

TECHNICAL REPORTS

These reports are available from Technical Information Center (DTIC) and the National Technical Information Service (NTIS). To obtain a copy of any of the reports, please contact DTIC or NTIS directly, providing them with the ADA number for the desired report. (For example, ADA169705 is the DTIC number for the SEI report Toward a Reform of the Defense Department Software Acquisition Policy.)

Software Engineering Institute
Carnegie Mellon University
Information Management
Pittsburgh, Pennsylvania 15213

Defense Technical Information Center (DTIC)
ATTN: FDRA
Cameron Station
Alexandria VA 22304-6145

National Technical Information Service (NTIS)
U.S. Department of Commerce
Springfield, VA 22161

1987 Technical Reports
1 January 1987 — 31 December 1987
Annotated list of available documents for public release.

ADA182982
Druffel, Wood, Pethia, Greenberger, Shuba
Software and System Warranty Issues

This report addresses technical and administrative issues associated with the system warranty process, and recommends a straightforward, two-page generic system warranty clause that covers software, not in isolation, but as part of a warranted system. The report describes one approach to relieving problems of system failure, and addresses legal, technical, and administrative issues that support warranty enforcement. The goal is to ease the government's burden of proving the existence of a defect for

which the warranty clause provides a remedy. The key to satisfying that goal is to develop technical tests and specifications that provide objective and demonstrable standards against which a claim for breach of warranty can be measured.

ADA181852
Feiler
User Interface Technology Survey

This report is one of a series of survey reports. It is not intended to provide an exhaustive discussion of topics pertinent to the area of distributed systems technology. Rather, it is intended as an informative review of the technology surveyed. These surveys were conducted in late 1985 and early 1986.

One of the core technology areas in which project members conducted a survey was user interface technology. This report attempts to do two things: specify an understanding of user interfaces by presenting a taxonomy that encompasses the various aspects of user interfaces, and indicate the state of the technology today by highlighting some of the major issues.

ADA181156
Newcomer
Tool Interface Technology

This report is one of a series of survey reports. It is not intended to provide an exhaustive discussion of

topics pertinent to the area of user interface technology. Rather, it is intended as an informative review of the technology surveyed. These surveys were conducted in late 1985 and early 1986.

ADA182895

Firth, Mosley, Pethia, Roberts, Wood, W.
A Guide to the Classification and Assessment of
Software Engineering Tools

Software engineering tools are computer programs that assist people in doing the work of software engineering. As understanding of the software engineering process has broadened and the need to solve problems has intensified, there has been increasing interest in using software engineering tools. Understanding what a tool does and comparing it to similar tools are difficult tasks given the diversity of functionality that exists. This report describes a tool classification technique that helps those investigating tools decide where a tool fits in the software engineering process and identify what a tool does or doesn't do. It also provides guidance to the tool evaluation process and lists specific criteria that should be considered when evaluating tools.

ADA182895

Humphrey
Characterizing the Software Process: A Maturity
Framework

Improvement in the performance of software development organizations is an essential national need. The improvement process has five basic elements: 1. an understanding of the current status of the development process, 2. a vision of the desired process, 3. a prioritized list of required improvement actions, 4. a plan to accomplish these actions, and 5. the resources and commitment to execute the plan. This paper addresses the first three of these elements by providing a model for software organizational improvement. The structure of this model provides five maturity levels, identifies the key improvements required at each level, and establishes a priority order for implementation. This model has been tested with a number of organizations and found to reasonably represent the status and needs of actual software development groups.

ADA200542

Dart, Ellison, Feiler, Habermann
Software Development Environments

"Environment" refers to the collection of hardware and software tools that a system developer uses to build software systems. As technology improves and user expectations grow, an environment's functionality tends to change. Over the last 20 years, the set of software tools available to developers has expanded considerably. We can illustrate this change by observing some distinctions in the terminology. "Programming environment" and "software development environment" are often used synonymously, but here we make a distinction between the two.

ADA188929

D'Ippolito, Lee, K., Plinta, Rissman, Van Scoy
Prototype Real-Time Monitor: Requirements

The requirements imposed by flight simulators and good software engineering practice on Ada systems force software engineers to seek new solutions to the problem of monitoring executing software. This report examines some of these requirements and, based on these requirements, defines a subset for implementation as a prototype real-time monitor (RTM).

ADA188930

Van Scoy, Plinta, Coddington, T., D'Ippolito, Lee, K.
Prototype Real-Time Monitor: User's Manual

This report defines the user interface to the prototype real-time monitor (RTM). It defines the concepts and commands needed by a software engineer to use the RTM. In addition to defining the user interface, the report explains the steps needed to tailor the RTM to work with the user's application.

ADA188931

Van Scoy, Plinta, D'Ippolito, Lee, K., Rissman
Prototype Real-Time Monitor: Design

This report describes the software design used to implement the prototype real-time monitor requirements. The design is presented at three levels: system level, object level, and package architecture level. The report concludes with a discussion of the key implementation obstacles that had to be overcome to develop a working prototype: determining system addresses, communicating with an executing application, accessing application memory, converting data into human-readable form, and distributed CPU architectures.