the fundamentals of Scrum.
- Understand the fundamentals of agile delivery.
- Learn about what actually works in practice.
- Discover how to extend Scrum into an agile delivery process.

As an experienced agile practitioner, I want to learn:

- How to scale agile approaches for distributed teams.
- How to overcome the challenges faced by distributed teams.

- How to tailor existing agile practices to reflect the realities of distribution.
- About "new" agile practices that we might need to adopt.
- Techniques so that distributed team members can communicate effectively.
- How to extend Scrum with proven techniques from Extreme Programming, Agile Modeling, and other agile methods.
- How to address architectural issues on a distributed agile team.
- How agile teams address documentation.
- How agile teams can interact effectively with non-agile teams.

As a ScrumMaster, I want to learn how to:

- Lead a distributed agile team.
- Facilitate a distributed "Scrum of Scrums."
- Facilitate the successful initiation of a distributed agile project.
- Facilitate communication and collaboration between distributed team members.

As a product owner, I want to learn:

- How to manage a product backlog on a distributed team.
- About different categories of stakeholders whom I will need to represent.
- About techniques to understand and capture the goals of those stakeholders.
- How to manage requirements with other product owners on other sub-teams.
- What to do during an end-of-sprint review.
- How I can streamline things for the delivery team that I'm working with.

As an agile skeptic, I want to:

- See examples of how agile works in practice.
- Hear about the challenges faced by agile teams.
- Hear about where agile strategies don't work well and what to do about it.

I work with organizations around the world helping them to scale agile strategies to meet their real-world needs. Although this book is focused on providing strategies for dealing with geographical distribution, it also covers many of the issues that you'll run into with large teams: complex problem domains and complex technical domains. An important aspect of scaling agile techniques is to first recognize that there's more to scalability than dealing with large teams, something that this book clearly demonstrates.

At the risk of sounding a bit corny, I've eagerly awaited the publication of this book for some time. I've known two of the authors, Elizabeth and Matt, for several years and have had the pleasure of working with them and learning from them as a result. Along with hundreds of other IBMers, I watched this book get written and provided input where I could. The reason why I'm so excited about it is that I've wanted something to which I could refer the customers who I work with and honestly say, "yes, we know that this works because this is what we do in practice."

IBM is doing some very interesting work when it comes to scaling agile. We haven't published enough externally, in my opinion, due to a preference for actively sharing our experiences internally. This book collects many of our experiences into a coherent whole and more importantly shares them outside the IBM process ecosystem. The bottom line is that I think that you'll get a lot out of this book.

Scott W. Ambler
Chief Methodologist/Agile, IBM Rational
https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/

Foreword by Roman Pichler

My early experiences of distributed software development include attending conference calls without having access to the documents being discussed; hoping for an upgrade to get some sleep on a long-haul flight; and being surprised when I finally met the face to a voice I only knew from telephone conversations. Distributed software development is challenging, as Elizabeth Woodward, Matthew Ganis, and Steffan Surdek clearly point out in their book, *A Practical Guide to Distributed Scrum*. Things don't get easier when Scrum is applied. I remember one Scrum project I worked on where the build times grew longer and longer, as the number of teams and locations increased. On another project, we had to figure out how to run the sprint meetings with teams distributed across several times zones. And on a third project, an empowered chief product owner was missing, causing the distributed teams to follow their own goals and agendas.

I wish *A Practical Guide to Distributed Scrum* had been available then. It would have saved me plenty of time and provided invaluable help. The advice in Chapter 2 on telephone dynamics would have helped avoid the painful telephone calls; following the recommendation in Chapter 3 to organize for the lowest level of distribution could have made some of the flights unnecessary; using videoconferencing as suggested in Chapters 2 and 6 would have made communication more effective; employing continuous integration, as discussed in Chapter 7, would have mitigated the build time problem; Chapters 4, 5, 8, and 9 would have helped greatly to organize the distributed sprint meetings; and the advice in Chapter 3 would have avoided the chief product owner issue.

The book does an outstanding job at raising the reader's awareness of the problems that are likely to arise on a distributed Scrum project. It excels at providing practical tips to avoid and overcome these issues. Woodward, Ganis, and Surdek take the reader on journey through the

lifecycle of a distributed Scrum project—from creating the product vision and setting up the right project organization to preparing the sprint planning meeting and running distributed sprint review and retrospective meetings. They point out the traps along the way and explain specific practices like lookahead planning, test automation, and being aware of culture and language differences.

If you are involved in distributed Scrum projects, then this book is for you.

Roman Pichler
Author of *Agile Product Management with Scrum*

Foreword by Matthew Wang

Agile development has gained popularity in recent years. Software companies adopted agile to better respond to frequent requirement changes. When agile was introduced in the mid 90's, collocated development teams were common (that is, the whole project team was in one location). So the daily Scrum meetings could be conducted in the same place at the same time. Nowadays, a distributed development team is the norm as companies embrace global sourcing. A project even could have team members located in different continents. The challenge is how to apply the agile principles that were established in a collocated environment to a distributed development environment.

As a global integrated enterprise, IBM has been on Agile for years and has set up the agile community to drive this effort. The book *A Practical Guide to Distributed Scrum* is one of the major achievements of the community. It offers an unprecedented view on how distributed teams can also implement Scrum, the leading framework for agile software development, with tips and recommendations contributed by over 1,300 members from 30 countries in IBM. The book itself was developed in an agile way—a successful distributed agile project by itself.

IBM China Development Labs (CDL) has been embarking on the agile journey since 2005. It has accumulated wealthy knowledge and experience on agile development. Particularly, most CDL teams need to conduct product development by collaborating with worldwide teams; distributed Scrum is well adopted in CDL. For example, driven by the agile community, more than 60 teams at CDL have used Rational Team Concert, a collaborative software development environment, to assist agile development. The teams also have done tremendous innovations such as leveraging the 12-hour time difference between development and test teams to maximize agile benefits. Some of the practice have been included with this book; you can read about the details in various chapters.

The methodology of software development has been evolving continuously. Software companies will be more successful only when they can grasp the new methodology. Distributed teams will continue to be a trend in software industry due to the benefits of talent availability and cost reduction, so distributed agile development skill is critical. I highly recommend this book to software developers—you can learn a great deal from the well-structured content, which is practical and can be put into use right away.

Matthew Wang
Vice President
IBM China Development Laboratories

Preface

You know, we do this stuff all the time. We should just write a book about it!

John Sutcliffe

Agile software development is a growing trend as companies look for ways to improve quality, reduce time to deliver software to the market, and more accurately deliver software that meets the needs of their clients. At the same time, globalization, distributed software development work, and telecommuting are rapidly changing how software development teams work.

In 2007, *Dr. Dobbs Journal* reported that agile software development had successfully crossed Moore's technology-adoption chasm in that 69% of agile survey respondents indicated their organizations were doing one or more agile projects (Ambler 2007). Of the remaining respondents, 24% believed their organizations would do so within the next year. Of the 44% indicating a 90%+ success rate for agile projects, co-located respondents were 60% more successful than non-co-located.

Telecommuting is also a growing trend that increases the likelihood that even teams working in the same geographical areas will be functioning as a distributed team at least part time. In a 2008 CompTIA study, a majority of respondents (78%) reported that some employees within their organizations telecommute at least part time (CompTIA 2008). With telecommuting providing significant increases in productivity, cost savings for companies, a reduction in a company's carbon footprint, and a compensation trend for those hiring, telecommuting is a trend that is likely to continue to grow.

A number of great articles on implementing agile in the distributed enterprise environment have recently been presented at conferences, such as the Agile 2008 conference. However, only a few books have been published on distributed agile. Only one book has been published specifically on Enterprise Scrum. This book is complementary to such books in that it provides practical

tips, recommendations, and experiences that enrich the knowledge of readers of seeking information on Enterprise Scrum.

This book shares the experience of a community of 1,300 Scrum team members from across IBM business units and geographies. The diversity provides a unique perspective into a broad array of challenges specific to working in a fully distributed software development environment.

Who Should Read This Book?

The target audience for this book is members of software development organizations worldwide, across industries who work with distributed team members and in particular those who are adopting Scrum for agile development. Software developers, testers, technical writers, team leaders, managers, and others who telecommute, work with outsourced teams, or work as part of a fully-distributed team will benefit from the practical guidance provided in this book. Because the book applies to geographically distributed teams, the audience for this book is global.

This book applies not only to the large-scale software top-100 companies but also to smaller companies that perform contract work for larger, distributed organizations and to small companies that find value in allowing their software development team members to telecommute.

Managers who manage distributed teams would be interested in this book as a way to scale agile methods to benefit their organizations.

Why a Practical Guide to Distributed Scrum?

Distributed development is quickly becoming the norm rather than the exception. There are many challenges faced by new teams or teams early in their journey toward distributed development. While creating the content for the book, our aim was to share the challenges that we faced and propose solutions that teams can adapt and adopt. Our goal is not to prescribe solutions, but to present options and to have teams decide what will work best for them in their world.

In all our chapters, we first provide a baseline from the regular Scrum perspective and then we discuss the challenges distributed teams face; we then provide guidance or approaches to make that piece of Scrum smoother.

How We Wrote This Book

IBM Quality Software Engineering (QSE), an organization responsible for helping teams to improve their software development practices, facilitates several communities including the QSE Scrum Community. This book, started at the end of October 2008, would not have been possible without the contribution of its members. The effort was to be a community-developed book, with Elizabeth and Matt serving as authors. Steffan joined the brainstorming group in January 2009 and signed on as the third co-author shortly after that.

Throughout the content development process, the initial core group of contributors and members of the Scrum Community participated in numerous brainstorming sessions, presentations, and content reviews. Thomas Starz, Gregg Gibson, John Sutcliffe, Elizabeth, Matt, and Steffan served as facilitators for many of the sessions and converted many hours of recordings and notes into sentences and paragraphs. We also had the pleasure of hosting a Scrum Community presentation by Roman Pichler, author of Agile Product Management with Scrum, who kindly offered to review an early version of Chapter 3, "Starting a Scrum Project."

One of the most interesting aspects of writing this book was that everyone who participated or contributed in any way in the project was distributed. Elizabeth, Steffan, and Matt wrote a lot of the rough content for the chapters, and the participants from the QSE Scrum Community were just amazing in quickly reviewing these drafts for us and contributing their own ideas and content to help make the book better.

Starting around April 2009, we used two-week Sprints to convert the content from the brainstorming sessions into chapters on specific topics. As authors, we overcame the distance separating us (Elizabeth is in Texas, Matt is in New York, and Steffan is in Quebec) with weekly conference calls and by being very open to changes the other authors felt necessary.

