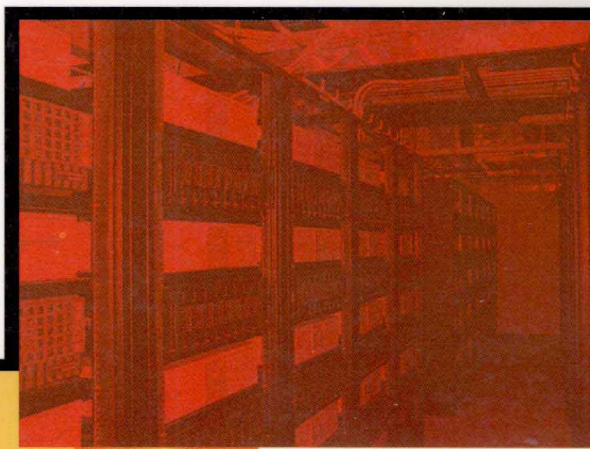
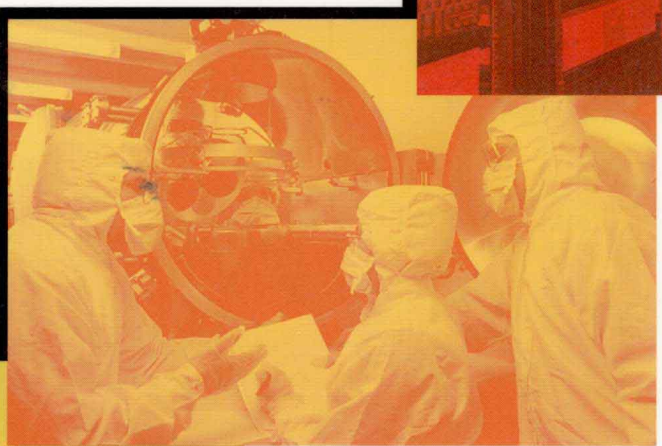
The Practice of System and Network Administration





Thomas A. Limoncelli
Christine Hogan

The Practice of System and Network Administration

Thomas A. Limoncelli
and
Christine Hogan

江苏工业学院图书馆
藏书章

ADDISON–WESLEY
Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Copyright © 2002 by Lumeta Corporation and Christine Hogan

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

The publisher offers discounts on this book when ordered in quantity for special sales. For more information, please contact:

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

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

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web at www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Limoncelli, Tom.

The practice of system and network administration/ Tom Limoncelli, Christine Hogan

p. cm.

Includes bibliographical references and index.

ISBN 0-201-70271-1

1. Computer networks—Management. 2. Computer systems. I. Hogan, Christine. II. Title.

TK5105.5.L53 2001

004.6--dc21 2001041266

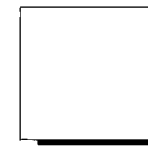
ISBN 0-201-70271-1

Text printed on recycled and acid-free paper

4 5 6 7 8 9 10—CRS—0706050403

Fourth printing, April 2003

Preface



The goal of this book is to write down all the things that we've learned from our mentors and our real-world experiences. These are the things that are beyond what the manuals and the usual system administration books teach.

System administrators (SAs) often find themselves swamped with work, struggling to keep the site running, and faced with requests for new technologies from their customers. Servers are overloaded or unreliable, but fixing the problem requires weeks of planning and painstakingly untangling a mess of services so that they can be moved to new machines. Hidden dependencies are lurking around every corner, and getting bitten by one can be catastrophic. In the meantime, repetitive day-to-day tasks still need to be done. The challenges seem insurmountable.

Most sites grow organically, with little thought given to the big picture as each little change is implemented. Haphazardly, SAs learn about the fundamentals of good site design and support practices. They are taught by mentors, if at all, about the importance of simplicity, clarity, generality, automation, communication, and doing the basics first. These six principles are recurring themes in this book.

- *Simplicity* means that the smallest solution that solves the entire problem is the best solution. It keeps the systems easy to understand and reduces complex interactions between components that can cause debugging nightmares.

- *Clarity* means that the solution is not convoluted. It can be easily explained to someone on the project or even outside the project. Clarity makes it easier to change the system, as well as to maintain and debug it.

