Software Development

Manual for the Planning, Realization, and Installation of DP Systems

Wolfgang End, Horst Gotthardt. and **Rolf Winkelmann**

Siemens AG, Munich

Translated by

A. J. and B. D. Tebby

Software Development

Manual for the Planning, Realization, and Installation of DP Systems

Wolfgang End, Horst Gotthardt. and **Rolf Winkelmann**

Siemens AG, Munich

Translated by

A. J. and B. D. Tebby

A Wiley Heyden Publication

JOHN WILEY & SONS

Chichester · New York · Brisbane · Toronto · Singapore

First published under the title Softwareentwicklung:
Leitfaden für Planung, Realisierung und Einführung
von DV-Verfahren. 3 überarbeitete und ergänzte Auflage
Siemens Aktiengesellschaft 1980

Library of Congress Cataloging in Publication Data:

End, Wolfgang.
Software development
Translation of Softwareentwicklung.
"A Wiley Heyden publication"
Bibliography: p.
Includes index
1. Computer programming management I. Gotthardt,
Horst II. Winkelmann, Rolf III. Title.
QA76.6.E513 .98 001.64'25 82-7448

ISBN 0 471 26238 2 (U.S.)

British Library Cataloguing in Publication Data:

End, Wolfgang
Software development.
1. Computer programming management
I. Title II. Gotthardt, Horst III. Winkelmann,
Rolf
001.64'25 QA76.6

ISBN 0 471 26238 2

Typeset in Great Britain by Computer Photosetting (Bilston) Ltd, Great Yarmouth

Preface

The aims of all those who are engaged individually or as part of a team in design, development or management, or of teams, who are involved in planning, implementing, releasing or supporting data-processing systems, must be to keep the costs of the development and maintenance of the software as low as possible and to improve the quality of the software products. This manual is an aid for all these groups as they: follow a systematic approach to planning and implementation; monitor the individual work stages from conception to release; select and use suitable aids for planning and implementation.

The book is based on the experiences of several data-processing departments and the training personnel of the company. It is intended to be a working and training document and can also be used as a reference book. The appendix on 'Check Lists for Software Development', which is a summary of all the check lists, is intended to be a further aid for those involved in software development.

Siemens Aktiengesellschaft

Contents

Introduction

Background

Data processing makes a significant contribution to the execution of tasks in all fields, from research and development through production to business control and management. In addition it enables people to master new types of problems, for example in research when establishing optimum values by simulation, or in management when considering the merits of alternative decisions by calculating their effects on the financial results of the company. As a result of the introduction of data-processing into flight safety systems and medicine human lives may depend on the quality of the software.

The increasing penetration of data-processing into all fields as well as the trend towards developing integrated solutions, means that the development, use, and maintenance of computer programs require a large number of personnel, considerable machine time and extensive hardware investment. In comparison with other business expenditure the resulting costs have reached a considerably high level. In 1977 personnel costs in the data-processing field in the Federal Republic of Germany reached £3000 million and the annual expenditure in rental, depreciation, and maintenance of hardware cost about £2000 million. Analyses of completed software projects show that 'planned' schedules and personnel and computing costs have been exceeded especially in the projects which lacked adequate and satisfactory project planning, monitoring, and control. A 10 % saving by rationalizing software development and by improving the quality of the software products would result in annual savings far in excess of £250 million (only testing time and reruns are considered in the case of the hardware).

Aim of the Manual

Economic development of software requires a clear conception of the objective, procedure, methods, and techniques to be employed in the planning and development. In particular:

(a) software developments should proceed from conception to release in

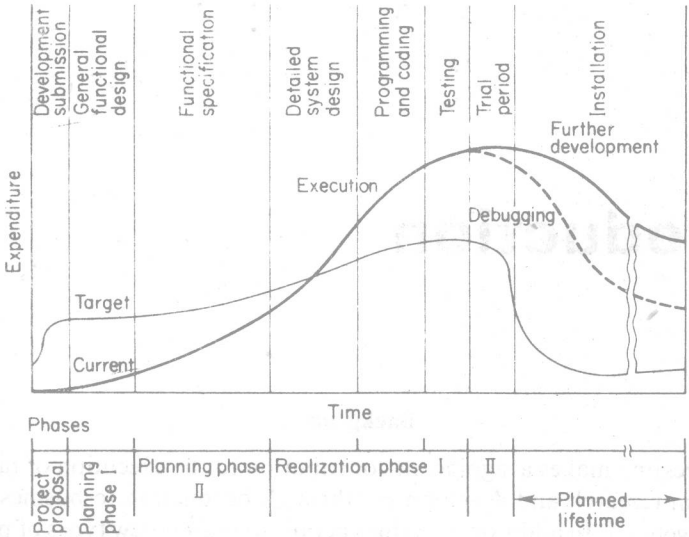


Diagram of expenditure during the development and support of programs

predefined phases with breaks for reviews. In this way one phase must be completed before the next phase is started; and

(b) interdisciplinary groups appropriate to the complexity of the tasks should be set up to determine how the project should be approached. A steering committee or manager examines the results, approves the start of the next phase and coordinates this to fit in with other development projects.

