

The Structured Systems Development Manual

The Structured Systems Development Manual

~~David BENJ~~

~~Independent Consultant,~~
~~Assistant Professor of Computer Science~~
~~Pratt Institute~~

Susan Suchman

Independent Consultant,
Assistant Professor of Technical Writing
New York Institute of Technology



YOURION PRESS
Prentice Hall Building
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

Bellin, David. (date)
The structured systems development manual / David Bellin, Susan
Suchman.
p. cm.
Bibliography: p.
Includes index.
ISBN 0-13-853623-6 :
1. System design. 2. Electronic data processing—Structured
techniques. I. Suchman, Susan. II. Title.
QA76.9.S88B45 1989
004.2'1—dc20

89-32453
CIP

Editorial/production supervision: bookworks
Cover design: 20/20 Services, Inc.
Manufacturing buyer: Mary Ann Gloriande



© 1990 by System Aid Computer Control, Inc. and Quill Pen, Inc.
Published by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

The publisher offers discounts on this book when ordered
in bulk quantities. For more information, write or call:

Special Sales
Prentice-Hall, Inc.
College Technical and Reference Division
Englewood Cliffs, NJ 07632
(201) 592-2498

All rights reserved. No part of this book may be
reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-853623-6

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

YOURDON PRESS COMPUTING SERIES
Ed Yourdon, Advisor

BELLIN AND SUCHMAN The Structured Systems Development Manual
BLOCK The Politics of Projects
BODDIE Crunch Mode: Building Effective Systems on a Tight Schedule
BOULDIN Agents of Change: Managing the Introduction of Automated Tools
BRILL Building Controls into Structured Systems
BRILL Techniques of EDP Project Management
CHANG Principles of Visual Programming Systems
CONNELL & SHAFER Structured Rapid Prototyping: An Evolutionary Approach to Software Development
CONSTANTINE & YOURDON Structured Design: Fundamentals of a Discipline of Computer Program and System Design
DeMARCO Concise Notes on Software Engineering
DeMARCO Controlling Software Projects: Management, Measurement, and Estimates
DeMARCO Structured Analysis and System Specification
DeSALVO & LIEBOWITZ Managing Artificial Intelligence and Expert Systems
FLAVIN Fundamental Concepts in Information Modeling
FRANTZEN & McEVROY A Game Plan for System Development: Strategy and Steps for Designing Your Own System
INMON Information Engineering for the Practitioner: Putting Theory Into Practice
KELLER Expert Systems Technology: Development and Application
KELLER The Practice of Structured Analysis: Exploding Myths
KING Creating Effective Software: Computer Program Design Using the Jackson Method
KING Current Practices in Software Development: A Guide to Successful Systems
MACDONALD Intuition to Implementation: Communicating About Systems Towards a Language of Structure in DP System Development
McMENAMIN & PALMER Essential Systems Analysis
ORR Structured Systems Development
PAGE-JONES The Practical Guide to Structured Systems Design, 2/E
PETERS Software Design: Methods and Techniques
RIPPS An Implementation Guide for Real-Time Programming
RUHL The Programmer's Survival Guide: Career Strategies for Computer Professionals
SCHMITT The OS/2 Programming Environment
SHLAER & MELLOR Object-Oriented Systems Analysis: Modeling the World in Data
THOMSETT People and Project Management
TOIGO Disaster Recovery Planning
VESELY Strategic Data Management: The Key to Corporate Competitiveness
WARD System Development Without Pain: A User's Guide to Modeling Organizational Patterns
WARD & MELLOR Structured Development for Real-Time Systems, Volumes I, II, and III
WEAVER Using the Structured Techniques: A Case Study
WEINBERG Structured Analysis
YOURDON Classics in Software Engineering
YOURDON Managing the Structured Techniques
YOURDON Managing the System Life Cycle
YOURDON Modern Structured Analysis
YOURDON Structured Walkthroughs, 4/E
YOURDON Techniques of Program Structure and Design
YOURDON Writings of the Revolution: Selected Readings on Software Engineering

FOREWORD

In the past quarter century or so we've come a long way in our understanding of how to put together significant software systems. Back in the late 1960's, despite Structured Programming's good influence on systems-in-the-small, systems-in-the-large were often ossified, inflexible behemoths. Enter Larry Constantine with the fundamental ideas of Structured Design, which led to the sound construction of well-designed, maintainable software. Larry also gave us the notation of the structure chart and the data-flow diagram for us to communicate our systems ideas to others in a graphic form.

