



SOFTWARE ENGINEERING NOTES

An Informal Newsletter of the
SPECIAL INTEREST GROUP ON
SOFTWARE ENGINEERING

Volume 15, Number 5

Oct 1990

Letter from the ExecComm, Tony Wasserman	1
Letter from the Editor, P.G. Neumann	3
Risks to the Public, P.G. Neumann and contributors	3
Correctness is not congruent with quality, D.W. Ketchum	18
Correctness is congruent with quality, C.M. Lott	19
Heisenberg Uncertainty, P. Laplante	21
AI based simulation in reusability, M.W. Aiken	23
Domain description grammar: application of linguistic semantics, R.P. Carasik, S.M. Johnson, D.A. Patterson, G.A. Von Glahn	28
An experience in teaching OOD for Ada software, C. Jean, A. Strohmeier	44
DoD-Std-2167A and methodologies, S.P. Overmyer	50
Biblio of Object-Oriented System Development, S. Mrdalj	60
ABSTRACTS in Software Engineering	63
CALENDAR	65

 * SPECIAL INTEREST GROUP ON SOFTWARE ENGINEERING *

SOFTWARE ENGINEERING NOTES (SEN) is an informal publication of the ACM Special Interest Group on Software Engineering (SIGSOFT), concerned with the design and development of high-quality software. Relevant topics include methodologies for design and implementation, programming techniques, tools, practical experience, software economics, portability of programs, program validation, quality assurance, etc. SIGSOFT seeks to address research and development issues in these areas and to provide a common ground for both, through sponsorship of conferences, symposia and workshops, organization of sessions at major conferences, and the publication of SEN.

Material should be submitted to the editor camera-ready (*originals, single-spaced*) -- or (if brief or newsy) on-line (Neumann@csl.sri.com). Avoid low-resolution dot-matrix printers and unnecessary white space. Lengthy material should be double-columned and reduced, or formatted two-pages-to-a-sheet. Letters, short papers, abstracts of work in progress and available reports, book reviews, specialized bibliographies, notices, news reports, etc., are all welcome. All contributions are normally taken as personal rather than organizational statements. Technical contributions are considered as unrefereed working papers. Controversy is encouraged, but not personal attacks.

Members interested in participating in SIGSOFT activities may contact the Chairman or the Vice Chairman. To join SIGSOFT, contact ACM headquarters, 11 West 42nd St., New York, NY 10036, 212-869-7440. Annual dues are \$16 for ACM members and associates, \$10 for student members, \$38 for others whose major professional allegiance is outside of information processing or computing. Institutional or library subscriptions to SEN are \$30 per year. Nonspecial back issues are \$4 for members and \$9.50 for nonmembers, as available, plus handling; please order *only* from ACM Order Dept., POBox 64145, Baltimore MD 21264.

SIGSOFT EXECUTIVE COMMITTEE

CHAIRMAN:

Richard N. Taylor
 ICS Dept, University of Calif.,
 Irvine CA 92717
 714-856-6429 (messages -7403)
 (taylor@ics.uci.edu)

VICE CHAIRMAN:

Lori Clarke
 University of Mass., COINS
 Amherst MA 01003
 413-545-1328
 (clarke@cs.umass.edu)

SECRETARY/TREASURER:

Elaine J. Weyuker
 NYU Courant Institute
 251 Mercer Street
 New York NY 10012
 212-998-3082
 (weyuker@cs.nyu.edu)

NEWSLETTER EDITOR:

Peter G. Neumann
 SRI International, BN168
 333 Ravenswood Ave.
 Menlo Park CA 94025-3493
 415-859-2375
 (neumann@csl.sri.com)

THREE AT-LARGE MEMBERS:

Nancy Leveson, ICS Dept, U. California, Irvine CA 92717, 714-856-5517 (nancy@ics.uci.edu)
 Dewayne E. Perry, AT&T Bell Laboratories, Room 3D-454, 600 Mountain Avenue,
 Murray Hill NJ 07974, 201-582-2529 (dep@research.att.com)
 Anthony I. Wasserman, Interactive Development Environments, 595 Market St., 12th Floor,
 SF 94105, 415-543-0900 (tonyw@ide.com or tony@ide.UUCP)

MOST RECENT PAST CHAIRMAN:

W. Richards Adrion, University of Mass., COINS, Amherst MA 01003, 413-545-2742,
 (adrion@sh.cs.net)

LETTER FROM THE FIRST CHAIRMAN

I volunteered to write the Letter from the Executive Committee for this issue and thought that I would take the opportunity to reflect on SIGSOFT's history and the advances in software engineering over that period.

The Origins of SIGSOFT

For me, the idea for SIGSOFT goes back to a letter I wrote to Peter Denning, then ACM President, in September, 1973, proposing a Special Interest Group on Programming (or Programming Methodology). At that time, many of the "classic" papers on top-down design, stepwise refinement, and modularity had already been published, and the great structured programming "go to" debate was well underway. Peter's feeling, though, was that this topic was adequately covered by SIGPLAN, and nothing happened for more than a year.

In early 1975, though, the IEEE Computer Society created a Technical Committee on Software Engineering and began publication of IEEE Transactions on Software Engineering, with Raymond Yeh as TC Chairman and TCSE Editor-in-Chief. To me, as a Chapter Chairman and participant in numerous other ACM activities, it seemed that ACM should also support this important, emerging field. Working with Stockton Gaines, then Chairman of SIGOPS, we drafted a letter and a petition in support of an ACM Special Interest Committee on Software Engineering. We co-chaired an organizational meeting at the International Conference on Reliable Software in Spring, 1975, attended by about 80 people. Jean Sammet, then President of ACM, attended the meeting and subsequently supported the formation of SICSOFT in a letter to David Brandin, then Chairman of the SIG/SIC Board. (At the time, SIGs started as Special Interest Committees until they became "proven".)

