

Data Structured Software Maintenance

"Higgins is one of the pioneers in the field of data structured program maintenance. He has done a good job of explaining the basics of data structured program design according to the Warnier/Orr methodology, and has given good tips and examples for maintaining large programs."

—Girish Parikh, Author and Lecturer

"In *Data Structured Software Maintenance*, Higgins offers a realistic assessment of the problem of software maintenance, and he avoids a lot of seemingly easy answers. I recommend the book to anyone who maintains software."

—Stephen M. McMenam, Principal
The Atlantic Systems Guild

"In one sense, this is the first adult guide to the life cycle of software design. . . . Higgins' style is crisp and lively, and his examples clear and down-to-earth. He is also one of those enviable people who is equally clear on paper as in person."

—Nicholas Zvegintzov, from the foreword

"I enjoyed *Data Structured Software Maintenance* and in my opinion it's a good book. . . . Dave Higgins' solutions seem to be practical at the level of programming. The book is COBOL-oriented but the readers using other languages can find analogous solutions to solve their problems. The programming example is very good because it is very simple. Some could say that it is too simple at the logical level, but we must not forget that no one book can cover an entire topic even if it is well defined. This book will be useful above all for programmers and I hope that it will help them to obtain good results."

—Jean-Dominique Warnier

Author and Originator of
the Data Structured Approach to Software Design

About the Author



Dave Higgins is a senior partner of The Ken Orr Institute based in Topeka, Kansas. Together with Ken Orr and the late Jean-Dominique Warnier, Dave was one of the principal architects of the Data Structured Software Development methodology, more widely known as the Warnier/Orr approach.

Partial Contents

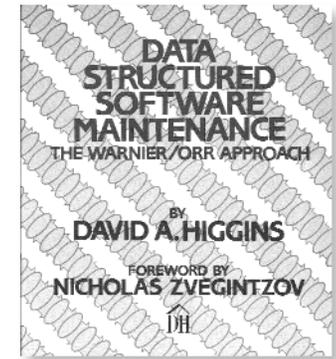
1. **Maintenance Problems**
 - Psychological Objections
 - Technical Issues
 2. **Good Programs**
 - Existing Software
 - Bad Programs
 3. **Improving Program Maintainability**
 - Use Meaningful Names
 - Improve the Organization
 - Improve the Documentation
 - Redesign the Program
 - Reorganize the Data in the Data Base
 - Revise System Requirements and Organization Plans
 4. **Program Design**
 - Logical Mapping
 - Physical Mappings
 5. **Logical Design**
 - Developing the Logical Data Structure
 6. **Physical Design**
 - Developing the Physical Output Mapping
 - Developing the Physical Input Mapping
 7. **Maintaining Good Designs**
 - Understanding Program Entropy
 - Documenting Complex Outputs
 8. **Program Repair**
 - Aspects of Maintainability
 9. **Modifying Traditional Programs**
 - Program Redesign
 10. **Program Enhancement: Part One**
 - Evaluating the Design
 - Modifying the Code
 11. **Program Enhancement: Part Two**
 - Developing the Logical Redesign
 12. **Maintaining Large Programs**
 - Applying DSPD to Large Programs
 13. **Getting Started**
 - Structured Maintenance
 - Tooling Up
- Appendix A: Warnier/Orr Diagrams**
Appendix B: Coding Warnier/Orr Designs

Data Structured Software Maintenance

The Warnier/Orr Approach

by David A. Higgins

foreword by Nicholas Zvegintzov



ISBN: 978-0-932633-03-3
©1986 212 pages softcover
\$29.95 (includes \$6 UPS in US)

Improve Your
Software Maintenance Strategy

Data Structured Software Maintenance proposes a long-term solution to the problem of program maintenance, the largest single expense of data processing departments today. Traditional maintenance procedures cause programs to become unmaintainable over time because of the cumulative effect of changes to the system.

In this book, David A. Higgins argues against the practice of patching a program and redesigning just the part that needs repair or enhancement. Instead, readers are encouraged to use a structured method like the Warnier/Orr approach to redesign and document the existing programs so that they are easier to maintain over the long term.

The Warnier/Orr data structured methodology addresses more than just the coding style or the control structures of a program, and it can be applied even to programs that weren't developed with the method. The ultimate goal of *Data Structured Software Maintenance* is to have a good design for each program and to have the program closely match the design.

Other topics include a definition of good, maintainable programs, logical and physical design, repair and modification of traditional programs, maintenance of large programs, and installation of the Warnier/Orr method into an organization. Numerous examples and more than one hundred figures illustrate the text.

Read more about this book at
<http://www.dorsethouse.com/books/dssm.html>

Studies suggest that the software maintenance process begins without proper knowledge of the software system. This occurs because the software maintenance team is unaware of the requirements and design documentation. Also, traditional models fail to capture the evolutionary nature of the software. To overcome these problems, software maintenance models have been proposed, which include quick fix model, iterative enhancement model, and reuse-oriented model. This model is an approach to modify the software code with little consideration for its impact on the overall structure of the software system. Sometimes, users do not wait for long time. Rather, they require the modified software system to be delivered to them in the least possible time. Software Development Associate Engineer Certification Software Quality and Maintenance Associate Engineer Certification Software Engineering Management Associate Engineer Certification These professional certificates are recognized by major IT companies The syllabus is prepared for teachers responsible for the course (or closely related disciplines), teaching assistants, students enrolled on the course as well as experts and statutory bodies carrying out assigned or regular accreditations. The knowledge and ability to employ function, object, data-structure and component-based design methodologies in a typical software design project. Software Maintenance Causes of Software Maintenance Problems Software Maintenance Cost Factors. next prev. Data Structure Metrics. Essentially the need for software development and other activities are to process data. Some data is input to a system, program or module; some data may be used internally, and some data is the output from a system, program, or module. Example: Program.