People in the 1970's, those who saw the benefits inherent in the principles of Structured Design, grabbed these principles and ran with them. There was a welter of books and papers on the subject from folks like Glen Myers, Wayne Stevens, Larry Constantine/Ed Yourdon and (ahem!) myself. But, even as all this interest in Structured Design was being generated, other folks were asking a rather different question: What about the users?

By the late 1970's, Tom DeMarco, Chris Gane, Trish Sarson and Victor Weinberg (to mention but four) had put into place the ideas and notation of a discipline for eliciting user requirements that we now know as Structured Analysis. This, in turn, was taken to new heights in the 1980's by practitioners such as Matt Flavin (who added a vital information modeling component), Steve McMenamin and John Palmer (who defined the basis of essential modeling), and

Paul Ward and Steve Mellor (who addressed issues of concern to builders of so-called "real-time" systems).

In the late 1980's we saw the consolidation of the various "structured" disciplines and the arrival of usable Computer-Aided Software Engineering (CASE) tools that were more than just toys. So by now we know the principles, notation, and techniques of the structured approach to systems development and we have CASE tools to support our knowledge. And yet, today many shops have not adopted and absorbed structured techniques -- despite all their potential for quality and productivity improvements. Why not?

From my experience of training and consulting on lots of structured projects, I believe that the reasons for nonadoption of structured techniques are many and complex, involving normal human fear and resistance to change. But there is one salient, almost mechanical reason that structured techniques are not universally espoused: Since they are not trivial, they are not easy to install in a typical mid-to-large shopful of software developers.

What, for example, are the life cycle steps of the structured approach? What forms, if any, should we use to capture the status of each step? What naming conventions should we use for our data dictionary? Answering these questions could easily take two to three months of work (tooling up) before the real work on a first structured project actually begins. Pity the poor project team, sitting there working out the details, with the fiery, dragon's breath of users and management on the backs of their necks!

So this is where The Structured Systems Development Manual by David Bellin and Susan Suchman comes in. This book is a *very* practical guide to the structured systems techniques that takes you step by step through the systems lifecycle. It shows you what to do and when, which techniques to use, what the format of the deliverable should be...and more. There is also a brief resume of the meaning and purpose of the standard structured notations such as data flow diagrams, the entity-relationship diagram, and the structure chart.

There are now several "classic" books on structured techniques by authors such as those I mentioned above. But perhaps these classics are only necessary, not sufficient, reading for introducing structured techniques successfully into your shop. To make the techniques really take off like a rocket you should also read, digest, understand, tailor (perhaps), and then adopt the practical steps and guidelines in David's and Susan's book, the book you are now holding.

Go for it and have fun developing truly professional systems!

Meilir Page-Jones

PREFACE

Structured analysis and design are the leading methodologies used when developing computer systems. Business is investing heavily in training data processing staff in these methodologies; however, a training course is insufficient without follow-up. Like any other skill, structured techniques require extra guidance the first time they are used.

The Structured Systems Development Manual addresses this need. It is written for those who have begun to learn structured techniques but feels a need for guidance in their application, including managers, analysts, programmers, and users. Based upon our experience as teachers and consultants to first-time users, we have developed a step by step system that combines review with practice. It allows you to start right in on an actual project, but provides you with guidance and suggestions as you work.

We have also developed a leveled set of data flow diagrams that present all the activities and the flow of information within the systems development process itself. Thus in explaining what you are going to do at each stage of a project, we also show you how the tools of structured techniques are used. As you review and learn, you are also sharpening your familiarity with the methodology. This approach makes the book a valuable resource for you when you need to familiarize other people with the system. Reading about the diagramming tools means reading the diagrams. Users can come up to speed quickly and comfortably using the same resources you do.

In addition to the leveled set of data flow diagrams that show you what to do and when, there are over 23 custom-designed forms. Each form is keyed to a process on the data flow diagram and includes clear indications of the information required at that point in the project. By grouping the forms and following the stages shown on the data flow diagrams, managers and project team members can easily keep track of information in a manner that ensures consistency and completeness. The forms can be used independently as a basis for developing standards, or integrated into your current standards.