The method of developing software illustrated in this manual and the planning and development aids described can be used in industry, commerce, and finance as well as by government departments and public administration. There is no difference in dealing with developments in the fields of business systems, manufacturing or science and technology. It is also possible to use this approach when undertaking problems which will not be dealt with by computer.

The aim is to run a project on the lines represented by the curve labelled 'Target' (see diagram). Typical faults are inadequate planning at the start of a project, insufficiently systematic progress during the project (for example, lack of reviews) and unsatisfactory realization (techniques, methods, and standards). These usually lead to overexpenditure during installation (testing and trial periods) and to further development and debugging during the planned lifetime (support phase).

The proposals put forward in this manual can and should be used as established conventions by all those developing software systems, thereby improving the quality of their software products.

Limitation

The remarks in this manual apply chiefly to the development of single user or general purpose applications software.

Structure of the Manual

The manual is divided into two parts.

Part I *Phases of software development*
- Project proposal phase
- Planning phase I
- Planning phase II
- Realization phase I
- Realization phase II
- Installation phase

The features of this structure are: (a) strict division into individual development stages with clearly defined breaks for reviews; (b) the main planning effort occurs in the early phases of the software development.

Part II *Planning and implementation aids*
- Planning, managing, and monitoring data-processing projects
- Estimating the duration of a project
- Cost-effectiveness of data processing systems
- Data preparation
- Programming standards
- Estimating run times and optimizing block sizes
- Minimizing test cases
- Software engineering

The planning and implementation aids and working methods will indicate to those planning, developing or using software how to solve or approach the solution of problems which may arise in the individual phases of a software development.

Using the Manual

Standardized network diagram

At the beginning of each phase, a standardized network diagram (MPM network diagram in vertical format) illustrates how each task fits into the whole. The standardized network diagram can be simplified or expanded for particular phases depending on the size and complexity of the project.

Tasks

The various tasks within each phase are dealt with under the same heading in the individual chapters of the manual. The tasks are described and methods and techniques are explained.

Check lists

Check lists are either incorporated in the task description or summarized at the

end of each section. They are a means of checking whether the basic requirements of a task have been taken into consideration.

All the check lists are summarized in the section 'Check Lists for Software Development'.

Personnel–task matrix

The personnel–task matrix indicates the assignment of personnel to tasks and is included under the heading 'The Organization of the Phase'.

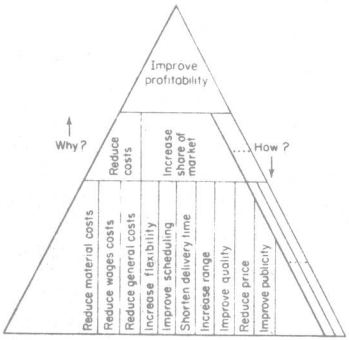
Glossary

The Siemens software products mentioned in the manual and abbreviations used are listed and explained in the Glossary.

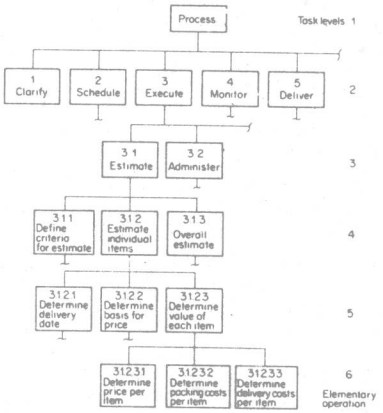
Examples

Through all stages from the formulation of the problem and objectives, to the programming itself, an example is given of each step in software development. The summary on the following pages shows how the chapters are interrelated.

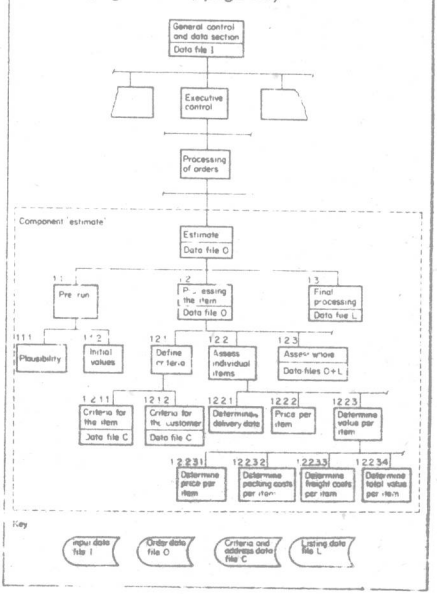
1 The objectives should be formulated on the basis of the requirement (Figure 1.8, page 30)



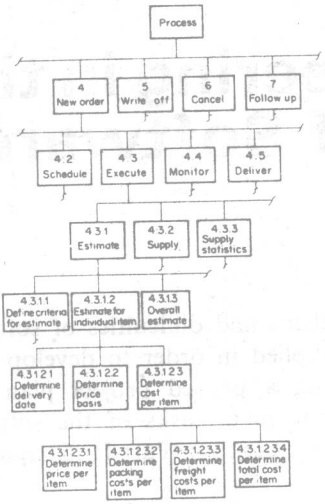
2 After the objectives have been defined, the status study and analysis are carried out and a structure diagram of the operations is drawn up (Figure 2.4, page 48)