It was our intention to capture experiences and helpful recommendations not just from within IBM, but also from knowledgeable Scrum Team members, coaches, and consultants *outside* of IBM. We felt very strongly that we needed to take an agile approach to developing this book. We wanted to share chapters outside of IBM as they were being developed and gather feedback and quotes that would enrich the content and be helpful to any team using Scrum in a distributed environment. To do this, we had to work to adapt the IBM Press publication and IBM Legal processes. Steve Stansel and Nicole Gallo saw the value of an Agile approach and went out of their way to work with us to make the necessary changes. Peter Santhanam and Dave Hayward provided web server access, so that we could deliver chapters through www.distributedscrum.com. The process opened a door for other authors to take an Agile approach to writing books.

Because of the level of participation from so many contributors, we decided early in the development of *A Practical Guide to Distributed Scrum* that we would donate all royalties to charitable organizations. Members of the IBM QSE Scrum Community voted on the following charities to benefit from these efforts:

- Children's Hunger
- Alzheimer's Association
- Alzheimer Society

We are thrilled to have had an opportunity to collaborate with so many experienced, thoughtful, and motivated Scrum practitioners through this effort. We hope that this book will make a difference not only for those benefitting from these charities, but for the Scrum and agile community at large.

References

Ambler, Scott. *Survey Says...Agile Has Crossed the Chasm. Dr. Dobb's Journal*. http://www.ddj.com/architect/200001986.

CompTIA. *Summary of "Trends in Telecommuting: Organizations Are Realizing Benefits and Addressing Challenges*." http://www.comptia.org/sections/research/reports/200809-TelecomSummary.aspx, 2008.

Acknowledgments

We are grateful to many people who played a significant role in the development of this book. Each author has provided personal acknowledgments, and together we recognize the community that made this book possible.

Personal Acknowledgments from Elizabeth

Stacia Broderick, a Certified Scrum Trainer, begins her ScrumMaster class with: "Think back to your favorite project. What did you enjoy most about it?" If I were to answer that question today, writing *A Practical Guide to Distributed Scrum* would be at the top of the list. It has been a real pleasure and a blessing to be able to work on this cool project with such a talented group of people.

I want to thank my handsome husband for being my partner in this great adventure of life, for sharing the beauty of each precious day with me, for being supportive, and for "removing blockers." I'd like to thank our older sons, Michael and Zachary, for listening to never-ending Scrum stories and providing continuous stakeholder feedback. I'd like to thank our youngest son, Lawson, for pulling me away to play with blocks and puzzles in support of sustainable pace. I'd also like to thank my mother for always being available for distributed meetings with me, regardless of her normal "working hours."

A few projects from the past had an influence on my passion for writing this book. I would like to thank Roger Tucker for giving me a chance to work on the Central Valley Internet Project, a project that demonstrated the value of short Sprints, continuous stakeholder feedback, and a prioritized backlog. I would like to thank Doug Stephen for the opportunity to work with his agile teams and experience sustainable hyperproductivity. I would like to thank Jim Jones for giving me the opportunity to create and lead IBM's Scrum Community. And, I would like to give Bill

Krebs, Ted Rivera, Paul Gibson, and Matt the recognition they deserve for blazing an Agile path within IBM.

There are a few people that I would like to give an extra "thanks" beyond our group acknowledgments. I would like to thank John Sutcliffe for suggesting the idea in the first place. John's passion and enthusiasm for facilitating teams permeates this book. I want to thank Ming Zhi Xie for leading the IBM Scrum Community in China, for organizing face-to-face meetings and for effectively providing insight to the Chinese culture. I thank Thomas Starz, for brainstorming and presenting the content at Scrum events in Germany, and Gregg Gibson, for spending countless hours facilitating brainstorming sessions and transcribing.

I would personally like to thank Scott Ambler for thoroughly reviewing, sharing his years of experience as an agile proponent, and debating earlier versions of this book. This book is better for his insight.

Finally, I would like to thank my friends and partners, Matt and Steffan. Matt, your enthusiasm and energy for every project you work on is contagious, and this was no exception. It is always a pleasure to work with you. Steff, I can't thank you enough. Your willingness to do what it takes to deliver and fanatical championing of StyleWriter made this project possible. Your unparalleled drive, passion for excellence, sense of humor, and hockey stories made this project a favorite!

Personal Acknowledgments from Steffan

This book has taken me through quite a journey in the past year, and writing this feels like my own little personal Academy Awards moment.

To start, I would like to thank my family, starting with my wife Annie, for understanding that "just one more night of editing" usually meant that I would probably be editing or writing stuff every night of the week. I am eternally grateful for her patience and support while working on this book. I would also like to thank my children, my son Jonathan and daughter Caroline, for always finding a way to make daddy smile, for giving me a change of pace, and for helping me see what is important in my life.

Professionally, I would like to thank and recognize the following people. Steve Teleki, my former manager, for encouraging me to do more writing and for providing constructive feedback and ideas on how to write and present effectively. Robert Begg, for your continued support and for keeping me involved in coaching teams in IBM. Yves Ferland, a former manager of mine in IBM and one of my mentors—we are less in touch, but I always appreciate my talks with you whenever we meet.

I would like to thank Matt and Elizabeth for inviting me to be part of the writing team. Matt, you are such a positive person all the time and easy to work with. It has been a pleasure working with you! Elizabeth, to say you literally turned my life upside down (in a good, fun, and exciting way) in the last year would be quite the understatement. Thank you for being the great leader you are and for keeping us all on track and focused throughout this project!

Finally, I would personally like to thank the core group of participants from our brainstorming sessions; this book is a reflection of all your contributions, and I am grateful to all of you. I have some quick thoughts for some of you, with whom I interacted with a bit more. Leslie Ekas, brainstorming with you is always interesting and fun; we could go on for hours on end. Thomas Starz, you were tireless in taking part in the brainstorming sessions, reviewing materials, and providing feedback on all the chapters. Paul Sims, I appreciated your in-depth reviews, constructive comments, and any discussions we had together.

Personal Acknowledgments from Matt

There are many people that I have crossed paths with in my personal, academic, and business life that have affected me in a variety of ways. To thank them all for their advice and council (and at times pleasant and heated debates) would be impossible, but rest assured, I am indebted to each and every one of you.

There are some people I do need to thank and acknowledge. First and foremost is my management team here at IBM: Monica Piccinini, John Rosato, and Steve Wright. They've always supported my crazy requests to participate in agile efforts across the company, external speaking engagements, or funding my (never-ending list of) conferences. Without them, I would have never succeeded. There are a number of agile zealots like myself inside IBM, and together, we've formed a large following, and interest in the topic that always amazes me. People like Paul Gorans, Michael "Max" Maximilien, Bill Krebs, and Ted Rivera deserve a great deal of thanks (and I'm sure I owe each of them a drink or two) for all their help over the years.

As things get "closer to home," I think about how grateful I am to have both Elizabeth and Steff in my business and personal life. I have never met two more committed, caring, fun, and knowledgeable people. I (and I think IBM) owes them a huge debt of thanks!

Finally, I'd like to thank my family: my wife Karen, my daughter Taylor, and my son Matthew. They put up with a lot from me. Every vacation, baseball game, soccer tournament, and quiet night at home, they find me with a pad of paper writing something. Trust me guys, in between the writing, I saw every hit, kick, and score!

Joint Acknowledgments

We wish to thank our executive reviewers—Matthew Wang, Vice President of China Development Lab; Ponani Gopalakrishnan, Vice President of India Software Development Lab; and Helen McKinstry, Director of IBM Quality Software Engineering (QSE)—for their support of this project.

We are grateful to the team at Pearson Education: Chris Guzikowski, Chris Zahn, and Raina Chrobak, for guiding us through the process and for being so responsive. We are also grateful to Steve Stansel of IBM Press and Nicole Gallo of IBM Legal for their guidance. We would also like to thank Peter Santhanam and Dave Hayward, who provided web server space to host our chapters that we delivered through www.distributedscrum.com.

We also wish to thank our reviewers for sharing their feedback: Ken Schwaber, Roman Pichler, Stephen Forte, Linda Rising, Lisa Shoop, Amr Elssamadisy, and Scott Ambler.

The IBM Scrum Community has engaged in many brainstorming sessions, round tables, and presentations to share their experiences. The following individuals have led sessions, researched, developed content, discussed methods with those outside IBM, shared their experiences, and more:

Jan Acosta
Corville Allen
Pushpa Baskaran
Donald Bell
Diane Benze
Bob Campbell
Michael Demeter
Saurabh Dua
Leslie Ekas
Mike Eversole
Gregg Gibson
Justin Gordon
Brenda Hagler
Jeff Hedglin
Greg Hendley
Jyoti Jalvi
James L. Jones
Alan June
Elizabeth Kamau
Bill Krebs
John Langlois
Mark Levison
Monica Luke
Jean-Louis Marechaux
Berne Miller
Andy Pittaway
Ivan Portilla
Hector Rosas
Bob Sager
Julio Sanchez
Peter Santhanam
Guillaume Senneville
Ryan Shillington
Paul Sims
Manmohan Singh
Maya Srihari
Thomas Starz
John Sutcliffe
Teresa Shen Swingler
Mike Thompson
Jeff Treece
Mark Wainwright
Hanhong Xue
Ming Zhi Xie

About the Authors

Elizabeth Woodward is a Senior Software Consultant with IBM Quality Software Engineering under the Corporate Headquarters Office of Innovation and Technology. She has served as the project manager or development leader on more than 100 globally-distributed projects for IBM and other development companies. Elizabeth coaches distributed software development teams to improve efficiency and effectiveness of their development practices. She has co-chaired the IBM Academy of Technology Conference on Agile Methods, teaches courses on Disciplined Agile Development, and co-leads the IBM Agile Community.

Steffan Surdek is a User Experience Lead and Agile Champion in IBM. He has worked in the software development industry for over fifteen years as a software developer, architect, project manager, and team leader. Steffan has managed and coordinated large-scale projects with teams distributed in as many as five countries—India, Egypt, Israel, China, and Canada. He coaches distributed agile teams, is a co-leader of the IBM Agile Community, and teaches Disciplined Agile Development workshops. He is an active member of the Montreal Agile Community and has written on agile methods and globally distributed development for developerWorks and *Dr. Dobbs Journal*. In his spare time, he does some writing on his website at http://www.surdek.ca.