The steps in the process of systems development, the organization of information requirements, and the forms themselves have also been designed to complement the use of CASE tools if they are available to you. In some cases, CASE files can be substituted for the forms; in others the form contains needed information that is not included explicitly by most CASE tools.

Finally, using a new methodology also means managing a new methodology. It is our conviction that a major obstacle to the successful introduction of structured techniques in an organization is the absence of clear guidelines for management review. Consequently, in addition to providing project team members with a map of project activities, the leveled set of data flow diagrams also show how and when management activities fit into the project life cycle. This is reinforced by a discussion of what managers should look for at each stage of the systems development process and a phased view of project planning activities.

Our primary focus is practical experience. This is a "doing" book. We do provide a review of key references and a comprehensive bibliography for those who wish theoretical background. The more you work with structured techniques, the more interested you'll become in their theoretical underpinnings and the variety of ways in which they have been adapted and extended.

In summary, we have tried to make the learning process more productive for everyone involved in the introduction of structured techniques. Without slighting the subtlety or versatility of the tools, we have made them simpler to follow and to put into action. In the end, the goal is understanding, communication, and successful systems. This book will help you achieve that goal.

David Bellin
Susan Suchman

ACKNOWLEDGMENTS

We met many helpful people during several exciting years, which gave us the experience to write this manual. Mr. Ed Yourdon was a source of encouragement to us, as was Mr. Ed Moura, our editor at Prentice Hall. The writing of Mr. Meilir Page-Jones (see the Foreword) has helped us, as did that by James Martin. We would both like to thank our many consulting clients for their helpful insights during our assignments. A simplified version of portions of this manual were written by us in Spanish several years ago, during an aid project for the Nicaraguan Ministry of Finance under the auspices of the Institute for Technology and Development. Mr. Lawrence Kramer did a design of the Spanish language forms, which served as the basis for the present design. Pratt Institute helped with computer support and laser printing.

CONTENTS

FOREWORD by Meilir Page-Jones ix

PREFACE xi

ACKNOWLEDGMENTS xiii

SECTION I: MANAGING SYSTEMS DEVELOPMENT START-UP 1

CHAPTER 1: THE SCOPE OF STRUCTURED DEVELOPMENT 3

THE SYSTEM LIFE CYCLE 4

STANDARDS 9

COMPUTER AIDED SOFTWARE ENGINEERING TOOLS 9

Use of CASE Tools 10

QUALITY CONTROL AND MANAGERIAL REVIEW 10

The Review Process 12

THE ORGANIZATIONAL CONTEXT OF DEVELOPMENT 14

LITERATURE REVIEW 15

Structured Techniques 15

Project Management 17

CHAPTER 2: MANAGING STRUCTURED DEVELOPMENT 19

PROBLEMS IN MANAGING STRUCTURED DEVELOPMENT 19

THE PROCESS OF SYSTEMS DEVELOPMENT 20

APPROVE PRELIMINARY PLAN, PROCESS 1.1	22
Review System Request, Process 1.1.1	22
The Preliminary Work Plan, Process 1.1.2	24
APPROVE SYSTEM REQUIREMENTS, Process 1.3	24
Review System Requirement Specification, Process 1.3.1	25
The Managerial Approval Process	25
Create Design Work Plan, Process 1.3.2	26
APPROVE SYSTEM DESIGN, PROCESS 1.5	27
Review System Design Specifications, Process 1.5.1	27
Create Program Analysis Work Plan, Process 1.5.2	28
MANAGERIAL ISSUES FROM DESIGN TO IMPLEMENTATION	28
Flexible Management	29
SUPPORTING THE DEVELOPMENT PROCESS	29
The Development Team	30
Who Makes a Good Analyst?	30
TRAINING	31
The Learning Curve	32
USE OF OUTSIDE HELP	33
THE WALK-THROUGH PROCESS	35