- *Generality* means that the solution solves many problems at once. Sometimes the most general solution is the simplest. It also means using

vendor-independent open standard protocols that make systems more flexible and make it easier to link software packages together for better services.

- *Automation* is critical. Manual processes cannot be repeated accurately nor do they scale as well as automated processes. Automation is key to easing the system administration burden, and it eliminates tedious repetitive tasks and gives SAs more time to improve services.

- *Communication* between the right people can solve more problems than hardware or software. You need to communicate well with other SAs and with your customers. It is your responsibility to initiate communication. Communication ensures that everyone is working toward the same goals. Lack of communication leaves people concerned and annoyed. Communication also includes documentation: document customers' needs to make sure you agree on them, document design decisions you make, document maintenance procedures. Documentation makes systems easier to maintain and upgrade. Good communication and proper documentation also make it easier to hand off projects and maintenance when you leave or take on a new role.

- Doing the *basics first* means that you build the site on strong foundations by identifying and solving the basic problems before trying to attack more advanced ones. Doing the basics first makes adding advanced features considerably easier, and it makes services more robust. A good basic infrastructure can be repeatedly leveraged to improve the site with relatively little effort. Sometimes we see SAs at other sites making a huge effort to solve a problem that wouldn't exist, or would be a simple enhancement, if the site had a basic infrastructure in place. This book will help you identify what the basics are and show you how the other five principles apply. Each chapter looks at the basics of a given area. Get the fundamentals right, and everything else will fall into place.

These principles are universal. They apply at all levels of the system. They apply to physical networks and to computer hardware. They apply to all operating systems running at the site, all protocols used, all software, and all services provided. They apply at universities, non-profit institutions, government sites, businesses, and Internet service sites.

What Is an SA?

It's difficult to define what a system administrator is. Every company calls SAs something different. Sometimes they are called network administrators, system architects, or operators. Maybe the name isn't important—a rose by any other name...

此为试读,需要完整PDF请访问: www.ertongbook.com

Explaining What System Administration Entails
It's difficult to define system administration, but trying to explain it to a nontechnical person is even more difficult, especially if that person is your mom. Moms have the right to know how their offspring are paying their rent. A friend of Christine's always had trouble explaining to his mother what he did for a living and ended up giving a different answer every time she asked. Therefore she kept repeating the question every couple of months, waiting for an answer that would be meaningful to her. Then he started working for WebTV. When the product became available, he bought one for his Mom. From then on, he told her that he made sure that her WebTV service was working and was as fast as possible. She was very happy that she could now show her friends something and say, "That's what my son does!"

System administrators do many things. They look after computers, networks, and the people who use them. An SA may look after hardware, operating systems, software, configurations, applications, or security. A system administrator is someone who influences how effectively other people can use their computers and networks.

System Administration Matters

System administration matters because computers and networks matter. Computers are a lot more important than they were years ago. What happened?

First of all, the technology has changed. Corporate computers used to be independent, now they are connected. Business processes used to have a component that involved using a computer, now entire processes are done online and come to a halt if any part of the system is broken.

The widespread use of the Internet, intranets, and the move to a dot com world has redefined the way companies depend on computers. The Internet is a 24 × 7 operation, and sloppy operations can no longer be tolerated. A paper purchase order can be processed any time, anywhere; therefore there is an expectation that the computer system that automates the process will be available all the time, from anywhere. Nightly maintenance windows have become an unheard of luxury. That unreliable power system in the machine room that caused occasional but bearable problems now prevents sales from being recorded.

The biggest change, however, is due to CEOs[1] putting a new importance on computing. In business, nothing is important unless the CEO feels it is important. The CEO controls funding and sets priorities. Now CEOs have become dependent on email. They notice when an outage or an overloaded system slows down their email. The massive preparations for Y2K also brought home to CEOs how dependent their organizations have become on computers.

Management now has a more realistic view of computers. Previously people had unrealistic ideas of what computers could do; seeing them as portrayed in film: big, all-knowing, self-sufficient, miracle machines. This has changed. Even the need for SAs is now portrayed in films. In 1993, *Jurassic Park* (Crichton 1993) was the first mainstream movie to portray computers as needing system administration, leading to a better public understanding of what it is.

Computers matter more than ever. If computers are to work and work well, then system administration matters. We matter.