Matthew Ganis is an IBM Senior Technical Staff Member and ibm.com site architect. Matt was the co-leader of the IBM Agile Community and was an early adopter of agile within IBM. He currently teaches Disciplined Agile Development and has published numerous articles and papers on the use of agile methods within ibm.com—both within its traditional web development and the development/support of their Second Life Island. Matt has been the co-chair and chair of the Academy of Technology's Agile Conferences for the past two years and is a Certified Scrum-Master and Practitioner. Outside of IBM, Matt serves on the editorial board of the International Journal of AGILE AND EXTREME SOFTWARE DEVELOPMENT and is a steering committee member of New York City's Agile Project Leadership Network (APLN) chapter.

This page intentionally left blank

Contributors

This book would not have been possible without the valuable contribution of members of the IBM Scrum Community as well as others in IBM. To thank them for their efforts, we created this section of the book to give them some additional recognition.

The following individuals have led sessions, researched, developed content, discussed methods with those outside IBM, shared their experiences, and more:

Jan Acosta is a member of the QSE staff and has been with IBM for ten years. She's a Disciplined Agile Workshop facilitator and enjoys consulting with teams as they work to deploy agile within their organizations. Jan facilitates the QSE Technical Series calls and leads the test-driven development community.

Corville Allen is software engineer for the WebSphere Process Server product within IBM Software Group (SWG) Application and Integration Middleware (AIM).

Pushpa Baskaran is an advisory software engineer and technical team leader working in the Distributed Software Value chain group, in the IBM BT/CIO organization. She has over 14 years of software development experience. She also has J2EE/Java development, C, C++, SQL, shell programming, and ANT scripting experience. Pushpa is experienced in the EAD4J framework and uses the framework to develop J2EE web applications. She has hands-on experience in SOA, web services and Web 2.0 development and uses the technologies to develop applications that support IBM programs. Pushpa and her team are using agile development methodology's for delivering quality software and web applications that support IBM SWG programs. She also has experience developing unit tests for DB2 and Java, using JUnit and TC4DBO, which are included with the team's automated build process.

Donald Bell is a Rational Solution Architect in IBM Rational's Global Services Account Team. He assists project teams in adopting effective software development processes and tools that balance both the client's and IBM's risks and value. He is also contributing to IBM Rational's

Measured Capability Improvement Framework (MCIF) assets so organizations can show through measures how their development processes are improving.

Diane Benze began her career as a proofreader before transitioning to graphic arts in newspapers and commercial print shops. After a five-year detour as an editor at a technical translation agency, she returned to the publishing industry. During a stint as head beta tester for a new publishing application, the software company providing the product decided that it would behoove them to have her on their side rather than on the customer side. She made the leap to software quality testing for companies providing news and advertising software. From there, it was a short hop to testing for telephony and telecommunications software companies. About eight years ago, she accepted a job as a software tester at IBM Tivoli, where she tested the NetView product. She is now testing IBM Tivoli Netcool Management software.

Bob Campbell is a development manager for IBM Cognos "Adaptive Applications Framework"—an applications development framework created by the Performance Analytics group within SWG Information Management. His development teams are distributed around the UK, Canada, and India, and they have been practicing distributed agile development for the past three to four years. Bob has been part of Scrum development teams in various roles, including team member, ScrumMaster, and product owner.

Michael Demeter is a senior software engineer and team lead for UTM Appliances with IBM Global Technology Services (GTS). He also serves as a QSE Development Top Gun.

Saurabh Dua is a project manager in SWG, responsible for the delivery of IBM Smart Business Developers' Toolkit enabling small to medium-sized customers to easily assemble and deploy applications onto Smart Business and Lotus Foundation appliances. He is a proponent for agile development and has been involved in a task force to transition all current processes to the Scrum iterative development agile model in his organization.

Leslie Ekas joined IBM through the FileNet acquisition. She left her development manager position in FileNet 2008 to become a coach in the Agile/Lean Center of Competence. As part of the SWG Strategy and Technology, IBM formed this group to help accelerate the adoption of Agile/Lean best practices throughout SWG. Leslie started practicing Scrum and applying Lean techniques to improve the effectiveness of her teams at the beginning of 2006. She promotes the culture of continuous improvement to invigorate innovation and keep IBM competitive.

Mike Eversole is a manager with SWG AIM Software responsible for Install Strategy and Development.

Gregg Gibson is a development manager responsible for embedded software development in IBM Systems and Technology Group (STG). He has been a developer and team leader on iterative development projects, and is an advocate for agile adoption within STG. He is a senior member of the Institute for Electronic and Electrical Engineers (IEEE).

Justin Gordon joined IBM through the Trigo acquisition in April 2004. Justin has contributed widely and deeply to this product, now known as InfoSphere Master Data Management (MDM) Server for PIM. In 2004, Justin led a rewrite of that product's storage layer to be vastly scalable through a binary storage format that allows reading and writing large semi-structured

hierarchical documents, like XML, without any of the parsing costs of XML. During this intense development phase, Justin pioneered the use of test-driven development, JUnit, and other agile development techniques. Justin founded the open source project, the "Dependent Object Framework," which vastly simplifies and accelerates JUnit testing with persistent dependencies. Justin has been passionately writing software for 22 years, focusing on Java for the past 10 years at several startups. Justin graduated in 1991 with an AB magna cum laude in applied mathematics from Harvard and received an MBA from UC Berkeley in 2001.

Brenda Hagler is the team lead for the IBM Human Ability and Accessibility Center Test Lab. Brenda has worked in various product development roles. Her background includes various products, such as AIX, IBM Tivoli Risk Manager, and IBM Tivoli Access Manager for Business Integration. She is a Test Top Gun for the IBM Research division. Her focus has been accessibility verification test (AVT) since she joined the HA&AC in 2004. She supports IBM development and test teams worldwide.

Jeff Hedglin is the manager of Cryptographic Development in STG.

Greg Hendley is a software engineer for FIT-ITE infrastructure development with SWG, Strategy.

Jyoti Jalvi is a lead engineer in the IBM Informix Engineering Operations team. Jyoti is responsible for implementing several key enhancements to the Informix build process. In this role, she has been a change agent in moving traditional Informix builds to a Build Management System (BuildForge), successfully designed and implemented the build infrastructure, and led the effort of build migration. In line with the organizational strategy to use agile development, Jyoti collaborated with the development organization and implemented the Continuous Integration process that provides completely automated build, testing, and regression analysis. Jyoti has 10 years of software development experience. She provides guidance on build/testing best practices and consulting on migration to BuildForge to her team as well as other organizations under Integration Management.

James L. Jones is the manager of QSE Enablement Services, under IBM Corporate Headquarters Office of Innovation and Technology. Jim is responsible for developing the consulting services and resources that will enable teams to improve their software development skills.

Alan June is an executive project manager in STG, responsible for leading the deployment of the industry-accepted practices of Lean/Agile into the z/OS software development processes. He works with ~1200 people on 60 teams working at 5 major sites and a number of local sites to use Lean/Agile development methods to deliver z/OS software. Alan has over 27 years with IBM. He acquired experience in Project and Program Management, Business Management (including First Line and Functional Management), Product Planning, Software Packaging, and Customer Service.

Elizabeth Kamau is a member of the Software Quality Assurance team for Integrated Technology Delivery, Technology Integration, and Management.

Bill "AgileBill" Krebs is founder of Agile Dimensions LLC, trainer and consultant at Davisbase LLC, and is on the board of directors at Rockcliffe University Consortium. He teaches

using face-to-face instruction methods and immersive web technology. Bill is a certified ScrumMaster and Practitioner, certified MBTI facilitator, and will receive his Certificate in Virtual Worlds this year in 2010. Bill worked as a developer, performance engineer, and consultant at five IBM Labs from 1983 to 2008. He has used and taught Agile software development methods such as Scrum and Lean since 2001 to over 1,000 engineers and managers worldwide. He has presented at Agile conferences and IBM Research, and served as the co-chair for the IBM Academy of Technology conference on Agile. Bill is a member of Agile RTP, Agile 3d, IEEE, ACM, Scrum Alliance, Agile Alliance, and PMI.

John Langlois is an IBM-certified executive project manager chartered to deliver integrated solutions on the Power Systems platform. From 1996 through 2001, John led the most successful notebook project in history: ThinkPad T series. John's website, www.projectEZ.com, is dedicated to helping project managers guide troubled projects through rough waters.

Mark Levison is a founding partner and consultant with The Agile Consortium, an Agile and Lean consulting company that focuses on helping its customers to deliver working software every two weeks. Mark has been an agile practitioner since 2001, introducing agile methods one practice at a time inside a small team. In the past three years, as an employee of Cognos and IBM, he's been responsible for introducing Scrum to the organization and coaching a number of teams. He also publishes a blog—Notes from a Tool User.

Monica Luke is the test automation architect for System Verification Test for the Rational Jazz products including Rational Team Concert. Monica has 15 years experience doing test automation, the last six of those with IBM Rational.

Jean-Louis Marechaux is a software engineer for Rational at IBM Canada Lab. His main areas of expertise are software architecture, design, model-driven development, and Java Enterprise Edition (Java EE) technologies. He has been applying agile principles with collocated and distributed teams for the past three to four years, both for consulting services and for internal IBM projects.

Berne Miller, PMP, has more than 25 years experience in project management. Certified as a Project Management Professional (PMP) in 1989, Berne has managed activities as large as a ten-year, billion-dollar-plus ship design and construction effort for the U.S. Coast Guard, and as small as a multi-week software modification effort for his current employer. In between, Berne has managed projects in communications, education management, electrical generation and distribution, financial management, natural resources extraction, facilities construction, and regional economic development. Other assignments have included heading the technical services division of a government systems acquisition organization, a stint as a corporate-level strategic planner, postings in executive ranks, and CEO of Southeast Conference (in Alaska). For the last eight years, Berne has served as program manager for the System Availability and Performance Management (SAPM) Project Development Team (PDT), where part of his role is to coach, advise, and mentor project managers for projects in the SAPM portfolio. Berne's particular interest in the agile rollout within SAPM is definition and collection of executive-level project status metrics.

In his spare time, Berne teaches modules in the Austin PMI Chapter PMP exam preparation course. He is also designated a Certified Professional Logistician (CPL).

Andy Pittaway has worked in Information Technology for 27 years, of which the past 11 years have been in various client-facing leadership roles with IBM Global Business Services. He is an executive project manager working currently on assignment in Italy. He is passionate about applying accelerated techniques and driving real business value from the effective and efficient use of these. He has set up and led development centers using Accelerated Solution Delivery (ASD). He has also led Speed to Market initiatives, which focus on the application of accelerators alongside the removal of blockages and implementation of techniques to ensure high levels of business orientation in technical teams. One of the areas that Andy has also been focused upon is the behavioral and cultural aspects that teams need to address in order to make accelerated techniques work.