SECTION II: FUNDAMENTALS 37

CHAPTER 3: FUNDAMENTAL THEORY 39

THE USE OF GRAPHIC MODELS 40

An Example of a Graphic Model 41

SYMPTOMS OF MISSTATED SYSTEMS REQUIREMENTS 43

GRAPHIC TOOLS FOR STRUCTURED ANALYSIS AND DESIGN 44

THE IMPLEMENTATION NOTEBOOK 45

CHAPTER 4: RULES FOR THE CONSTRUCTION OF MODELS 47

DATA FLOW DIAGRAMS 48

DATA FLOW DIAGRAM CONVENTIONS 51

LEVELED SETS OF DATA FLOW DIAGRAMS 52

Numbering DFDs 52

Subdividing DFDs 53

Source of Data Flows 54

Interlocking DFDs 54

Process Specifications 55

Lower Levels 55

DFDs and CASE Tools 57

THE DATA DICTIONARY	57
Data Dictionary Notation	58
PROCESS SPECIFICATIONS	59
Choosing Nouns	60
Ordering Steps	60
Omitting Implementation Information	60
Organizing Process Specifications	61
Summary	61
THE IMPLEMENTATION NOTEBOOK	61
USING INFORMATION MODELS	62
STRUCTURE CHARTS	65
Transform Analysis	65
Refining The Structure Chart	67

SECTION III: PUTTING THEORY INTO PRACTICE 69

CHAPTER 5: THE ANALYSIS PHASE 71

PROCESS 1.2.1, CREATE CONTEXT DIAGRAM	73
Process 1.2.1.1, Identify Users	74
Process 1.2.1.2, Identify Events	75
Process 1.2.1.3, Identify Inputs and Outputs	75
Process 1.2.1.4, Balance The Model	76
Process 1.2.1.5, Write a Statement of Purpose	78
PROCESS 1.2.2, DRAW A PRELIMINARY DFD	79
Process 1.2.2.1, Draw a Process for Each Event	80
Process 1.2.2.2, Add Input and Output Data Flows	80
Process 1.2.2.3, Name Each Process	81
Process 1.2.2.4, Add Data Stores	81
PROCESS 1.2.3, COMPLETE LEVELED DFDs	81
Process 1.2.3.1, Identify Systems and Subsystems	83
Process 1.2.3.2, Verify Data Flows	84
Process 1.2.3.3, Complete Data Definitions	84
Process 1.2.3.4, Determine Reporting Requirements	85
Process 1.2.3.5, Write Process Specifications	85
SUMMARY	86

CHAPTER 6: SYSTEM DESIGN 87

PROCESS 1.4.1, DESCRIBE COMPUTER SYSTEM	88
PROCESS 1.4.2, NOTE SOFTWARE LIMITATIONS	89
PROCESS 1.4.3, NOTE DATA REQUIREMENTS	90
PROCESS 1.4.4, RECORD SECURITY REQUIREMENTS	94

PROCESS 1.4.5, RECORD CONTROL REQUIREMENTS	95
PROCESS 1.4.6, ANALYZE IMPLEMENTATION PROCESSES	95
SUMMARY	96

CHAPTER 7: THE PROGRAM ANALYSIS PHASE	97
PROCESS 1.6.1, IDENTIFY ESSENTIAL MODULES	98
PROCESS 1.6.2, IDENTIFY INPUT MODULES	103
PROCESS 1.6.3, IDENTIFY OUTPUT MODULES	103
PROCESS 1.6.4, ADD CONTROL PARAMETERS	103
PROCESS 1.6.5, VERIFY COHESION OF MODULES	104
PROCESS 1.6.6, MODULE SPECIFICATIONS	105
PROCESS 1.6.7, VERIFY DATA DICTIONARY	106
PROCESS 1.6.8, VERIFY FILE USAGE	107
SUMMARY	109

SECTION IV: CASE STUDY 111

CHAPTER 8: THE MEDICAL INVENTORY TRACKING SYSTEM	113
MITS: DESCRIPTION AND SYSTEM REQUEST	113
PROCESSING THE SYSTEM REQUEST	115
SYSTEM REQUIREMENTS SPECIFICATIONS	116
List of Diagrams and Forms	116
List of Process Specifications	127
IMPLEMENTATION NOTEBOOK	133
List of Diagrams and Forms	133
PROGRAM SPECIFICATIONS	144
List of Diagrams and Forms	144
List of Module Specifications	151

