Software development methodology

OVERVIEW

Beginning in the late 1960s with an interest in programming methodology, researchers and practitioners alike have begun to pay increased attention to the methods used to develop software. This interest in technique is driven by two major concerns: productivity and quality.

As the cost of software has soared, both in absolute terms and as a fraction of the budget of many organizations, managers have realized that the productivity of the software development group was a critical factor. The call has gone out for improvement and this in turn has caused many people to take a new look at the methods they are using.

The vastly expanded role of software-intensive systems in the operation of most organizations has also caused managers and users alike to look critically at the quality of the software produced. The most obvious quality factor is reliability, but there are many other factors such as security, efficiency, and maintainability. As with productivity, those responsible for building software have realized that one of the variables in building quality software is the method used.

This increased emphasis on methodology has rapidly expanded beyond the original concern with programming alone. Today, we realize that the entire life-cycle of software—from initial need through years of repair and enhancement—must be considered when we start to address issues of quality and productivity.

Matching this life-cycle awareness, we have organized five sessions for NCC '78 that address methodological questions in different phases of software development:

- **Software Design and Analysis** includes papers that address questions of overall system definition and organization.
- **Programming Methodology** will focus on several topics that are relevant to the way in which we create individual programs.
- **Software Verification, Validation, and Testing** samples several new and important concepts in the still-critical area of demonstrating system correctness.
Software Maintenance has been organized especially to address this phase that is often neglected in technical meetings. Although no papers appear in the Proceedings, the panelists represent a large amount of experience with maintenance activity, making this an especially interesting session.

User Experience with New Software Methods is an attempt to redress an imbalance inherent in technical meetings. A panel of six experts with personal and corporate experience using many of the new methods has been assembled. They will share their experience as users—not creators—of the new methods and discuss some of the problems encountered and benefits gained in implementing them. This should be an especially valuable session to attend, in addition to reading the position papers from the panelists that appear in the Proceedings.

All of the paper sessions include a discussant to provide balance and perspective on the presentations. We hope that these sessions will contribute to your understanding of new software methodologies and, in turn, help you and your organization to improve both the quality and productivity of your software development.