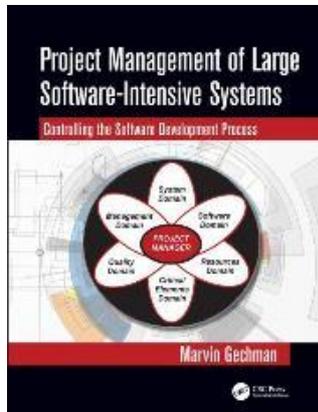


## PM WORLD BOOK REVIEW



Book Title: ***Project Management of Large Software-Intensive Systems: Controlling the Software Development Process***

Author: **Marvin Gechman**

Publisher: CRC Press

List Price: \$99.95      Format: Large paperback, 366 pages

Publication Date: 2019      ISBN: 978-0-367-13671-0

Reviewer: **Lauren Puglisi, MBA, CAPM**

Review Date: June 2019

---

### Introduction

Marvin Gechman offers a “shopping list” of activities and tailoring for large software system developments. With over 57 years of software experience, including 34 in the aerospace industry, he was chosen in 1994 as “the person most responsible for Software Process Improvement” at Lockheed Martin.

Currently President at Escon Software and System Consulting, Mr. Gechman explains how scaling up from smaller projects may no longer be appropriate on large systems development. Working well by themselves, individual components, when interfaced with numerous other hardware and software, can become a *System of Systems* and difficult to integrate by scaling alone.

Although in a software-intensive system the software is the predominant factor, there are many other scaling considerations such as staff, budgets, deadlines, policies, regulations, procedures, and constraints.

Mr. Gechman states the results of the CHAOS Report, published by The Standish Group, 2016:

- 32-35% of projects completed successfully
- 19-24% were failures
- Being late, over budget, or having fewer features challenged 44-49%

The principle of the book is to help project managers and their teams understand how to perform the intended system functions, make the integration seamless, and to increase success. This book deals with large systems and is advanced, technical and detailed.

## Overview of Book's Structure

As a good project manager would do, the author breaks down the information into six functional interactive domains: System, Software, Resources, Critical Elements, Quality, and Management.

The book is rich with definitions, acronyms, tables and figures. The figures are a visual expression of structures, functions, responsibilities, and project dependencies and hierarchies. Flowcharts assist with understanding complex interactions and the bibliography includes 185 references for further study.

There are also nine appendices with titles such as *Criteria for Evaluating Commercial Off-the-Shelf (COTS) and Reusable Software Products*, and *Software Roles and Responsibilities for Skill Groups*.

## Highlights

I have chosen one highlight from each of the sixteen chapters to offer a taste of what Mr. Gechman deems important.

### Chapter 1: **Software Project Management Introduction**

Success and failure depend on strategic planning.

### Chapter 2: **Software Project Management Activities**

Every large project should use an automated database for requirements management and traceability.

### Chapter 3: **System and Software Lifecycle Processes**

Failure can happen even when the system meets requirements and quality but has not been tested in the intended operational and expected environment. (building a plane vs. flying it)

### Chapter 4: **Software Development Methodologies**

An *activity network* should be included in every *Software Development Plan*. The depiction sequences all relationships and dependencies.

### Chapter 5: **Software Management Domain**

Software architecture can be significantly impacted by safety and security concerns.

### Chapter 6: **Software Quality Domain**

From the start, quality is built into every process and component.

### Chapter 7: **Managing the Software Project Team**

The single most important function of the project manager is finding, nurturing, and retaining competent staff.

**Chapter 8: Managing Software Costs and Schedules**

An *Integrated Master Plan* must be linked to an *Integrated Master Schedule* with methodologies planned and followed.

**Chapter 9: Managing Software Facilities, Reuse and Tools**

Teams should use reusable software products whenever possible.

**Chapter 10: Systems Engineering Domains**

A *Requirements Database* should be used to document all requirements and interfaces for the entire system, subsystem to subsystem, and subsystem to external requirements.

**Chapter 11: Software Engineering Domains**

*Human System Integration* is a process of understanding the *man-machine* interface.

**Chapter 12: Collaboration of Software and Systems Engineering**

Recognizing when system capabilities need a software contribution is a challenge and needs to be recognized early in the development.

**Chapter 13: Software Documentation and Work Products**

The closest you can get to a *silver bullet* is the right documentation at the right critical decision-making point in the project.

**Chapter 14: Software Estimating Methods**

*Equivalent Source Lines of Code* (ESLOC) is used to estimate the real cost when design or code is reused.

**Chapter 15: Managing Software Performance with Measurements**

The needs of the project determine the value of collecting, analyzing, and documenting metrics.

**Chapter 16: Managing Software System Sustainment**

The ability to upgrade without affecting the operational system must be integrated into the original design.

**Who might benefit from the Book?**

- Project managers are given proven practices and recommendations.
- Program managers can add management skills to their technical skill training.
- Project teams get a better understanding of what they should be doing at each stage of the development to improve collaboration.
- College students interested in in-depth systems management can read Parts 1-3 in the first semester and parts 4-5 in the second.
- Executive management can gain a better understanding of the challenges and processes to offer better project support.

## Conclusion

Mr. Gechman has written an advanced, technical, and detailed treatise on large system integrations. With the disappointing statistics of the CHAOS Report provided in the introduction, the author lays out strategic, documented methods and recommendations to improve large system successes. The quote in Chapter 6 states the conclusion:

*“The quality of a software product stems, in large part, from the quality of the process used to create it.”*

- Watts Humphrey

---

For more about this book, go to: <https://www.crcpress.com/Project-Management-of-Large-Software-Intensive-Systems/Gechman/p/book/9780367136710>

*Editor’s note: This book review was the result of a partnership between the publisher, PM World and the [PMI Dallas Chapter](#). Authors and publishers provide the books to PM World; books are delivered to the PMI Dallas Chapter, where they are offered free to PMI members to review; book reviews are published in the PM World Journal and PM World Library. PMI Dallas Chapter members can keep the books as well as claim PDUs for PMP recertification when their reviews are published. If you are an author or publisher of a project management-related book, and would like the book reviewed through this program, please contact [editor@peworldjournal.net](mailto:editor@peworldjournal.net).*

---

## About the Reviewer



**Lauren Puglisi**

Texas, USA



**Lauren Puglisi**, MBA, CAPM is an accountant, project management professional, and business analyst with 20 years of experience in the real estate and civil engineering fields.

Lauren can be contacted at [www.linkedin.com/in/laurenpuglisimba](http://www.linkedin.com/in/laurenpuglisimba)