

IMPLEMENTATION AND MAINTENANCE

The five chapters in this section discuss implementation and maintenance issues. An application is never completed until it is retired. After analysis and design, we must be able to implement the design on computer hardware using computer software or our work is useless. The first three chapters in this section relate to implementation issues: selecting a computer language; evaluating and selecting hardware, software packages, or consulting services; and testing/quality assurance of the finished product.

Chapter 15 defines characteristics of languages, to allow us to distinguish between ten languages that are evaluated. Then, the languages are matched to the application types discussed in Chapter 1 and to the methodologies discussed in Chapters 7–12. Language selection, rather than code structure, is emphasized because of the increased use of computer-aided software engineering (CASE) tools to

generate code. The language selected must be able to support the application requirements. In Chapter 15, we first describe identifying characteristics of languages. Then, the implementation of each characteristic is described for ten languages. Based on the language characteristics, we define the types of applications for which each language is best suited.

Similarly, outsourcing and use of software packages are growing in all industries because it is frequently cheaper to *buy* rather than *build* an application and/or its environment. In Chapter 16, we discuss the evaluation process and highlight the types and alternatives for soliciting bids from vendors. Sections and contents of a request for proposal (RFP) are defined and developed for the ABC case to show what they look like. Hardware, software, and consulting services might all be contracted for in the same request, or could individually be the subject

of RFPs. Examples of RFP expectation criteria for each type of work are provided to give a sense of the level of detail to which work is defined in an RFP. Then, vendor proposal evaluation alternatives are defined and discussed in relation to ABC Video's application.

Regardless of the development product—packaged software, generated CASE code, or manually programmed code—proving that the software works by testing it at various levels of detail and aggregation is required. Chapter 17 defines the different strategies for testing and types of testing performed. Test types are matched to strategies to develop an effective overall strategy for testing applications. For each level of testing, key issues in test case development are identified. Based on research on testing errors found, guidelines for deciding when to stop testing at each level are provided. The ABC case is then analyzed to demonstrate how the theories apply in practice.

The last two chapters relate to change. Chapter 18 discusses application change management that all take place throughout the life of a project. Change is a way of life in computing and application development is no exception. In Chapter 18, we first discuss how to design for reusability by using templates and reusable modules. Then, change management techniques that apply to documents, decisions, software, and application configurations are presented. The automated tools section includes software representative of each type of change management.

Documentation for project work can be thousands of pages long. Since errors in code usually begin to be traced through documentation, it is important to identify changes to facilitate the error tracing process. Also, users and maintenance personnel who might only infrequently review documentation should be directed to the new information rather than having to read entire documents each time. The techniques for identifying change easily are identified in Chapter 18.

Similarly, application decisions might provide a useful trace of the considerations and discarded ideas throughout a project's life. Few project teams keep such a decision trace because, historically, to do so meant maintenance of more thousands of pages of paper. With automated decision support and sophisticated word processing, keeping a record of decision history is now feasible and can be useful in organizations with rapidly changing management or on projects that support business functions that are subject to rapid industry change.

Software changes and application configuration management are the other major topics of Chapter 18. A recent buzzword identifies *software reengineering*, also called *reverse engineering*, as the backward design of undocumented programs and applications that were probably built without the team having followed a methodology to guide the work. Also called *spaghetti code*, such applications can be maintained beyond a useful life. In the chapter, we describe how to decide when to reverse

engineer, reengineer, or retire applications and/or individual programs. Once the decision is made to maintain software, management of the software maintenance process is an important task in determining that the correct configuration of modules, functions, programs, and so on, is in production. The issue of configuration management is more complicated when multiple versions of software, such as a DOS and MVS versions, exist. Techniques and management practices for configuration management are described in the chapter.

Finally, your career is important and requires management by you for your working life. It is difficult to plan a career without having a sense of what opportunities and expectations are available. First, the typical job levels and types of jobs found in busi-

nesses are described. Then, one way to plan a career by thinking through your wants and requirements for technical, job, company, geography, and opportunities for advancement is developed. A method for defining your chances of job success is defined next. Trends of IS jobs over the last five years by geography, salary, and industry are discussed. Part of developing yourself into a professional and having a career is to maintain your professional status. Techniques for maintaining professional status and building on knowledge areas including education, professional association membership, accreditation, and reading are all defined, with suggested approaches to applying the information to your own situation.