Ivan Portilla is a certified consulting IT specialist with Business Consulting Services (BCS) at IBM. With almost 25 years of experience, Mr. Portilla is an industry thought leader, speaker, and coach in the areas of Agile Software Development, WebSphere Portal, and Web 2.0 technologies.

Hector Rosas is a systems software test architect with STG.

Bob Sager is with SWG, AIM Software. He is a development manager for WebSphere sMash and Project Zero.

Julio Sanchez is a SUN certified Java programmer. He has over eight years experience in device driver development for Point of Sale (POS) systems and device I/O. Julio is responsible for leading the JavaPOS product. That project used test-driven development to increase software quality by reducing the defect rate within the JavaPOS device driver software. Julio had gained experience by creating unit tests for Java using JUnit, ant, and Code Coverage tools. He is one of the promoters of the Agile and the eXtreme Programming software model within IBM Guadalajara, Mexico.

Peter Santhanam is the senior manager of the Software Engineering department in IBM Research. His current portfolio covers tools and methodology for the end-to-end software life cycle activities. His personal interests include holistic requirements capture, collaborative software development, automated test generation, software metrics, and process improvement. He has published over fifty technical papers in journals and conferences. Dr. Santhanam is a member of the ACM and a senior member of the IEEE.

Guillaume Senneville joined IBM in 2008 through the Cognos acquisition. Working as part of the SWG Information Management Business Intelligence and Performance Management team, he is a development manager leading a team that's actively using Scrum and other agile practices. Guillaume is currently co-leading the Cognos Agile Community, which is set up to complement other IBM agile communities for more local issues. He is an advocate of Scrum best practices and is interested in related topics like Lean Software Development. He has been a Certified ScrumMaster since 2007.

Ryan Shillington is the development manager of the WebSphere Business Services Fabric at IBM. He has worked on and led a few different agile teams, both distributed and completely local. When not pretending to run big blue, he moonlights as the co-owner of both Used Is Better and Austin Star Map.

Paul Sims joined IBM in 1984. Past projects include Series/1-based hardware for the Prodigy videotext service (a joint venture of IBM, Sears, and CBS Records), ISDN basic-rate adapters for 9370 and AS/400 systems, ISDN Q.921 microcode, Communications Utilities for OS/400, CD Showcase CDs to help sell AIX and OS/400 application development tools, IBM Distributed Debugger development manager, ADTC operations manager, WebSphere Commerce system test, and continual test improvement lead. Paul is now a software engineer in the Software Group Scenario Analysis Lab and a Disciplined Agile Development Workshop facilitator.

Manmohan Singh works as a development manager of WebSphere Business Services Fabric and manages the team in Mumbai. He has close to 11 years of experience in the IT industry across different geographies. He is an agile enthusiast and has been contributing to Agile Conferences of Academy of Technology. He is a six sigma green belt certified professional and keen on learning and contributing toward software development process improvements.

Maya Srihari, a member of the QSE staff, leads QSE deployment initiatives in the India software lab. She is responsible for enabling teams in the lab to adopt QSE recommended practices for improvement and establishing local communities and events where engineers can share their best practices and learning. Maya has been conducting the two-day Disciplined Agile Development workshops in the India lab and is part of the lab's Agile focus group, which has been set up to drive agile adoption. She has experience in leading quality initiatives in India and providing process consultancy to other IBM units in AP and Europe.

Thomas Starz is a software developer and agile coach with IBM SWG and works as part of Tivoli Service Automation Manager team. Thomas was among the early adopters of agile methods in his organization. He gathered his first agile experiences through practicing with student teams before he introduced agile practices in product development. He is a Certified ScrumMaster and Practitioner and an active member of the IBM Scrum Community. He regularly teaches an internal IBM Disciplined Agile Development class and works as a mentor and coach for several teams and individuals in various locations.

John Sutcliffe is a software development manager and Certified ScrumMaster working with distributed teams at IBM, in the Information Management group working for the Cognos Software Group. John has been deeply involved with the Cognos team in its conversion to Agile practices and their integration into IBM. John recently served a term as a member of the QSE Scrum Community Leadership Team. John was recently made a QSE Fellow for his contributions to the QSE team. John has spoken on facilitation skills and their importance at several IBM internal conferences and virtual events.

Teresa Shen Swingler is an advisory software engineer focusing on user interface design for storage management. She has worked in a distributed agile environment with IBM Director as part of STG.

Michael Thompson is a staff software engineer for WebSphere Application Server's Security team. He has a background in open source, Linux, and C/C++ development, and currently develops in Java with JUnit. He is a QSE TopGun, agile coach, and TDD advocate for the WebSphere organization, and has taught IBM Agile and TDD/unit test classes. With a passion for improving development processes and practices, Michael has embraced agile and champions unit test and static analysis within as part of development and continuous test activities. Michael is a graduate of Clarkson University with a BS in Computer Science.

Jeff Treece is a software development manager for IBM Software Group Information Management.

Mark Wainwright is a software engineer and coach on the Agile/Lean Center of Competence team where he uses his experience of leading agile development teams to help and encourage other SWG teams adopt agile practices. Over a period of 20 years, he has worked in services, sales, and the software group as a programmer, project manager, pre-sales technical support, architect, and integration test architect. He has been promoting agile development since 2006.

Hanhong Xue is an IBM senior software engineer leading a software development team in communication protocols for high-performance computing in IBM Systems and Technology Group. He has been practicing agile with a globally distributed team and within a waterfall-oriented environment since 2006.

Ming Zhi Xie is a CDL Testing Center (Testing as a Service) technical leader, an agile coach, and an agile practitioner. Ming Zhi frequently leads agile sessions and workshops at the IBM China Lab. He has published a series of papers on agile and agile testing via conference and IBM internal and external websites. He is also an agile consultant serving IBM China internal and external customers in dealing with agile deployment and agile testing consulting services.

CHAPTER 6

Distributed Daily Scrum Meetings

Many attempts to communicate are nullified by saying too much.

Servant Leadership, Robert Greenleaf

Once the Scrum Team has completed Sprint Planning, they will begin working together to complete user stories and meet the Sprint Goal. The Daily Scrum, shown in Figure 6.1, provides just-in-time planning and serves as a heartbeat for the Sprint. It is the time each day when the Scrum Team gets together for just 15 minutes and each team member responds to three simple questions (Schwaber 2002):

- What did you do yesterday?
- What are you going to do today?
- Do you have any blockers?

Although the idea is simple, keeping everyone focused on just the three questions can be a Distributed Daily Scrum Meetings significant challenge. Humorous videos on YouTube show just how bad a collocated Daily Scrum can be (ScrumMasters 2, 2006). Distributed communication makes the Daily Scrum even more challenging because communication through teleconference or some other channel is not as effective as being face-to-face. Many of subtleties of facial expression and body language are missing when holding meetings through teleconferences and to a lesser degree with video conferencing.

The combination of inefficient communication channels coupled with a meeting of team members located in different countries can cause the Daily Scrum heartbeat to become irregular and impact the health of the team.

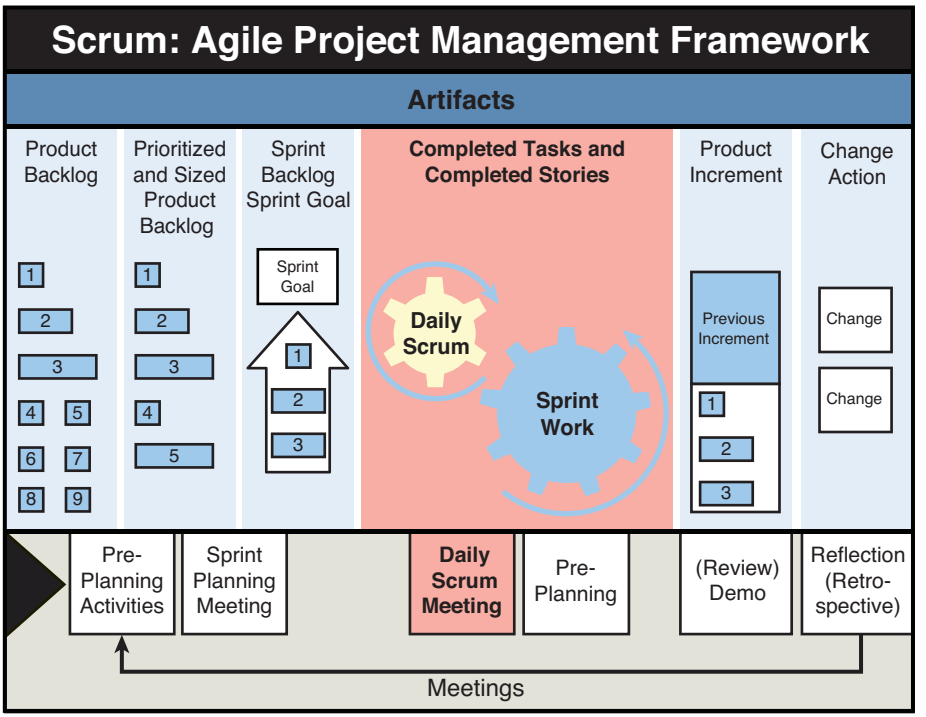

Figure 6.1 The Daily Scrum meeting

Ming Zhi Xie, an agile testing specialist with IBM Software Group's lab in Beijing, China, shared the early experience of his team with Scrum:

> *We would have liked to take part in the Daily Scrums, but with our teams 12 hours apart, it was difficult to schedule. It would have meant team members in China would be on the phone at 10 o'clock at night or team members in North Carolina would be on the phone at 3 o'clock in the morning. Neither was a good long-term solution. Separating into two local teams was unrealistic because of skills. We clearly needed an alternative.*

This chapter talks about how to answer the three questions and provides some tested techniques for conducting effective Daily Scrum meetings with distributed teams (Schwaber 2002).

Using the Three Questions Effectively

To make the Daily Scrum useful to everyone, it is important that team members understand the purpose behind the three questions they need to answer in the meeting. It is also important they provide responses that bring value to the rest of the team.

Answering the Three Questions

Here is an example where John, a fictional team member, answers the three questions of the Daily Scrum meeting:

> *Yesterday: I started writing the code to load information into the database.*
>
> *Today: I'm going to handle some of the data issues that cropped up yesterday.*
>
> *Blockers: I have no blockers.*

On the surface, it appears John is answering the Daily Scrum questions, but how much value are these answers providing to the rest of the team. His responses are brief and resemble someone providing a status report, but the Daily Scrum meeting is not a status meeting. It is an opportunity for a team to get together to find ways to help one another and to make sure the team as a whole is able to continue making progress.

Before the Daily Scrum meeting, everyone needs to examine how their current work impacts other team members or identify who can help them resolve an issue. During the meeting, they need to communicate any relevant information to the rest of the team.

