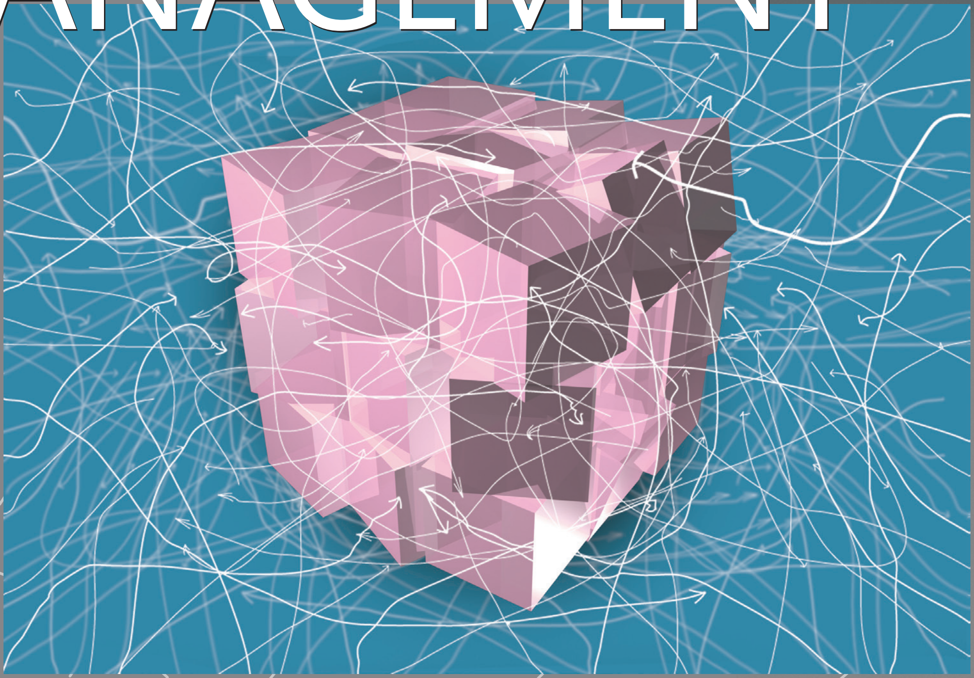
Microsoft

BEST PRACTICES

SOFTWARE CHANGE MANAGEMENT



Case Studies and Practical Advice

Donald J. Reifer

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2011942372
ISBN: 978-0-7356-6475-3

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at *http://www.microsoft.com/learning/booksurvey*.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the author, Microsoft Corporation, nor its resellers or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Devon Musgrave
Developmental Editor: Devon Musgrave
Project Editor: Valerie Woolley
Editorial Production: Curtis Philips, Publishing.com
Copyeditor: Roger LeBlanc
Indexer: Lucie Haskins
Cover: Twist Creative • Seattle

- The SEI also maintains the Software Process Improvement Network (SPIN) Directory at *http://www.sei.cmu.edu/spin/*, which lists process improvement professionals in a given geographic region who are engaged in such activities.
- As you would expect, the International Organization for Standardization (ISO) also provides resources to help understand, justify, and use its standards in practice at *www.iso.org*.
- The Data & Analysis Center for Software (DACS) has a multitude of resources available to support process improvement at *http://www.thedacs.com/databases/url/key/39*. These include lists of best practices, experts, service providers/consultants, and other items.
- An excellent report titled “Measurement for Process Improvement” by Joyce Statz and published by the Practical Software and Systems Measurement (PSM) group is available at *http://www.psmsc.com/Downloads/TechnologyPapers/PI_Measurement_v1.0.pdf*.
- Process improvement conferences are held in the United States and Europe annually under the auspices of the SEI that publish a lot of relevant information. For information on these conferences go to *http://www.sei.cmu.edu/sepg*.

CHAPTER 4

Industrial Case: Moving to Commercial Off-the-Shelf and Open-Source Software Usage in Telecommunications

Setting the stage

This next case study occurs in a large telecommunications firm. The firm wants to move from a custom architecture to an open architecture for its Switching Systems division's product offerings. This division has resisted past attempts to make a move to a new platform and architecture because it had millions of dollars invested in specialized software, which its sales and management leadership viewed as a discriminator in the marketplace. Because of new sales opportunities, the firm initiated the development of a new switch that embraces many innovative concepts. By bringing this switch to market, the firm hopes to retain its market share and position in the future. Everyone in the firm's organizational chain, depicted in Figure 4-1, agrees that it is time to make the changeover and do it right.

The key change being proposed is a move to a new architecture and an open system platform. As shown in Figure 4-2, the applications software will run on top of a POSIX platform riding on top of multiple processors, which execute in parallel to provide growth paths in case more lines need to be added by telephone operating companies (the customers). POSIX will be configured to run using existing facilities to provide platform-designated services on an on-demand basis. Such services include, but are not limited to, configuration and initialization, relational database management, dispatching, distribution, querying, scheduling, and security. Services will run to completion to avoid interrupts that could cause execution to stall, stop, or be rescheduled.