About the Book

This book was born from our experiences as SAs in a variety of companies. We have helped sites to grow. We have worked at small start-ups and universities, where lack of funding was an issue. We have worked at mid-size and large multinationals, where mergers and spin-offs give rise to more challenges. We've worked at fast-paced companies that do business on the Internet and have high-availability, high-performance, and rapid scaling issues. On the surface, these are very different environments with diverse challenges. But underneath, they all need the same building blocks, and the same fundamental principles apply.

This book gives you a framework—a way of thinking about system administration problems—rather than a narrow how-to solution to a particular problem. Given a solid framework, you can solve problems every time they appear, no matter what operating system (OS), brand of computer, or type of environment. This book is unique because it looks at system administration from this point of view, whereas most books for SAs focus on how to maintain one particular type of OS. With experience, however, all SAs learn that the big-picture problems and solutions are largely independent of the platform. This book will change the way you approach your work as an SA and the way you view the site you maintain.

[1] We use the term chief executive officer (CEO) loosely to mean the top person in an organization. Educational institutions have CEOs, they're just referred to as president, provost, proctor, or head. Governments have CEOs—they're just referred to as mayor, governor, Prime Minister, leader, or President.

The principles in this book apply to all environments. The approaches described may need to be scaled up or down, depending on your environment, but the basic principles still apply. In chapters where we felt that how to apply the information to other environments might not be obvious, we have included a section that illustrates how to apply the principles at different companies.

This book is not about how to configure or debug a particular OS. It will not tell you how to recover the shared libraries or DLLs when someone accidentally removes them. There are some excellent books that do cover those topics, and we will refer you to many of them throughout the book. What we *will* discuss here are the principles of good system administration, both basic and advanced, that we have learned through our own and others' experiences. These principles apply to all OSs. Following them *well* can make your life a lot easier. If you improve the way you approach problems, the benefit will be multiplied. Get the fundamentals right, and everything else falls into place. If they aren't done well, you will waste time repeatedly fixing the same things, and your customers[2] will be unhappy because they can't work effectively with broken machines.

We believe that SAs of all levels will benefit from reading this book. It gives junior SAs insight into the bigger picture of how sites work, their roles in the organizations, and how their careers can progress. Intermediate SAs will learn how to approach more complex problems and how to improve the sites, making their jobs easier and more interesting and their customers happier. It will help you to understand what is behind your day-to-day work, to learn the things that you can do now to save time in the future, to decide policy, to be architects and designers, to plan far into the future, to negotiate with vendors, and to interface with management. These are the things that concern senior SAs. None of them are listed in an OS's manual. Even senior SAs and systems architects can learn from our experiences and the experiences of our colleagues that are captured in these pages, as we have learned from each other in writing this book. We also cover several management topics, both for SA managers and for SAs who aspire to move into management.

The easiest way to learn usually is by example, particularly in the case of practical areas like system administration. Throughout the book, we use examples to illustrate the points we are making. The examples are mostly from medium or large sites, where scale adds its own problems. Typically, the examples are generic rather than specific to a particular OS, although some are OS-specific, usually UNIX or Windows. One of the strongest motivations we

[2]Throughout the book we refer to the end-user of our systems as "customers" rather than "users." A detailed explanation of why we do this is in Section 26.1.2.

had for writing this book is the understanding that the problems SAs face are the same across all OSs. A new OS that is significantly different from what we are used to can seem like a black box, a nuisance, or even a threat. However, despite the unfamiliar interface, as we get used to the new technology, eventually we realize that we face the same set of problems in deploying, scaling, and maintaining the new OS. Recognizing that fact, knowing what problems need solving, and understanding how to approach the solutions by building on experience with other OSs let us master the new challenges more easily.

We want this book to be something that changes your career. We want you to become so successful that if you see us on the street you'll give us a great big hug.

Organization

This book has four major parts:

- Part I, *The Principles,* discusses the most basic issues SAs deal with, but we view them from the perspective of the frameworks that will lead you to doing them well.
- Part II, *The Processes,* deals with change and the frameworks for making changes in ways that ensure success.
- Part III, *The Practices,* collects our thoughts on what makes a great system, a great email service, a great print service, a great helpdesk, and so on.
- Part IV, *Management,* comes next. Don't be afraid—it won't bite you. Actually, it will bite you, and we want you to be prepared. This part should help you understand your organization, your customers, yourself, and your managers. It ends with an exciting chapter on how to fire other SAs—a very delicate situation indeed.