APPENDIX A: MASTER FORMS SET 159

APPENDIX B: MASTER LEVELED DFDs OF THE SYSTEM DEVELOPMENT CYCLE 187

APPENDIX C: CASE VENDOR LIST 201

APPENDIX D: GLOSSARY 205

APPENDIX E: ANNOTATED BIBLIOGRAPHY 211

INDEX 217

SECTION I: MANAGING SYSTEMS DEVELOPMENT START-UP

CHAPTER 1: THE SCOPE OF STRUCTURED DEVELOPMENT

This is a book that is meant to be used. Our method has been to extract the core of current structured methodologies and to provide a practical guide to their use in a project. This is a method for "on-the-job" learning, where an actual project provides you with a learning experience, much in the same manner that a good case study in a college course would. There is simply *no* similar practical implementation method in existence. Our method works and has been proven in practice. In addition to the ease of learning that comes from use on the job, we offer the added advantage of *flexibility*: The forms and procedures discussed herein can be adapted to your own company's standards and realities.

Our scope is limited to that stated in the title: the actual process of systems analysis and design. We show you how to manage this process. Not included are attempts to discuss other aspects of the software life cycle, such as programming, testing, implementation and maintenance. This is not to say that our method will not possibly help these other stages of the life cycle. However, we think it is of more use in a text such as this to restrict ourselves to a clearly defined, learnable (hence, teachable and usable) subset of the life cycle.

In introducing the methods of this book, we assume most readers to be somewhat conversant in structured methodologies of some kind. Examples of these methodologies include Yourdon Structured Methods, Jackson Design, Harlan's Black Boxes, etc. A general review summarizing this theoretical background is given in Chapter 3.

THE SYSTEM LIFE CYCLE

All structured methodologies have an idealized life cycle for the entire systems development process. Most large corporations have also developed an internal "system life cycle" model, which is usually unique to that company. There is little standardization or agreement on details such as how many phases should be in the life cycle or the exact nature of the activities in each phase. The most common life cycles are variants on the "waterfall" model, shown in Figure 1-1.

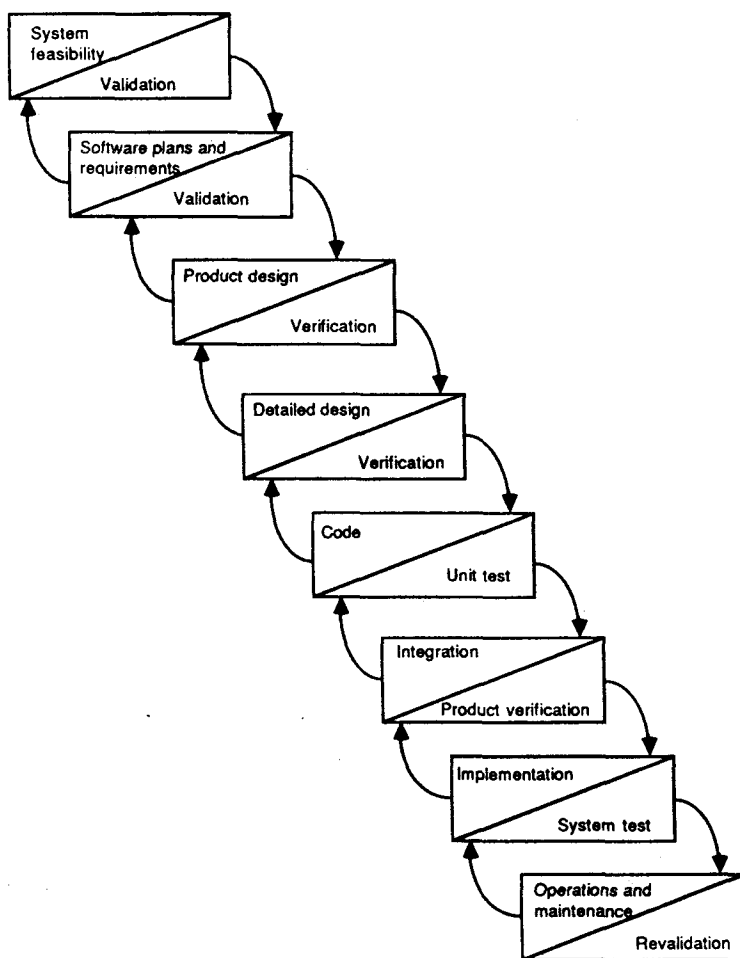


Figure 1.1 The "Waterfall" System Life Cycle Model