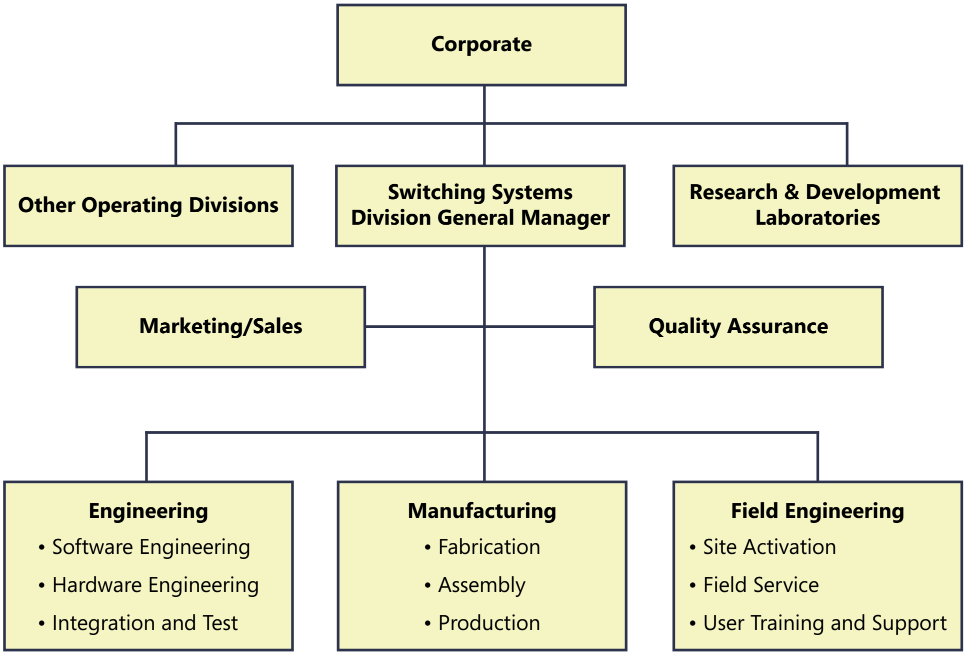


FIGURE 4-1 Telecommunications firm organizational structure.

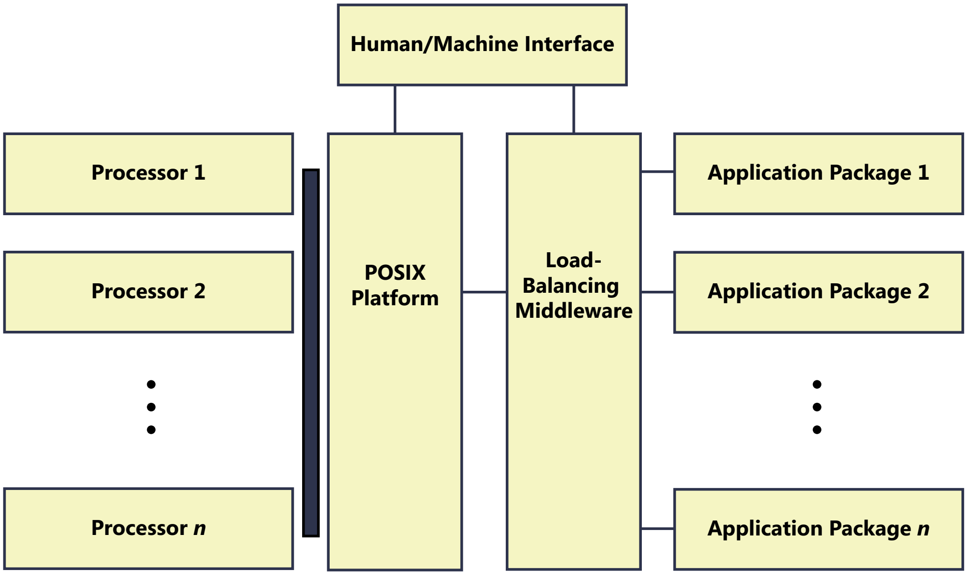


FIGURE 4-2 Top-level switching system architecture.

In addition to using normal switching applications, the new system will provide users with a novel, knowledge-based, self-regulating load-balancing dispatcher and an innovative self-diagnosis and repair system that will act as the marketplace discriminator for sales. The front end of the switch will provide a wide range of network-based and Internet-accessible communications capabilities. It will provide users with easy access to features and query-on-example ability. Context-sensitive help will be provided along with many improved user-interface features to make the system intuitive, easy to understand, and fun to use.

The two key innovative technical enablers that make it feasible to build such a system now are the following: new dispatching algorithms that the Research and Development (R&D) Laboratories invented that facilitate the optimum scheduling of application threads running on parallel processors, and new middleware that allows the system to bind components together using rule-based, load-balancing techniques. Components that are scheduled are fragments of applications packaged by the middleware to execute in parallel on different processors (parallel threads) and share results (self-combinations). Application fragments can include commercial off-the-shelf (COTS) components, open-source or custom routines, modules, or programs, as long as each scheduled and combined entity adheres to the packaging rules, runs under POSIX, and uses the system's data model.