The book ends with several *appendices.*

- Appendix A discusses the roles that you and others play. It's a catalog of the various people we've met or worked with and the value they bring to an organization.
- Appendix B connects the dots. It covers many situations you may experience and points you to the various places in the book that should be helpful. Please don't look at it now because you may find it so interesting that you won't return to finish reading this preface.
- Appendix C contains a list of acronyms used in the text.

Each chapter discusses a different topic, and the topics vary from the technical to the nontechnical. If one chapter doesn't apply to you, feel free to skip it. The chapters are linked to each other, so you may find yourself

returning to a chapter that you previously thought was boring. We won't be offended.

There are two halves to each chapter: The Basics and The Icing. *The Basics* discusses the essentials that you just plain have to get right. Skipping any of these items will simply create more work for you in the future. Consider them investments that pay off in efficiency later on. *The Icing* deals with the cool things that you can do to be spectacular. Don't spend your time with these things until you are done with The Basics. We have made an attempt to drive the points home through anecdotes and case studies from personal experience. We hope that this makes the advice here more real for you. Never trust salespeople who don't use their own products.

What's Next?

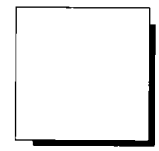
Each chapter stands on its own. Feel free to jump around. However, we have carefully ordered the chapters so that they make the most sense if you read the book from start to finish. Either way, we hope you enjoy the book. We have learned a lot and had a lot of fun writing it. Let's begin.

Thomas A. Limoncelli
Lumeta Corporation
tom@limoncelli.org

Christine Hogan
Independent Consultant
chogan@chogan.com

P.S. Books, like software, always have bugs. We intend to maintain a list of updates to this book on its web site: `http://www.awl.com/cseng/0201702711` or our website, `http://www.EverythingSysAdmin.com`. Please visit!

Acknowledgments



We can't possibly thank everyone that helped us in some way or another, but that isn't going to stop us from trying.

Much of this book was inspired by Kernighan and Pike's *The Practice of Programming* (Kernighan and Pike 1999) and John Bentley's second edition of *Programming Pearls* (Bentley 1999).

We are grateful to Global Networking and Computing, Inc. (GNAC, pronounced Gee-Nack), Synopsys, and Eircom for permitting us to use photographs of their data center facilities to illustrate real-life examples of the good practices that we talk about.

We are indebted to the following people for their helpful editing: Valerie Natale, Anne Marie Quint, Josh Simon, and Amara Willey.

The people we have met through USENIX and SAGE and the LISA conferences have been major influences in our lives and careers. We would not be qualified to write this book if we hadn't met the people we did and learned so much from them.

Dozens of people helped us as we wrote this book—some by supplying anecdotes, some by reviewing parts of or the entire book, others by mentoring us during our careers. The only fair way to thank them all is alphabetically and to apologize in advance to anyone that we left out: Rajeev Agrawala, Al Aho, Jeff Allen, Eric Anderson, Ann Benninger, Eric Berglund, Melissa Binde, Steven Branigan, Sheila Brown-Klinger, Strata Rose Chalup, Brent Chapman, Bill Cheswick, Lee Damon, Tina Darmohray, Bach Thuoc (Daisy) Davis, R. Drew Davis, Ingo Dean, Arnold de Leon, Barbara Dijker, Viktor Dukhovni, Chelle-Marie Ehlers, Michael Erlinger, Paul Evans, Rémy Evard, Lookman Fazal, Robert Fulmer, Carson Gaspar, Paul Glick, David "Zonker" Harris, Cappy Harrison, Sandra Henry-Stocker, Mark Horton, Tim Hunter, Jeff Jensen, Jennifer Joy, Alan Judge, Christophe Kalt, Scott C. Kennedy, Brian Kernighan, Jim Lambert, Eliot Lear, Steven Levine, Branden Lewis, Les Lloyd, Ralph Loura, Bryan MacDonald, Sherry

