

Effective and orderly software development has become a key issue in the management of data processing. The term *software* is recognized as meaning more than just programs. It encompasses the machines, people, and procedures that surround and are integral to a system. This expanded view of software and the need for its efficient development in an increasingly complex environment have given rise to the field of software engineering, which may be defined as the growing set of disciplines and procedures that facilitate the dependable development and maintenance of software systems. This issue of the *IBM Systems Journal* focuses on application development and the broader concept of software engineering.

Preface

Having as its objective a more orderly development process, the IBM Federal Systems Division has developed a software engineering program over the past several years that is based on twenty years of contract programming experience and reflects many advances in computer science. Consisting of a set of established practices in the design, development, and management of software, this program is discussed in a series of five papers. The paper by Mills introduces the principles of software engineering. O'Neill describes the development of the practices and the extensive educational program for translating them into daily operation. Linger details design practices in his paper; Dyer, development practices; and Quinnan, management practices.

One way to enhance the productivity of the application development process is to solicit the direct participation of the intended user. The paper by Mishelevich and Van Slyke discusses a hospital development system that encourages such participation and is finding applicability across a number of industries. The IBM Health Care Support/Patient Care System (PCS) has been specifically designed to facilitate implementation of large interactive applications. The CRT screen and printer format aids encourage direct end-user involvement in the design and test phases. In addition, a data-collection list facility helps the developer achieve a unique level of data and logic independence.

Another attempt to ease the user's interface with an application is the subject of the paper by Katz, Risman, and Rodeh. Linear programming models are often difficult to use because they require input data to be structured into matrices. At the IBM Israel Scientific Center, the authors have developed an experimental interactive system that allows the user to express data in terminology and forms that are natural to the statement of the problem rather than natural to the computer.

At the other end of the spectrum from the large user is the small computer user, whose application needs are also growing in sophistication. For this user, experienced programming personnel are often a luxury; he must look for already-available software packages. IBM has created a system that can produce both questionnaire-tailored packages for individual users and standard packages for general distribution. This system for generating application programs for small computers is discussed in the paper by Gordon.

Accurate documentation is the key to understanding and maintaining a system. Automating the documentation process helps ensure that it is done and that it truly reflects the system. GREENPRINT is an experimental program developed at the IBM Thomas J. Watson Research Center that produces a graphic representation of the structure of a PL/I program alongside the program source statements. GREENPRINT is discussed in the paper by Belady, Evangelisti, and Power.

The development of complex, interactive applications is requiring more user interaction with developers and more control of the whole process by management. Software engineering and human factors are becoming an integral part of this process, and these papers provide a sampling of some of this activity.

Connie A. Thiel
Editor