



Perry, Humphreys, Kan, Cohen, Wilson, Fenton, *et alia*.

Reviewed by Dr. Robert Bruce Kelsey, Storage Management Software, DEC — robert.kelsey@cxo.mts.dec.com

How To Run Successful Projects

Fergus O'Connell

How To Run Successful Projects is written by Fergus O'Connell and is published by Prentice-Hall 1994, softbound, ISBN 0-13-138793-6, 170 pages, \$44.95.

O'Connell wants to dispel the myth that software project management is a "black art." His Structured Project Management approach is not, however, a tedious, cumbersome, procedure-oriented workbook for those who lack telepathy or crystal balls. The book is a loosely connected, four part collection of (to use his words) "useful and practical ideas" presented "in an entertaining and stimulating way."

Parts I and II describe the Ten Steps of Structured Project Management. They are quick reads, and one doesn't notice that it isn't until chapter nine that O'Connell addresses software-specific planning. Structured Project Management isn't innovative - the steps include the ubiquitous visualizing the goal, setting the tasks, resource assignment, etc. What makes these sections worth reading is O'Connell's sense of humor and his emphasis on planning and people rather than on how to give good Gantt charts.

Part III begins with short summaries of techniques for interviewing, time management, conflict resolution, and accelerated analysis and design (a technique for brainstorming and documenting requirements). The final section offers a detailed project schedule using Boehm's 1981 data on the relationship between effort and schedule. Part IV describes O'Connell's Probability of Success Indicator and applies this rating system to some of the projects he has cited throughout the book.

Those not fond of militaristic similes and examples may not always find O'Connell entertaining, but those who think "project manager" and "tactician" are synonyms will certainly find a kindred spirit in O'Connell.

Reviewed by Dr. Robert Bruce Kelsey, Storage Management Software, DEC — robert.kelsey@cxo.mts.dec.com

A MAP For Software Acquisition

Jean E. Tardy

MAP For Software Acquisition is written by Jean E. Tardy and is published by Monterège Design 1991, softbound, ISBN 2-9802283-0-3, 243 pages, price not available.

"Most texts...do not have many items of value within their pages...."

So begins this self-styled "guide book" on how to acquire software. Tardy is the president of the consulting firm Monterège

Design, MAP stands for Monterège Acquisition Process, and the book is a consulting support tool, as the course outline in Appendix C makes clear. But instead of describing how to acquire software, the book is little more than a tedious taxonomy of acquisition activities. MAP is full of verbiage but offers little practical advice about how to manage a project, how to develop and track requirements, how to monitor product development, how to perform acceptance tests, and how to determine support requirements.

But what a taxonomy! After oversimplifying "Software Engineering Basics" and "The Software Lifecycle," Tardy dissects "acquisition:" the acquisition process (of which there are four types, Extension Acquisition, Substitution Acquisition, Component Acquisition, and Autonomous Acquisition) comprises production activities and management activities; four production activities (definition, design, development, and delivery) each generate one of the four major deliverables (specifications, plans, program, and product) while the product itself matures through five distinct baselines; management activities include project initiation, direction, monitoring, and the project conclusion. Etc., etc., for another hundred or so pages.

In Tardy's view, "software quality refers loosely to an assessment of software based on features which are *not part of its specific functional requirements*" (21, italics mine). That statement may be disputable but it is not without use, for one could on the same logic separate the quality of a book from its content. For a book that omits object oriented technology from its description of software engineering, cites only one source and only four quality factors for software, and uses the equations for estimation from Boehm without proper calibration (221), that may be advantageous....

Reviewed by Dr. Robert Bruce Kelsey, Storage Management Software, DEC — robert.kelsey@cxo.mts.dec.com

Debugging the Development Process

Steve Macguire

Debugging the Development Process is written by Steve MacGuire and published by Microsoft Press 1994, (paperback), ISBN 1-55615-650-2, 183 pages, \$24.95.

This book was a pleasure to read. It deals mostly with system delivery issues (of which programming is just a part) which are important to a technical contributor. It was easy to read and made a number of important and useful points, most of which are easy to implement in your day to day practice as a technical person.

The key points are summarized in larger, italic type centered on 3 inch lines. You can't miss them. For example:

Be sure that every report you ask for is worth the time it takes for the writer to prepare it.

Don't implement features simply because they are technically challenging or "cool" or fun or...