KEY POINT Team members should communicate information that brings value to others on the team. They should also try to identify team members that can help them resolve their issues.

Here is a better response where John provides the same basic information but adds some more information to provide added value to the team:

> *Yesterday: I started writing the code to load user information into the database. I've discovered that some of the international addresses in the sample data provided by Company ABC do not meet the format we had expected. Mary, I think you are going to run into the same problem with the code to import the data into the label printing module. We can get together offline to discuss.*
>
> *Today: I'm going to rewrite the module based on the sample data.*
>
> *Blockers: It looks like there is a problem with the constraints we put on the tables, and we may need to update the design.*

The greatest value of the Daily Scrum meeting is the communication between team members.

Coordinating the Team on a Daily Basis

The Daily Scrum meeting allows team members to communicate, coordinate their efforts, and possibly revise their plans daily based on the answers of the others. Using our preceding example, John clearly states Mary may run into the same problem that he faced when working with the sample

data. This may change Mary's plans for the day, because now she needs to get with John to discuss this. The plans of another team member may also change to address the blocker that he identified.

KEY POINT Priorities can change daily. The Daily Scrum meeting provides a daily synchronization point for the team and allows them to revise their plans regularly.

Committing to the Team

The Daily Scrum meeting is where team members make a verbal commitment to the team. When they state what they are going to do today, they are making a verbal commitment to the rest of the team. The next day, when they state what they did yesterday, it is an opportunity for the rest of the team to confirm they met their commitments.

The Daily Scrum creates peer pressure on two fronts, as follows:

- To complete work that is blocking the progress of other team members.
- To create a sense of accountability within the team, in the case where team members do not deliver on the tasks they committed to the previous day.

The Daily Scrum keeps team members accountable as they will eventually need to justify themselves to the rest of the team if they continually do not deliver on their commitments.

KEY POINT Team members are making a verbal commitment to their team when they state what they are going to do today.

Verifying Progress

The two first questions allow the ScrumMaster to test the focus of the team. The team always needs to be working on stories coming out of the product backlog.

The ScrumMaster also wants to know if the team is completing the work they planned to complete during the Sprint. If estimates are expanding or new tasks come up, there will be an impact on the burndown chart. The ScrumMaster should question team members that are doing work that is not helping the team make progress toward the Sprint goals.

KEY POINT Tasks not opening and closing regularly are an early sign the team may be going off track.

When team members are not showing regular progress on their tasks for the Sprint, it can be a sign of outside distractions having an impact on their work for the Sprint. These may be meetings, task assignments for work outside the Sprint, or requests from others. It is important

for team members to identify these distractions in the Daily Scrum so the ScrumMaster can help reduce or remove them. Steffan Surdek shares this story:

> *Once, I was working on a team with tight deadlines, and some of the team members that were on the critical path were regularly getting requests from another group to provide them with information. When I learned of these interruptions, I directed these team members to redirect the next question my way before taking any action on it. After they forwarded me the next question, I replied to the other team and let them know they were interfering with the deliverables of these team members. I also told them they should come through me in the future for such requests and I would redirect their queries to someone who would be available to help them.*

KEY POINT Team members not showing regular progress may be facing outside distractions the ScrumMaster should reduce or remove.

There is also the possibility that team tmembers who are new to Scrum are not used to depending on their team to help them resolve problems. Smart technical people can feel intense pressure to solve problems on their own and not to depend on others. Admitting any kind of difficulty can make them very uncomfortable. When the Scrum Team notices that a member is experiencing problems, they can encourage that person to work with others to get the help needed.

Resolving Blockers

The Daily Scrum allows team members to identify blockers they are facing that are preventing them from making progress on their work for the Sprint. Sutherland states there are different kinds of blockers a team may face (Sutherland 2006), as follows:

- *Software or technology not showing tup at the right time. These may be dependencies across teams or groups. The ScrumMaster or the team may not be able to resolve this. The management team may need to step in.*
- *Meetings irrelevant to the Scrum Team. Here again, the ScrumMaster may not be able to fix this, but it is the ScrumMaster's role to recognize it and bring it to the attention of the management team.*
- *Hard technical problems often slow things down. The team can often fix this, but at times, management must bring in other resources.*

Blockers may create new issues that may in turn become new tasks or user stories in the backlog. These may force the team to change the current plan.

Sutherland also states that it is important for the ScrumMaster to create a list of blockers and assign them to the team or to managers. A major responsibility of the ScrumMaster is to manage, prioritize, and assure this impediment backlog is burning down. The Scrum Team should

expect management to help work the impediment backlog. Removing bottlenecks is the fastest way to improve productivity.

KEY POINT The ScrumMaster should create a list of blockers and assign them to team members or managers. The ScrumMaster should also ensure the team is burning through the blocker list.

Daily Scrum Logistics

Scrum encourages collocated teams to meet in the same room at the same time every day for the Daily Scrum. To adapt this meeting to work in a distributed environment, the first step is for the team to decide how to best conduct it.

The distribution level will have an impact on how to conduct the Daily Scrum meeting. Conducting the Daily Scrums when team members are in the same time zone and speak the same language is much simpler than for a team with members spread in multiple countries and time zones, having many different languages and cultures.

Distributed teams with no overlapping work hours have the biggest challenge because they are scheduling a 15-minute meeting at a time outside the normal working hours of some team members.

Ways of Communicating During the Daily Scrum

Teams can use different ways to communicate during their Daily Scrum meetings, from face-to-face meetings to using an instant messaging tool. Teams need to select the methods that best suit their needs as well as their distribution level.

Face-to-Face Meeting

We recommend face-to-face Daily Scrum meetings for collocated teams. They foster stronger communication between team members, put pressure on team members to deliver on their commitments to their team, and encourage the Scrum Team to self-organize.

When a Scrum Team meets in the same location every day, it creates a routine for the team members as well as accountability for the meeting to start at the scheduled time. Manmohan Singh, a ScrumMaster in IBM, shares the following story about how the Daily Scrum meeting evolved in his Scrum Team:

> *In the early days of our Daily Scrum meetings, I often had to remind team members to come to the meeting. There was lot of inertia in the team because we were transitioning from traditional waterfall to agile. After a few months, the meeting became part of the daily routine of the team because team members found value in the meeting. I believe conducting the Daily Scrum meetings at a fixed time every day instilled the habit of attending the Scrum in the team, and today the team meets daily at the scheduled time*

in the conference room. When I am absent that day, a senior developer or QA member starts the meeting, takes the Scrum notes, and then communicates the blockers and issues to the entire distributed team. This is a clear signal of my team starting to turn into a self-organizing team.

I find the real value of the Daily Scrum meeting is the verbal communication that occurs between team members. Often, when a team member verbally updates the whole team on his tasks, dependencies, and issues, other participants have new information related to the tasks and everyone benefits from this. We also have cases where a team member has a dependency or an issue and someone else on the team immediately has the information on how to address the issue. This is a great way ffor us to unblock the team quickly.

Table 6.1 summarizes the pros and cons of the Face-to-Face Meeting approach.

Table 6.1 Summary of Pros and Cons for the Face-to-Face Meeting Approach

Pros	Cons
• Highest collaboration level. • Richest communication level. • No loss of nonverbal communication. • Promotes team self-organization. • Whole team takes part in each day.	• Requires a collocated team. A distributed team with several team members in one location can meet face-to-face and use a teleconference call to include others in the Daily Scrum.

KEY POINT Having face-to-face Daily Scrum meetings gives the team the highest collaboration level possible.

Teleconference Meeting

When a face-to-face meeting is not possible for fthe entire Scrum Team, members can use a teleconference to call into the Daily Scrum. This method works well for distributed teams and for collocated part-time teams.

Distributed teams with overlapping work hours should find a time during the overlapping hours to meet every day at the same time and on the same teleconference number for the Daily Scrum. Collocated part-time teams may also find this approach useful as a backup to the face-to-face Daily Scrum. If a collocated team member is out of the office visiting a client site, at a conference, or working from home, he or she can dial into the Daily Scrum. It can be helpful to the Scrum Team to use the same teleconference number each day. This creates a routine for the team

and makes it easier for team members to dial into the Daily Scrum. Sirsidynix shows teams can successfully combine the Daily Scrum of a collocated and a remote Scrum Team (Sutherland, Viktorov, Blount, and Puntikov 2007).

Daily Scrums can be harder to facilitate in a teleconference call format than a face-to-face format; the "Tips for Distributed Daily Scrums" section, later in this chapter, provides some additional guidance to help make the meeting run more smoothly.

KEY POINT Distributed teams with overlapping work hours should use a teleconference call to the same phone number every day to hold their Daily Scrum meetings.

Table 6.2 shows the pros and cons of this approach.

Table 6.2 Summary of Pros and Cons for the Teleconference Meeting Approach

Pros	Cons
• Ideal for distributed teams with overlapping hours. • Good backup for collocated teams when team members are working remotely. • Allows team members to interact directly. • Whole team takes part in each day. • Team can discuss blockers and remove them immediately.	• Loss of nonverbal communication. • Hard to keep the people on the phone engaged.

Videoconference Meeting

An alternative to a teleconference is to do a videoconference instead. The main advantage of this approach is that team members get to see one another, so there is less nonverbal communication loss. The main challenge of this approach is how to be able to see everyone depending on the number of participants in the Daily Scrum.

This approach needs added hardware as each location joining the videoconference will need a webcam. To be able to conference multiple video streams at the same time, the team may also need extra software. There may be software and bandwidth limitations as well, which could cause problems with the video feed.

The other challenge is when there are multiple participants in one of the video streams—where should the focus of the webcam be? When only focusing on the current speaker, the remote participants will lose any nonverbal reactions of other participants at that location. When focusing on a larger group at a location, it may be difficult to see everyone.

Table 6.3 shows the pros and cons of this approach.

Table 6.3 Summary of Pros and Cons for the Videoconference Approach

Pros	Cons
• Potentially richer communication experience than a teleconference. • Allows team members to interact directly.	• Can be challenging to see everyone. • Needs added hardware and software. • Bandwidth or software limitations may affect the video streams.

Group Instant Messaging Approach

Another way to communicate during the Daily Scrum is to use an instant messaging tool with a group chat feature. At a regular time every day, the ScrumMaster invites all team members to a group chat to have them answer the three Daily Scrum questions.

This approach works with collocated or collocated part-time teams as well as distributed teams with overlapping work hours. The benefit is that by creating a transcript of the chat session, the ScrumMaster can send a set of notes to all team members by email or post on a wiki. These notes can help keep track of commitments individuals are making to the team and provide a way for members who are unavoidably absent from a Daily Scrum to review the tasks, commitments, and blockers of the Scrum Team.