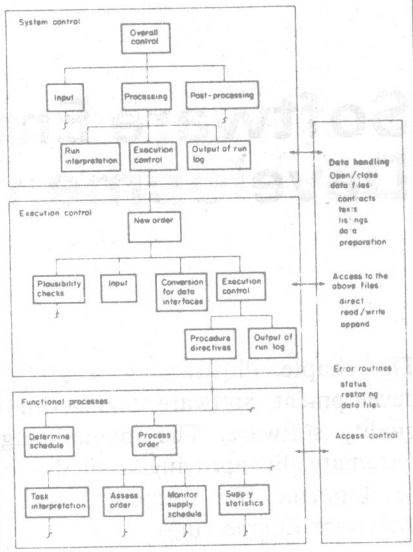
5 The individual components including the data flow are set out in detail in the functional specification (Figure 3.17, page 98)



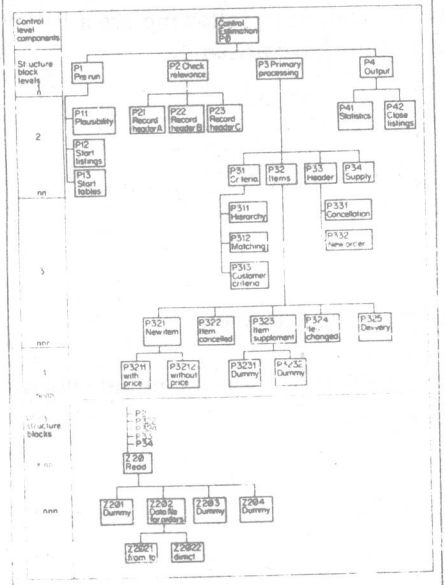
3 The structure diagram of the procedures is drawn up when the functions are defined, based on the structure diagram of the operations (Figure 3.3, page 69)



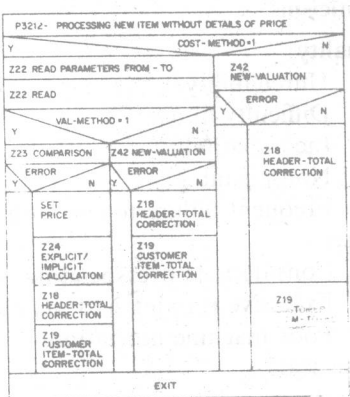
4 Design of the system structure (Figure 3.10, page 92)



6 Plan of the program organization in the specification for the 'estimation' component (Figure 4.3, page 122)



7 Structure diagram and pseudo-code for the structure block P3212 (Figure 4.4, page 125)



Software Engineering in the Development of Software

This chapter illustrates how, given the problems and difficulties of software development, software engineering can be applied in order to develop high quality software. The manual suggests using a phased project plan and systematically applying a methodology to the main tasks of the software development. This theme is examined in greater depth in Part II, Chapter 14 (Software Engineering).

Problems and Difficulties in Software Development

The economic importance of software stands in direct contrast to the problems and difficulties which until now have arisen in the development, implementation, maintenance, and support of software products. These problems can be grouped under three headings: quality, costs, and time-scale. The following are a list of criticisms and a list of reasons why the problems have arisen.

Criticisms

Quality:
- Unreliability
- Difficulty in use
- Incompatibility
- Non-portability
- Frequent failure to carry out the functional specification

Costs:
- Considerable overspending of the development costs in the initial stages
- Excessive running and maintenance costs
- Poor machine utilization

Time-scale:
- Overrunning the schedule
- Excessive development time
- Delayed documentation

Causes

The problems may have been caused by some of the following factors:

(a) inadequate formulation of the requirements by the user;
(b) inaccurate estimation of cost and time (e.g. failure to learn from past experience);
(c) unsatisfactory development methods;
(d) insufficient application of standard solutions;
(e) inappropriate application of software techniques, methods and tools;
(f) non-existent or inadequate testing methodology;
(g) inadequate management control;
(h) insufficient attention to documentation;
(i) incompetence of the personnel involved in the development.

Objectives of Software Engineering

High quality software should be developed by using the principles of software engineering.

The quality of the software is determined by several criteria, the most important of which are described below. In practice these criteria are taken as the hallmarks of quality software.

Functional performance

The critical measure of a software product is the extent to which the functional specification is fulfilled. The temporary omission of some functions is a means frequently used so that a 'partial solution' can be produced to meet an impending deadline.

Ease of use (User friendliness)

This includes all features which enable the operations service, data preparation service, and the user of the system to utilize the software product easily and efficiently.

Examples include: a language suited to the user; ease of learning; and robustness when mis-used.

Efficiency

Efficiency is mainly determined by the size and speed of a program or system. For a long time these were thought to be the only important considerations in the development of software.