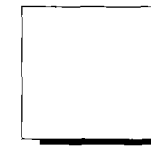
此为试读,需要完整PDF请访问: www.ertongbook.com

McBride, Mark Mellis, Cliff Miller, Hal Miller, Ruth Milner, D. Toby Morrill, Joe Morris, Timothy Murphy, Ravi Narayan, Nils-Peter Nelson, Evi Nemeth, William Ninke, Cat Okita, Jim Paradis, Pat Parseghian, David Parter, Rob Pike, Hal Pomeranz, David Presotto, Doug Reimer, Tommy Reingold, Mike Richichi, Matthew F. Ringel, Dennis Ritchie, Paul D. Rohrigstamper, Ben Rosengart, David Ross, Peter Salus, Scott Schultz, Darren Shaw, Glenn Sieb, Karl Siil, Cicely Smith, Bryan Stansell, Hal Stern, Jay Stiles, Kim Supsinkas, Ken Thompson, Greg Tusar, Kim Wallace, Rabbit Warren, Dr. Geri Weitzman, PhD, Glen Wiley, David Williamson, Pat Wilson, Jim Witthoff, Frank Wojcik, Jay Yu, and Elizabeth Zwicky.

Thanks also to Lumeta Corporation and Lucent Technologies/Bell Labs for their support in writing this book.

Last but not least, the people at Addison-Wesley made this a particularly great experience for us. In particular, our gratitude extends to Karen Gettman, Mary Hart, and Emily Frey.

About the Authors



Tom and Christine know each other through attending USENIX conferences and being actively involved in the system administration community. It was at one of these conferences that they first spoke about collaborating on this book.

Thomas A. Limoncelli

Tom is Director of Operations at Lumeta Corporation, a venture start-up in New Jersey that focuses on enterprise network and security management. Most of Tom's experience comes from his seven years at AT&T and Lucent Bell Labs, where he supported the network used by the researchers and scientists. He led the effort to separate the Holmdel Bell Labs network among AT&T, Lucent, and NCR when the company split into three.

He learned his fundamentals from his three years at Mentor Graphics, Corp. Before that, he was at Drew University in Madison, New Jersey, where he received his B.A. in Computer Science.

Outside of work, Tom is a grass-roots civil-rights activist, who is recognized on both state and national levels. Tom's first published paper (Limoncelli 1997) extolled the lessons SAs can learn from activists. Tom doesn't see much difference between his work and activism careers—both are about helping people.

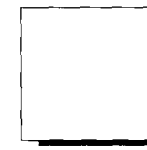
Christine Hogan

Christine's system administration career started at the Department of Mathematics in Trinity College, Dublin, where she worked for almost five years. After that, she went in search of sunshine and moved to Sicily, working for a year in a research company, and followed that with five years in California.

She was the security architect at Synopsys for a couple of years before joining some friends at GNAC, Inc., a few months after it was founded. While there, she worked with start-ups, e-commerce sites, bio-tech companies, and large, multinational hardware and software companies. On the technical side, she focused on security and networking, working with customers and helping GNAC, Inc., establish its data center and Internet connectivity. She also became involved with project management, customer management, and people management. After almost three years at GNAC, she went out on her own as an independent security consultant, working primarily at e-commerce sites for a few months, before returning to a colder climate.

She has recently returned to university and is studying for a Ph.D. in Imperial College, London. Christine also has a B.A. in mathematics and an M.Sc. in computer science from Trinity College, Dublin, and a Diploma in Legal Studies from the Dublin Institute of Technology.

Introduction



The manuals that came with your computer and network equipment do a great job of telling you what commands to type and what buttons to click—but they omit the principles that let you build an infrastructure that makes it all run smoothly. They don't describe the human side of the processes that make it all happen. They describe how to install various services, but they don't reveal to you the insights other system administrators (SAs) have gained through experience and what they have established as best practices. Finally, the manuals don't advise you on the management practices that system administration organization needs to have to keep it running; things like how to structure the organization; size it; and manage it, yourself, your team, and your boss. They also never discuss the one thing SAs always talk about: negotiating salary.