Another benefit of using a group instant messaging session is that team members can type as much information as they want in the chat session. The approach allows team members not fluent in the language to prepare their text ahead of time and paste it into the group chat.

Doing the Daily Scrum this way can be chaotic depending on how the ScrumMaster conducts the meeting. There are two ways to approach it:

- All team members type in their answers to the three questions and post them in the chat whenever they are ready. The ScrumMaster sorts through stuff the answers each member provides and asks clarifying questions. This approach can make the discussions that happen in the chat difficult to follow because other team members may be posting their comments during a discussion.
- Conduct the chat session like a teleconference and have the ScrumMaster go around each participant in the chat and ask them for their answers to the Daily Scrum questions. When other team members have questions for the current participant, they can ask them right away or can wait until it is their own turn to speak. At the beginning of the meeting, the ScrumMaster should ask the people to queue up their answers to the three questions and wait for their turn before sending it out the team.

When using either of these methods, it is good practice for team members to prepare the answers to the questions five minutes before the Daily Scrum meeting. This allows team members to pay closer attention to what other team members are saying in the chat session.

Using a group instant messaging session to conduct the Daily Scrum also has some challenges. Because team members are not face-to-face in the same location, there is a loss of nonverbal communication as well as subtle verbal cues, such as tone of voice or inflections. It becomes more important for team members to ask for clarifications when there are doubts.

The Daily Scrum works best when it occurs at the same time every day as this helps create a routine the team can get into. One of the traps of the Group Instant Messaging approach is that it can fall solely on the shoulders of the ScrumMaster to launch the group chat session at the chosen time. To work around this problem, the team can name two or three people to serve as initiators for the instant messaging session and to coordinate among themselves to make sure the meeting always starts on time. Before taking part in a group chat, people typically wait for a meeting invitation to arrive before engaging; if it does not appear, they will focus on their work instead and forget about the meeting.

Table 6.4 summarizes the pros and cons of this approach.

Table 6.4 Summary of Pros and Cons for Group Instant Messaging Group Approach

Pros	Cons
• Whole team takes part in each day. • Team can discuss blockers and remove them immediately. • A transcript of the chat can easily become a set of notes for the meeting. • May be easier for non-language speakers to write their thoughts instead of speaking.	• The team loses both face-to-face and verbal communication. Unlike teleconference interaction, the Scrum Team cannot hear voice inflections. • No guarantee the Scrum Team is paying attention to the chat session. • Full accountability for the meeting occurring at the same time every day is fully on the ScrumMaster. • Can be chaotic depending on how the team conducts the meeting.

Approaches to Handling Time Zone Issues

Teams can use four different methods to deal with distributed Daily Scrums where the team has members with no overlap in their work hours, as follows:

- Daily Scrums through documentation
- Liaison approach
- Alternating meeting times
- Share the pain

Each solution has pros and cons the ScrumMaster and team should consider in deciding how to conduct their Daily Scrum meetings.

Daily Scrums Through Documentation

Perhaps the least effective way of handling distributed Daily Scrums with no overlap in the workday is scheduling the Daily Scrum during a time when *most*—but not all—team members can meet. With this approach, team members who cannot meet as scheduled document their answers to the three questions in a wiki, through email, or in a document the team can access.

Figure 6.2 shows an extreme example of a team distributed in locations from San Jose, CA in the United States to Beijing, China. If the team adopts a single Daily Scrum meeting that is held at 9:00 AM in the morning in Toronto, Canada, the meeting will likely be outside standard working hours for the team members in San Jose, Bangalore, and Beijing. Such a team might try to compensate by having all team members document their answers to the three questions.

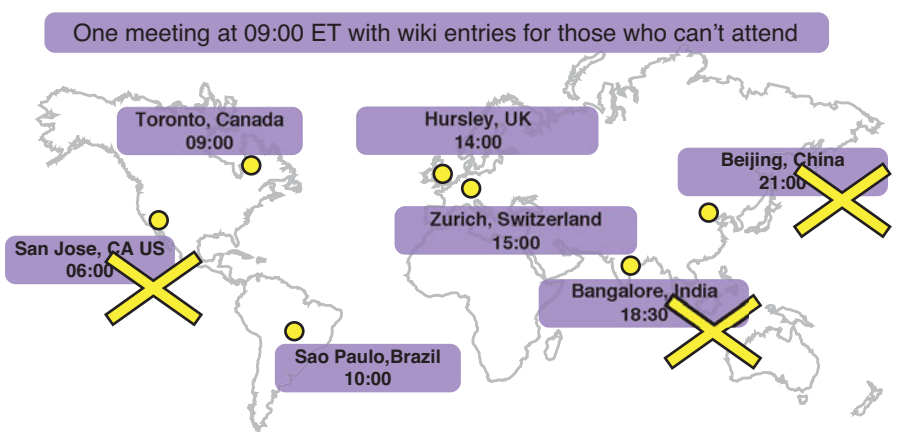

Figure 6.2 Single meeting time that works for both members

The primary advantage to this approach is that it can be better for sustainable pace. And it inconveniences no one since nobody is meeting outside their normal working hours.

However, because this approach is not interactive, those who cannot attend do not have the opportunity to hear directly from other team members or to get quick comments from them on their answers to the three questions. Also, this may cause members who do not take part in the Daily Scrum to feel less a part of the team or to feel the other members of the Scrum Team do not value their contribution as much. It is important that all members of the Scrum Team work together to make compromises to show they value everyone equally.

The lack of direct communication may also lead to some miscommunication when the documentation is not clear and the teams need some clarifications to understand. As an example, consider the statement: "We think we should meet biweekly." Does this mean the team would like to meet twice each week or once every two weeks? And consider the statement: "We should write the code in a way that is unlockable." Does this mean the programming language used should be unlockable or that the code should prevent locking?

Also, team members who do not verbally commit may not feel the same commitment to deliver. The Daily Scrum is a verbal commitment in front of peers that helps to put pressure on the team to meet their commitments.

Table 6.5 summarizes the pros and cons of the Single Meeting Time approach.

Table 6.5 Summary of Pros and Cons for Using a Single Meeting Time Approach

Pros	Cons
• Better for sustainable pace. • Nobody is inconvenienced by having to attend the Daily Scrum at a set time.	• Loss of information because of indirect communication. • This approach is not interactive and does not allow for questions. • Adversely impacts "whole team" experience since team members are not interacting. • Lessens accountability through peer pressure.

KEY POINT When having Daily Scrums with some members typing in their responses to the questions, it is important to have the ScrumMaster express their written comments out loud. The team should address their issues quickly, just as the blockers of those attending the meeting in person.

The Liaison Approach

Another approach to conducting Daily Scrums with distributed teams with no overlapping work hours is to conduct two different Daily Scrums and to have a liaison attend both. Each meeting is at a time that is convenient for half of the members of the Scrum Team. The liaison, who is commonly the ScrumMaster, verbally shares the information from the other team with the team they are meeting with.

Figure 6.3 shows an example of a team that holds two different Daily Scrum meetings each day. The first meeting is at a time convenient for the half of the team in the more western time zones. The second meeting is at a time convenient for the other half of the team in the more eastern time zones. The ScrumMaster serves as a liaison, attending both meetings and verbally presenting the notes from the half of the team that is not present.

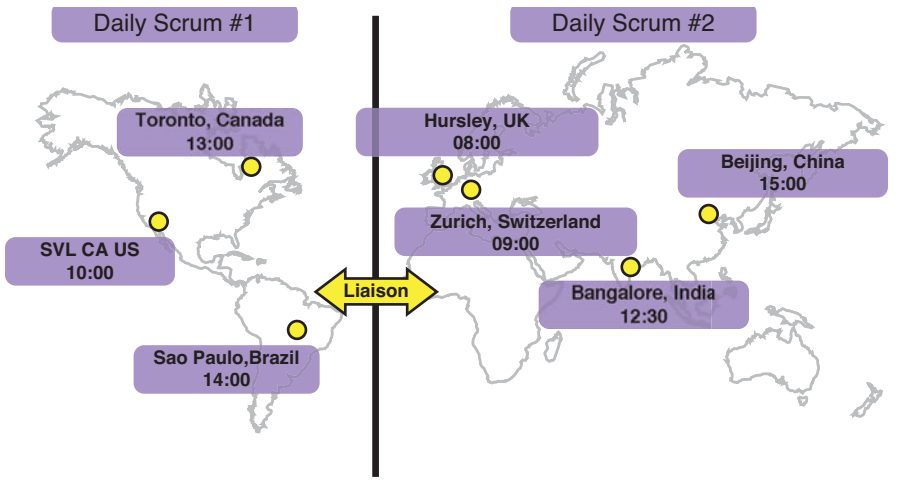

Figure 6.3 Liaison approach using two different Daily Scrum meetings

The advantage to this approach is that it helps with sustainable pace. No one other than the liaison is meeting outside their normal working hours. This can also help the whole team to have visibility into each others' efforts, though the team cannot address questions or blockers brought by team members who are not present immediately.

Pushpa Baskaran, a technical leader working at IBM, shares her experience using the Liaison approach:

> *I led a team of developers with subteams in China and the United States. It was impractical to expect all team members to be available for all meetings, and it was challenging to find a time slot that worked for everyone. So, I had two weekly scrum calls, one for the U.S. time zone, where Chinese developers were optional, and another for the Chinese time zone, where the American developers were optional. This approach allowed flexibility for team members and only the ScrumMaster (myself) needed to manage my work hours.*
>
> *When the China and North America team members were co-developing a task, one of them made the extra effort to work outside their regular hours. As the team gained more experience working in a globally distributed team environment, assigning and separating work modules, working outside regular hours reduced significantly.*

Possibly the worst side of this approach is that it is not the ideal method of communication. This can be a bit like a childhood game called "telephone," where one person tells a story to a second person, the second tells it to a third person, and so on. At the end of the chain, the story told is rarely the same as the story told by the original storyteller. The story told by the liaison will be similar, but not the same as the story told by the team member. To deal with this issue, some

teams have each group record their Daily Scrum responses. The ScrumMaster can then play back the responses from one half of the team when meeting with the other half of the team.

The team can also splinter into two factions that would likely negatively impact the ability of the team to work together as a whole team.

Another consideration in taking this approach is that, unless there is a rotation in the role, there can be an impact to the work-life balance of the liaison.

Table 6.6 summarizes the pros and cons of the Liaison approach.

