Verification and Validation: Techniques to Assure Reliability

Dolores R. Wallace, National Institute of Standards and Technology
Roger U. Fujii, Logicon

Businesses and governments are increasingly placing their investment dollars into software systems whose successful operation is critical to their bottom line. Producing reliable systems requires the use of several techniques—not just review, not just inspection, but not just one type of testing. It requires a well-planned, comprehensive application of several techniques by many players throughout the development life cycle.

These techniques fall under the label "software verification and validation." What is V&V? It is a rigorous methodology with many techniques used across the development life cycle. Software engineering practitioners frequently ask what the differences are between V&V and quality assurance. (The September 1987 IEEE Software explained quality-assurance techniques.) Other practitioners want to know how and when to use specific V&V techniques.

We hope this issue will help you understand what V&V is and how development, quality-assurance, V&V, and other practitioners can use V&V techniques to produce quality software. This issue will help make you aware of methods to select the appropriate techniques for your projects. It will expose you to some techniques and standards that you can apply immediately. And it will show you the relationships and dependencies among various techniques.

We have selected articles to show the broad spectrum of techniques used in V&V and the range of applications you can use it on:

- One article by us, "Software Verification and Validation: An Overview," explains V&V and how it fits in the development life cycle. A sidebar to the article describes some relevant standards that you can use together in a comprehensive approach to V&V that applies limited resources to those features of a project where software is critical to the overall success and where the software is most prone to failure.
- A major aspect of a V&V effort is locating errors in the complete system and demonstrating how well the system satisfies its specifications. "Quantifying Software Validation: When to Stop Testing?" by John D. Musa and A. Frank Ackerman explains how you can use reliability models to determine how much system testing to do and how to allocate resources among the components to be tested. It will show you how to take some of the guesswork out of targeting V&V resources.
- In "Software Inspections: An Effective Verification Process," Ackerman, Lynne Buchwald, and Frank Lewski explain how to perform software inspections to locate defects. The article presents metrics for inspections and examples of its effectiveness. It also contends that adding inspections to your current effort may be more cost-effective than traditional testing alone.
- In "V&V of Flight and Mission-Critical Software," Eugene "Lee" Duke describes the use of V&V to assure the safety of flight-control systems. The V&V process Duke describes is performed by the system's developers, not by a separate group as is typical, and it examines the software as part of the system, with emphasis on techniques for review, simulation, and integration and system testing.
- Finally, Janet R. Dunham projects the future of V&V technology in "V&V in the Next Decade." In this forecast, she examines potential changes resulting from today's research and evaluation studies of V&V techniques and tools and from today's software-engineering environments.

A V&V effort selects tasks from a broad spectrum of analysis and test techniques to tailor each V&V effort to the project's needs. The V&V techniques may include reliability models, group inspections, phase-by-phase analysis and test, sophisticated simulations, and hardware-test facilities. Using these V&V techniques lets you detect high-risk errors early, improve software performance, and establish higher confidence about the software's reliability. The additional cost of conducting V&V is offset by cost advantages of early error detection and improved software reliability.

V&V can coexist with other quality-engineering disciplines and complement many software-engineering disciplines. A major difference between V&V and other quality engineering functions is that V&V is the only methodology other than the developer's methodology that examines the software in detail from a systems viewpoint.

Contrary to popular belief that V&V acts as an adversary in the development process, the articles in this issue show how V&V is actually a partner with the development group in a project's success.

Industries and government agencies spend billions of dollars each year on software systems. As they have with other products, industries are discovering that software quality and reliability are vital to a nation's ability to maintain its competitiveness and high-technology posture in the marketplace. V&V helps them achieve these goals.

The authors' biographies appear on p. 17.