Organization

For this case study, assume that you are the lead software engineer in the Engineering division responsible for developing the new switching system. The initial target of opportunity for sales of the system is a telephone operating company that is your largest customer. This company worked with your people on the architectural specification for the system and helped generate the related functional and performance requirements for it. It wants to buy 100 of these switches, assuming that your organization can deliver them within three years. It is ready to help during the development in any manner possible. The company suggests that it perform the independent verification task, where it provides feedback during the development on the products as they incrementally roll off the drawing board.

Most of the work that the Switching Systems division currently performs is aimed at supporting systems in the field. Major developments like the new system come around once a decade. As such, this represents the means to update the organization's processes, practices, methods, tools, skills, and experience. Management, however, recognizes that by doing too much too quickly your company could fail. In response, they want to attack the development conservatively and use only proven technology. They form a task team to devise a project plan, and you are asked to be a member. You are thrilled and ready to start contributing to the effort.

Project

The project being planned involves the design, development, and validation of a test article that will be used as the model for product development. The development is targeted for three years. Getting manufacturing facilities ready for production will take about year. However, this can be accomplished easily in parallel with the product development because the production facilities are ready for use.

There are no budget details yet. Because the future of the division revolves around the success of this project, you believe management will allocate whatever resources are necessary to pull it off. Even though their funds seem limitless, management wants you to justify every penny.

The planning team is made up of the following six people: you, the team lead, a financial person, the chief engineer, a process person, and a customer representative. Besides several other responsibilities, you have been asked to handle all planning activities associated with COTS and open-source software. Everyone on the team is excited and wants to do a good job.

Your team assessed the current situation and found that both systems requirements and architecture specifications for the new system have been completed by the startup team. These specifications were reviewed as a first order of business and judged to be well done. The team also found that a feasibility study was completed that identified 26 candidate COTS and open-source application software packages for potential use on the project.[1] Several of these candidates have been analyzed on a try-before-you-buy basis, and the results were documented. During development, you know you will still have to select packages, negotiate licenses, and integrate these packages as part of the switching system. You also know that your work with COTS and open-source software will not stop here. There will be annual updates and licensing costs to worry about after the system is operational. Licensing concerns you because there might be run-time license costs associated with some of the packages that have not been accounted for.

Based on this completed work, the team feels much better because their planning efforts would not have to start at square one. In addition, the startup team has completed a high-level budget of $770 million over the three-year development schedule and determined details of the first year's operation in the field, which appear in Table 4-1. As part of your tasking, you are asked to review COTS and open-source forecasts to determine whether or not they are realistic for the job at hand.

TABLE 4-1 Top-level budget for new telecommunications system development and maintenance.

Task	Subtask	Forecasted Budget by Year (in millions)			
		Year 1	Year 2	Year 3	Year 4
Systems engineering	System engineering plan and trade studies	$10	$10	$—	$—
	Integration product team operations	5	10	5	—
Project management	Project management	20	20	20	—
	Measurement and analysis	2	2	2	—
Product support	Configuration management	4	4	4	—
	Quality assurance	4	4	4	—
	Supplier management and licensing	2	2	2	—
	Security and network protection	3	3	3	—
Hardware engineering	Hardware acquisition and readiness	10	10	10	—
	Interface development and test (both hardware and software)	10	10	10	—

Task	Subtask	Forecasted Budget by Year (in millions)			
		Year 1	Year 2	Year 3	Year 4
Software engineering	Requirements analysis	$10	$5	$—	$—
	Software development	25	50	50	—
	COTS and open-source package acquisition and readiness	5	10	10	—
	Software integration and test	10	10	10	—
	Licenses	2	2	2	—
System integration and test	Test planning and readiness	10	—	—	—
	Hardware and software integration and testing	—	25	25	—
	System test and evaluation	—	—	10	—
Manufacturing	Specification	8	3	3	—
	Test article fabrication, assembly, and production	20	25	25	—
	Production article fabrication, assembly, and production	—	25	25	—
Systems test	Test article testing	—	10	10	—
	Acceptance test and evaluation	—	—	10	—
Deployment	Staging and delivery	—	—	25	—
	Dual operations and cutover	—	—	25	—
Operations and maintenance	Planned product improvements (both hardware and software updates and optimizations)	—	—	—	75
	Licenses	—	—	—	5
TOTALS		**$160**	**$240**	**$290**	**$80**

Process

Your next step in the planning process is to determine what work needs to be accomplished to get the product out, determine who will do it, when it will be done, and at what cost. The team lead suggests that the team use a divide-and-conquer strategy to develop the work plan, where each member of the team develops a task list for different parts of the effort. Of course, you are given the COTS and open-source package work as part of your assignment. Your job is to determine whether the line item totals under “Software engineering” titled “COTS and open-source package acquisition and readiness” and “Licenses” are adequate to cover the work required to be completed in these areas.

You first identify the tasks required to employ COTS and open-source packages[2] in the development. Completion of these tasks assumes that the middleware performs as specified and that each package can be cleanly integrated into the system without any rework other than tailoring. The process model used to describe the activities being performed to put COTS and open-source software to work throughout the life cycle is shown in Figure 4-3.