Table 6.6 Summary of Pros and Cons for the Liaison Approach

Pros	Cons
• Better for sustainable pace. • Some degree of visibility into each of the tasks of each team member. • Richer communication medium than through documentation. Allows for questions.	• Liaison might present the wrong information. • Possible splintering of team into factions. • Negative impact on work-life balance of the liaison. • Negative impact on "whole team" view.

Some teams improve the Liaison approach by having the whole team meet for the Daily Scrum at least occasionally. Gregg Gibson, of IBM System and Technology Group's Management Module Firmware organization, reports many teams within his organization tried both the Liaison approach and the Sharing the Pain approach described later in this chapter:

> *We are finding that even with the Liaison approach, it is helpful to get the entire team in a meeting at least occasionally. Otherwise, the team devolves into multiple teams that don't always work together effectively.*

KEY POINT When using the Liaison approach, consider rotating the liaison responsibilities to reduce the burden on any one person.

Alternating Meeting Times

A third approach to scheduling Daily Scrums where members of the team have no overlap in work hours is to alternate meeting times. With this approach, the team holds one Daily Scrum during the normal workday for part of the Scrum Team and holds the other Daily Scrum during the normal work hours of the other part of the Scrum Team.

Ling, a Scrum Team member in Beijing, worked on a Scrum Team with members in Austin, TX and Raleigh, NC:

> *Our team was meeting daily at 12:00 PM Eastern Time. For me, in Beijing, it was after midnight. I often could not attend the Daily Scrums. The team later decided to switch to alternating meeting teams so I could attend at least part of the time in person.*

Figure 6.4 shows a team that alternates their meetings between two different times. One meeting is most convenient for team members in Toronto, Sao Paulo, and Hursley. The other is more convenient for team members in Hursley, Zurich, Bangalore, and Beijing.

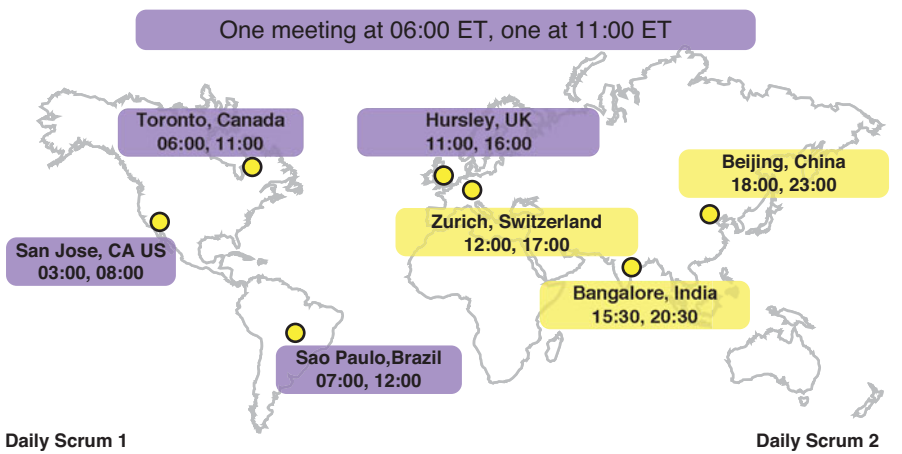

Figure 6.4 Alternating Meeting Times approach

The advantage of this approach is everyone has an opportunity to attend during their normal working hours at least every other day. Unlike the approach of having a fixed time always outside working hours for some team members, every team member has an opportunity to attend at least part of the time.

Figure 6.4 shows the given schedule of alternating meeting times that would possibly allow team members in Toronto, Zurich, and Hursley to meet daily comfortably, while other members may only join in every other day. Ideally, team members will be comfortable meeting outside traditional working hours. That can allow the whole team to meet daily, rather than every other day.

When the whole team does not meet daily, the team may not get the full benefit of the Daily Scrum.

KEY POINT When meeting at alternating times, have teams members who cannot take part in the meeting log their responses to the three questions. At the meeting, have the ScrumMaster or a team member read the written responses from absent team members.

When using the Alternating Meeting Times approach, the team should check with everyone to make sure that they have the ability to call into the meeting outside their working hours.

Elizabeth Kumau, a test expert with IBM Test Services, gives this experience:

Some of my team members in India have to stay in the office or go to the office at night to attend meetings scheduled outside their normal working hours. Out of four team members, only one has home connection access.

In this example, some team members would not be able to attend some meetings when using the Alternating Meeting Times approach.

Table 6.7 summarizes the pros and cons of the Alternating Meeting Times approach.

Table 6.7 Summary of Pros and Cons for the Alternating Meeting Times Approach

Pros	Cons
• Everyone has a chance to attend the meeting. • Shares the pain across team members.	• Loss of information from team members who do not show up because the time is not good for them.

Sharing the Pain

A fourth approach—the one that most closely aligns with the spirit of co-located Scrum—is to have the team members *share the pain*. With this approach, the team works together to share the pain associated with being part of a distributed team. They select a time to meet that is best for the team as a whole.

Although sharing the pain may not sound like a great solution for the team, flexible work hours can be a benefit both to the employer and to the Scrum Team members. Matt, a ScrumMaster with IBM Sales and Distribution, describes the advantage of working flexible hours:

I had to be on a 15-minute Daily Scrum at 7:00 PM, after my normal workday. However, I also had the flexibility to attend my daughter's father-and-daughter lunch at school, which was on the other side of town. Working with a global team isn't always the most convenient, but I appreciate getting to be there for my kids' special events.

The best way to carry out this approach is to ask each member of the team the following questions:

- What times are best for you to meet?
- What times are you willing to meet?
- What times are off-limits?

Everyone considers the possibility of working outside their traditional work hours. The resulting Daily Scrum time is the time that works best for the entire team as a whole.

Elizabeth, a ScrumMaster with IBM Quality Software Engineering, talks about a case where asking the three questions would have been helpful:

> *I have learned to always ask the team what will work for them. Some of us are morning people, so a 5:00 AM meeting may not be a problem. Others are not available until 10:00 AM, but 9:00 PM is not a problem for them. I was working with a team that had members in Hyderabad, India and Saint John, Canada. I assumed—always a bad idea—the team I was working with in India was working on their local time. At the end of a 3:00 AM meeting to show my support for global teaming, they politely asked why I had called the meeting so early. Oops! They were working on Eastern Time! Now, I always ask the three questions: when is the best time, when are you available, and when is off-limits?*

The advantage to the approach of sharing the pain is the whole team is taking part in the Daily Scrum each day, and the time selected for the meeting has the whole team sharing the pain. Everyone hears directly from the other team members, so there is less of a chance of confusion than the team might have with other methods.

However, not everyone may like working outside their normal working hours. When using this approach, the team should have the flexibility to take some time off from their normal working hours to compensate.

Table 6.8 summarizes the pros and cons of the Sharing the Pain approach.

Table 6.8 Summary of Pros and Cons for the Sharing the Pain Approach

Pros	Cons
• Aligns best with the interactive spirit of Scrum and Agile. • Whole team takes part in each day. • Everyone hears directly from team. • Team can discuss blockers and remove them immediately. • Team is better able to hold one another accountable. • Flexibility in work schedule.	• Some may not like working outside normal work hours. • Can be challenging for sustainable pace.

Tips for Distributed Daily Scrums

After the team decides on the times that they will meet for the Daily Scrum, they will start attending the Daily Scrums through a teleconference. And, while the general teleconference tips presented in Chapter 2 apply for the Daily Scrum meeting, the team will find there are some special nuances to the Daily Scrum.

The distributed Daily Scrum is unlike the other Scrum meetings in that the Daily Scrum is not a brainstorming, collaborative working session. Instead, it is a brief, intense 15 minutes with members' one-by-one answering questions. Although team members may ask quick questions for clarification, they should handle any in-depth discussion outside the Daily Scrum. This section offers tips to help teams to be more effective with the Daily Scrum.

Removing Side Conversations

Side conversations during the Daily Scrum are a distraction during a co-located Daily Scrum, but they are even more problematic during the *distributed* Daily Scrum. Side conversations introduce two main problems for the distributed team:

- **Increased distraction.** Background noise can be distracting on a teleconference. If the quality of the teleconference or phone lines is already making it difficult to hear everyone, background noise and other voices can make listening even more difficult.
- **Exclusion.** If the Scrum Team is a new team just beginning to learn to work together, and if there are issues between team members or new members on the team, background discussions can cause members on the phone to feel excluded.

Side conversations can be problematic where part of the team meets in one room and other members have dialed into the teleconference using separate phone lines, as shown in Figure 6.5.

One method to prevent side conversations is to have all members dial into the teleconference. Each person dialed in is immediately in an equal position to everyone else. Everyone is speaking directly into the phone, which can make it easier for everyone on the team to hear the conversation. And, verbal side conversations are less of a problem.

Of course, to promote teaming and to give everyone a chance to speak without disruptions, members of the team and the ScrumMaster should help to end any side conversations quickly.

Keeping the Team Engaged

Distributed team members who have teleconferenced into the Daily Scrum can become detached and distracted by other tasks.

Unfortunately, most of us have heard the clicking in the background as one person types away on an email message while another team member is diligently answering the three questions. Most of us who work as part of a distributed team have ourselves had distractions at one time or another by an email message, an instant message, or some other tasks in the middle of the Daily Scrum. Some of us have also heard the ever-popular excuse of "I was on mute" from a team member called on unexpectedly and scrambling to think of a response.

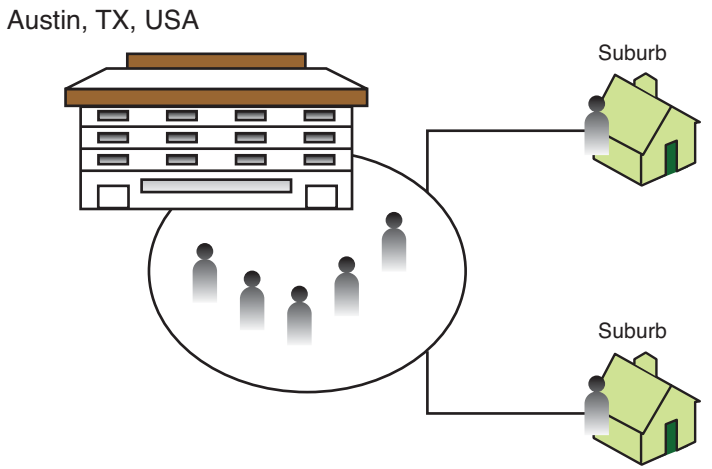

Figure 6.5 Scrum Team with collocated and remote team members

Because the Daily Scrum is an intense 15 minutes, it is especially important for team members to remain actively engaged during the call. If necessary, close down applications, step away from the machine, turn on a screen saver, or just commit to stay focused.