That's why we wrote this book. This is where the computer manuals left off.[1]

The chapters of this book are divided into two primary sections: (1) identifying the "The Basics" and (2) the "The Icing" for each topic. The Basics sections are the things that you should be doing, and The Icing sections are the things you should aspire to do when you have everything else under control. Senior SAs need to be able to say, "This is enough for now, but in the future we will try to accomplish these other things." That is the premise behind the way the chapters are organized. First, deal with the Basics and ignore the Icing sections. When you have finished with the Basics in each chapter, look at the Icing—what's described there will seem more attainable.

[1] We know what you're wondering, and the answer is on page 592. Come back here when you're done.

However, don't completely skip the Icing section if you feel you are still mired in getting the Basics right. Reading the Icing will give you a vision of the ultimate direction in which you will want to go. Ever wonder how those older SAs seem to knowingly leave hooks in just the right places so that future growth "just happens"? That's because they know the things that are in the Icing sections.

I.1 Do These Now!

We tried to order the chapters logically. There are, however a couple issues that are so critical that we want to discuss them first. If you aren't doing these things you're in for a heap of trouble elsewhere. Do the following things right now.

I.1.1 Use a Trouble-Ticket System

SAs receive too many requests to remember them all in their heads. You need software to track the flood of requests you receive. Whether you call this software "request management" or "trouble-ticket tracking" you need it. If you are the only SA, at least you need a personal digital assistant (PDA) to track your to-do list.

Without such a system, we are certain that you are forgetting people's requests or not doing a task because you thought your coworker was working on it. Our customers get really upset when they feel their requests are being ignored.

Fixing the Lack of Follow-Through

Tom started working at a site that didn't have a request tracking system. On his first day, his coworkers complained that the customers didn't like them and did nothing but complain. The next day Tom had lunch with some of the customers. They were very appreciative of the work that the SAs did, *when* they completed their requests! However, they felt that most of their requests were flat out ignored.

Tom spent the next couple days installing a request tracking system. Ironically, this required putting off requests he got from customers, but it wasn't like they weren't used to service delays already. A month later he visited the same customers and they were much happier. They felt they were being heard; their requests were being assigned an ID number and they could see when the request was completed. If something wasn't completed, they had an audit trail to show to management to prove their point—there was less finger pointing. It wasn't a cure-all, but the tracking system got rid of an entire class of complaints and put the focus on the tasks at hand,

rather than not managing the complaints. It unstuck the processes from the no-win situations they were in.

The SAs were happier also. It had been frustrating to have to deal with claims that a request was dropped when there was no proof whether it was dropped or whether a request was ever received. Now the complaints were about things that SAs could control—Are tasks getting done? Are reported problems being fixed? There was accountability for their actions. The SAs discovered other benefits too. They now had the ability to report to management how many requests were being handled each week, and to change the debate from "who messed up" (which is rarely productive) to "how many SAs are needed to fulfill all the requests" (which turned out to be the core problem).

There is a more complete discussion of request tracking software in Section 15.1.7. Until you get there, install a simple system right now. If you use UNIX, we recommend "Request Tracker."

Chapter 16 discusses how to process a single request, with advice for collecting requests, qualifying them, and getting the requested work done. Chapter 15 contains a complete discussion of managing a helpdesk. Maybe you will want to give that chapter to your boss to read.

I.1.2 Manage Quick Requests Right

Ever notice how difficult it is to get anything done when people keep interrupting you? Too many distractions make it impossible to finish any long-term projects. To fix this, organize your SA team so that one person handles the day-to-day interruptions, thereby letting everyone else work on their projects uninterrupted. This person is your shield.

If the interruption is a simple request, the "shield" should process it. If it is a more complicated request, she should delegate it (or "assign" it, in your helpdesk software) or, if possible, start working on it between all the interruptions.

If there are only two SAs, take turns. One person can handle interruptions in the morning and the other can take the afternoon shift. If you have a large SA team that handles dozens or hundreds of requests each day, you can reorganize your team so that some people handle interruptions and others deal with long-term projects.

Many sites are still stuck in the "every SA should be equally trained in everything" mentality. Remember, specialization is a good thing. Try it. You'll like it. Amazingly enough, your customers will like it too. You see, customers generally do have a perception of how long something should take to be completed. If you match that expectation, they will be much happier.

此为试读,需要完整PDF请访问: www.ertongbook.com