Ms. Sammet appointed Thomas B. Steel, Jr. as Chairman of SICSOFT, and I became Vice Chairman. Mr. Steel was very busy with his professional responsibilities and was unable to devote time to SICSOFT; he eventually resigned in 1976, at which time I became Chairman. Nonetheless, the membership grew, and our proposal to convert to a SIG was approved the following year. I became the first elected Chairman of SIGSOFT, serving until 1979, and as a member of SIGSOFT's Executive Committee ever since. In looking back, I think that I took three actions that have proved to be significant: 1) agreeing with IEEE TCSE to split the surplus from the International Conference on Software Engineering; 2) appointing Peter Neumann as Editor of Software Engineering Notes (*SEN*), and, of course; 3) spearheading the effort to get SIGSOFT formally approved by the ACM hierarchy. These decisions made money for ACM and created a newsletter that has remained consistently outstanding for 15 years. People joined (and continue to join) ACM and SIGSOFT to get their own copy of *SEN*.

Software Engineering in 1975

In the early 1970s, software engineering focused predominantly on issues of programming methodology. By 1975, though, software specification and design were receiving more attention. Work in program verification, which had begun in the late 1960s, led to work in formal specifications, with numerous projects in the area of abstract data types and formal specification languages. The first published paper on Structured Design appeared in 1974; other design methods, including those of Warnier and Jackson, were also described at that time. Work in the area of requirements analysis, such as SADT (tm), was underway by then, but not generally known at that time.

Other well-known ideas of method and process had their beginnings in the mid-1970s, but didn't make a broad impact until some years later. Data flow diagrams, as used in Structured Analysis, were mentioned briefly in the Structured Design paper. Similarly, object-oriented programming, in the form of Smalltalk from Alan Kay's group at Xerox PARC, was just getting recognized. On a personal note, I wrote in 1975 about Application Development Teams; that's the same idea found in today's Joint Application Development.

The Thompson/Ritchie paper on the Unix (R) operating system was published in CACM in July, 1974, and was followed by broad distribution of Unix, initially to universities. Brian Kernighan and his Bell Labs colleagues helped to popularize the notion of software tools and the principles by which they should be constructed. These ideas were published and presented at the First National Conference on Software

Engineering, held in Washington in September, 1975, and subsequently expanded into a book.

In addition to the tools in the Unix programming environment, other software tools were emerging. Program design languages and static analysis tools were available. Of course, personal computers and workstations didn't really exist in 1975, so tools were text-oriented and ran on minis. Large graphics displays were dedicated to applications such as aircraft design and molecular modeling.

The first courses on software engineering were also taught in the 1974-75 period at schools such as University of Toronto and University of California, Irvine. David Wortman's course at Toronto ran a project called "Software Hut", in which they tried to replicate the activities of a small software house. The first conferences on software engineering education were also held, one sponsored by IBM Canada in summer 1975 and the other organized by Peter Freeman and me, held at Irvine in early 1976. Key topics of these meetings included technology transfer, university-industry cooperation, and preparation of students for industrial software development; these themes remain familiar today.

A Sense of History

To me, it's interesting to see how much of today's research and development in software engineering has its origins in the work of the mid-1970s or earlier. Indeed, many of the issues about programming methodology and management of software projects were discussed in the 1968 and 1969 NATO meetings in Garmisch and Rome.

To be sure, there are some important differences between software engineering in the mid-1970s and software engineering today. First, many organizations have systematized their software engineering practices, either through internal effort or externally imposed guidelines, e.g., DoD 2167A. As a result, the notion of a software engineering process is much better understood.

Second, there has been a vast amount of work in the area of software tools and environments, with various tools aimed at different aspects of software development. The cost of software developers continues to grow, while the price of powerful hardware continues to drop. That trend makes it possible to build sophisticated software development environments, where an integrated set of tools are available on the software engineer's desktop. A decade ago, I wrote about "personal development systems" for the programmer, and today's computing environments make that a reality. Tomorrow's software development environments, incorporating frameworks, repositories, and graphical user interfaces, will require even more computing resources to meet the growing expectations of software developers.

Third, and perhaps most important, problems of software development have become widely recognized, not just among the community of software developers, but at the highest levels of government and industry. This recognition has come from delays in well-known software products, internal software development problems, and well-publicized software errors, such as those covered in Peter Neumann's electronic and *SEN* Risks Forum. Today's executives and managers see the strategic role of software in the success of their organizations; as such, they have become much more receptive to ideas of software engineering as a way to gain improved control over their software development processes. Major computer vendors, including IBM, Digital, and H-P, have also invested heavily in promoting software engineering to their customers and prospects.

The field of software engineering has a short, but lively, history, and almost all of the key contributors are still active in the field. There is a strong connection between today's key issues in software engineering and the foundations established between 1967 and 1977. The goals of software engineering remain unchanged, as we continue to seek ways to improve the software development process and the products created with that process. I'd like to see the software engineering community recognize those links and develop a strong sense of that history. I hope that SIGSOFT and *SEN* can continue to play an important role in that activity.

Anthony I. Wasserman