Possibly the best way to stay engaged and to make sure that others on the team stay engaged is to make the time as valuable as possible for the team. Bill Krebs, a coach with IBM Quality Software Engineering, suggests:

> *To help your team members to find value in your responses, focus on the three A's of the Scrum meeting:*
>
> - ***Awareness.*** *Build awareness of what the team is working on.*
> - ***Advertising.*** *Advertise for collaboration.*
> - ***Attack blockers.*** *The team and ScrumMaster should strive to fix all blockers within one hour of the Daily Scrum.*

Facilitating the Meeting

Facilitating a Daily Scrum meeting can be more challenging for the ScrumMaster working with a distributed team. On collocated teams, members are able to read body language and use body language to show a speaker they are taking the meeting off track. The distributed ScrumMaster and team members have to rely on other techniques.

As an example, the ScrumMaster with a co-located team starts with the person on his or her left and goes around the room asking for responses to the three questions. In a distributed environment, as individuals come into the call, they will identify who they are. The ScrumMaster then

calls each person and asks for their response. They may respond in the order they arrived at the teleconference or the ScrumMaster may choose to call on each person. One team member reported:

> *Early on, we experimented with having the team members go in alphabetical order by name, so the ScrumMaster wouldn't have to call on individuals. We found it to be distracting and much less efficient than just having the ScrumMaster call on people.*

Another example is a case where the ScrumMaster is trying to move on from one team member to the next. In a co-located environment, the ScrumMaster can turn his or her body to the next team member when someone is being too verbose or elaborating beyond the responses to the three questions. By teleconference, this technique is not available. Instead, the ScrumMaster is likely to have to interrupt the speaker verbally. Although this would be rude in other cases, verbally interrupting can help the team to more efficiently use the Daily Scrum time.

Using a technique for getting people to focus on just the three questions and to streamline their responses can help to prevent having to interrupt. One technique presented by Bill Krebs of IBM Quality Software Engineering is to focus on the "top two":

> *I like to ask teams to give two headlines of what they did yesterday, two headlines of what they are going to do today, and any blockers. This helps to keep them from getting into too much detail about how they did the work.*

Jean-Louis Marecheaux has used a countdown timer to streamline responses:

> *I have used a countdown timer in the past, visually displayed on a web conference like NetMeeting or ST Unyte. It helped speakers keep track of the time left within the allowed time slot.*

Taking Daily Scrum Notes

Notes are typically not taken during the Daily Scrum; the team must only log the blockers. However, when working with a distributed team, it can be helpful to have the notes available in a common location. Wikis that allow multiple users to write at the same time can be a good repository for Daily Scrum notes. Regardless of the archiving tool used, it is easier for the team to use a single, common archive.

Having the notes available in a common location allows those not able to attend the Daily Scrum to share their tasks with others and provides them with a way to learn about the tasks of other team members.

Taking notes for the Daily Scrum can also help distributed teams that are dealing with language difficulties. For those who are using their second language on the teleconference, having the text prepared earlier can help them to state their responses. And, it can help others who are using the primary language of the meeting to understand what they are saying.

Daily Scrum notes are especially important for teams working in a regulatory compliance environment (such as the FDA, HIPPA, or others), which one-third of agile teams claim to do (Ambler 2009).

Dealing with Language Barriers

The Daily Scrum is an intense meeting with rapid communication. When a distributed team uses a language that is not the first language of all participants, communication becomes more difficult. Although one-on-one meetings and interactive meetings provide opportunities for those members using their second language to ask questions, translation on the Daily Scrum call can create significant delays.

There are two ways teams can use to deal with language barriers:

- **Use a chat session.** The person who is having difficulty can ping one of the team members who is not speaking to get clarification.
- **Set a longer time-box for the Daily Scrum.** Scrum is about the team working together effectively. If the team is consistently unable to complete the Daily Scrum in 15 minutes because of language issues, set a time-box that works for the team and stick to it. This is not an excuse to poorly manage the Daily Scrum, but instead is a way of handling a language barrier. Matt Ganis shared the following story:

  > *One of our Scrum Teams had a daily 15-minute Scrum scheduled between team members in India and the U.S. We quickly found that 15 minutes was not enough time to resolve issues or come to a common understanding on what we were discussing. We decided that since the Daily Scrum was the one time the team was sure to talk during the day (because of time zone issues) that it would make sense to expand the duration to 45–60 minutes. The first part of the meeting, we did a traditional standup meeting, followed by a more detailed discussion of issues and plans to address the "blockers."*

Tools to Help with Distributed Daily Scrum

Because the Daily Scrum is such a short meeting and focuses on verbal responses to three questions, there is little need for tools. However, a chat tool, wiki or other repository, and a full-duplex phone can be helpful.

Chat tools can be a distraction for some team members, but they can also be valuable in cases where a team is having language issues. The person who is having difficulty understanding someone else's responses can ask questions through chat without disrupting the Daily Scrum. The team may want to cut and paste or save the chat transcript to a wiki for future reference.

A wiki or other repository can be helpful for sharing Daily Scrum notes. Notes can be helpful in communicating with team members who are unable to attend a given Daily Scrum as discussed earlier in this chapter.

If part of the team is calling in from a conference room and using a speakerphone, it should be a full-duplex phone. With half-duplex phones, participants can only hear only one speaker at a time, either the team members in the room or an individual dialed into the teleconference separately. If someone goes on and on without pausing, it can be impossible to interrupt their monologue. With only 1–2 minutes granted for each person, this can be a major obstacle to hearing every team member within 15 minutes. Fortunately, the telephone buttons will still work while the person is speaking, so the ScrumMaster or a team member can push a button to get the attention of the other person. But, this is not ideal. Invest in a full-duplex phone if possible.

Videoconferencing tools offer mixed results. From performance issues to setting the camera so everyone is visible to the time needed to set up the Daily Scrum, videoconferencing today may be more of a headache than it is worth. There are, however, teams that have had success with using a wide-angle web camera for the Daily Scrums.

Scrum of Scrums

Besides the Daily Scrum, scaled Scrum Teams will want to engage in a Scrum of Scrums, where representatives from the different teams answer the following four questions:

- What has your team done since the last meeting?
- What will your team do before the next meeting?
- What blockers does your team have?
- What blockers are you about to cause for another team?

The Scrum of Scrums should include five to nine people. Each Scrum Team sends one or two representatives, depending on how many teams are working together. Representatives may be the ScrumMaster or development team members who are knowledgeable about the code of interest to the teams engaging in the Scrum of Scrums. The Product Owner (or Product Owner*s* in an environment with a hierarchy of Product Owners) may also take part in identifying dependencies or issues that might impact the priorities in the Product Backlog.

During the Scrum of Scrums, teams discuss future dependencies, commitments to other team members, issues with integration, and other points that impact one another.

The frequency and duration of the Scrum of Scrums meetings should be enough to allow the Scrum Teams to coordinate their work. Projects with higher dependencies or greater collaboration between teams should engage in Scrum of Scrums more often.

Summary

The first step to running effective Daily Scrums is to decide the meeting days and times for the team. The distribution level has an impact on how challenging the Daily Scrums will be for the team. Collocated teams with members that telecommute part-time can successfully engage in the Daily Scrum by calling into the meeting. Geographically distributed teams with an overlap in

workdays can set up a time to meet daily during their standard workday. Teams with no overlap in their workdays are the most challenging to schedule.

There are four different approaches that teams can take to dealing with no overlap in their workdays. The Sharing the Pain model more closely aligns with providing the same benefit as co-located teams.

The following sections will discuss how to conduct the Daily Scrum meeting first for teams with and without any overlapping work hours. Table 6.9 shows the different approaches with which collocated and distributed teams can decide to perform their Daily Scrum meetings and the method we recommend these teams use.

Table 6.9 Summary of Different Daily Scrum Approaches with Recommendations

Team Type	Possible Approaches	Recommendations
Collocated Team	• Face-to-Face Meeting • Teleconference • Videoconference • Instant Messaging	Face-to-face meeting; it increases the communication, commitment, and collaboration between team members.
Collocated Part-Time	• Face-to-Face Meeting • Teleconference • Videoconference • Instant Messaging	Face-to-face meeting for the collocated team members with a teleconference line open for remote team members.
Distributed with Overlapping Work Hours	• Face-to-Face Meeting • Teleconference • Videoconference • Instant Messaging	Team members should find a time where the work hours overlap and members in the same location meet face-to-face with a teleconference line open to talk with remote team members. If members in the different locations are unable to hear each other, they should consider having everyone dial in. When verbal language is a problem, you can use an instant messaging session at the same time for clarity.
Distributed without Overlapping Work Hours	• Share the Pain • Alternating Meeting Times • Using a Liaison • Documentation	Share the pain by selecting a standard meeting time when everyone will attend because it respects the spirit of Scrum and agile the best. Alternating meeting times also works well by allowing the team to switch off on times that are most comfortable.

Regardless of the approach the team selects for their Daily Scrums, the team has an opportunity to reevaluate the decision at the end of each Sprint. During the retrospective, the team can discuss whether they should consider adopting a different model to improve their performance.

Once the team begins conducting distributed Daily Scrums, they will discover some issues related to the unique points of the Daily Scrum (for example: brevity, intensity, and one-way communication pattern). To be productive, the team and ScrumMaster will need to remove side conversations, effectively facilitate the Daily Scrum, use notes where necessary, overcome language barriers, and recognize local schedules and holidays.

Tools are less of an issue with the Daily Scrum. A chat tool can help to overcome language barriers, the team can use a wiki or another repository to store notes, and a full-duplex phone can help when some of the team members meet in the same conference room.

Scaled teams will engage in Scrum of Scrums meetings as well as Daily Scrums to help with coordination and integration between teams.

References

Ambler, Scott. *Agile Practices Survey Results* (Ambysoft, July 2009). Retrieved December 16, 2009, from Scott Ambler's website: http://www.ambysoft.com/surveys/practices2009.html#Figure7.

Schwaber, K. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002.

ScrumMasters 2 (December 4, 2006). Retrieved November 10, 2008, from YouTube: http://www.youtube.com/watch?v=B3htbxIkzzM.

Sutherland, J., A. Viktorov, J. Blount, and N. Puntikov (2007). *Fully Distributed Scrum: The Secret Sauce for Hyperproductive Outsourced Development Teams*. HICSS'40, Hawaii International Conference on Software Systems. Waikoloa: University of Hawai'i at Manoa.

Sutherland, Jeff (June 3, 2006). *Why the Three Questions in the Daily Scrum Meeting*. Retrieved July 2, 2009, from Jeff Sutherland's website: http://jeffsutherland.com/Scrum/2006/06/why-three-questions-in-daily-Scrum.html.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X-Y

Z