Software Reliability is not a Warm Blanket
Paul B. Moranda

Myers' remarks in The Open Channel in Computer, June 1978, appear to evidence his scorn for reliability (as it has come to be defined).

In the broad sense, Myers is correct in his contention that the subject of software reliability includes many areas, other than the "numbers" he refers to. But to assert that the Program Committee of the 1975 International Conference on Reliable Software (not software reliability!) should be consulted as "mediators" is like asserting that engineering in the traditional sense of C.A.R. Hoare's keynote speech was the (or even "a") topic of the recent Third International Conference on Software Engineering in Atlanta.

We have to accept the proposition that the subject is as it does (early in the history of information theory, the IRE transactions on that subject included papers on radar detection, pattern recognition, matched filters, and the estimation of gyro drift). In the early days of software reliability there was a conference in Toronto on the subject, and the only mention of probability was tentative and almost apologetic. My fears are that the topics of design, testing, etc. (which are more properly functions of quality assurance), will dominate the "numbers" aspect of software reliability.

Myers' comment that the Jelinski-Moranda model is the "old hardware" model indicates a naivete about both software and hardware reliability. That model is the first event-altered rate model ever applied to software reliability (and it may be the first applied in the larger field of reliability). It is the concatenation of a sequence of detection rates (which Myers incorrectly calls models) that as a whole is the model and which as a whole permits the error content and the MTTF to be estimated.

Had Myers read with greater care the reference he cited (Myers' reference 18.1) he would (perhaps) not be so skeptical about the model being employed on software: The following quote from pp. 472-473 of that reference bears on this point:

Software errors or malfunctions do not, at first, seem to be of the same kinds that occur in hardware: there is no physical mechanism for generation of a software failure: the failures do not appear to be random in any active sense; and once an error is found it is in all likelihood removed from the pool of potential failure-producers.

Another view of software failures, one which emphasizes their passive instead of their active roles, makes the hypothesis of randomness more acceptable. If it is imagined that a large utility program is resident in a computing system and is servicing a steady stream of dissimilar computing jobs, each with its own peculiar demands on the master program, the jobs can be considered as entering at arbitrary points in the master utility program, with attendant opportunities to detect deficiencies in the utility program.

In this concept, that of passive deficiencies being uncovered by the variety of jobs handled by the master, the effect is that errors are detected in a random way.

So it is not the failure that is random, it is the detection, and Myers comment about wear-out of software is moot.

His next statement—that software reliability is not time-dependent—is difficult to understand unless he is saying, as I suspected earlier, that he "equates" reliability to perfection. His modifying comment concerning the "bathtub curve" is incorrect in that it is not the constant middle part of the curve that is the regime of applicability. It is the "leading edge" of the bathtub curve where the Jelinski-Moranda model has applicability; it is in this regime where failures, once fixed, do not recur. In the middle of the bathtub the implicit assumption in hardware analyses is that there are an infinite number of failure-makers in the system, and while I have heard accusations made about certain well-known operating systems having that many defects, the concept is probably not appropriate to most software. (It is the wrong part of the bathtub.)

Myers notes that he criticized the model because it assumes all corrections are "correct," and that I referred to this as "not uncommon." He then "has to disagree." My point was that the assumption is not uncommon among modelers, as is indicated in the subsequent part of the sentence (concerning an analogous assumption in hardware). He may be interested in the fact that if errors are introduced in the correction process, the forecasts which the model makes (when used in a moving average mode) will adjust, and provide a larger estimate of the error content resulting in a smaller estimate of the MTTF.

Myers' comment that "crashes" are not a serious problem in the context discussed is certainly valid, but it depends on the definition: my point is that one of the two teams testing for "some amount of time" will probably be stalled for an inordinate amount of time by some exceptionally difficult (to that team) problem, and the resultant error count will be in "error" unless the "same" amount of time is very long.
He next takes issue with my "journalism," correctly noting that it is wrong to replace "one basic assumption" by "the basic assumption." I apologize for this error. But in my opinion the crime is not heinous, and Myers, in focusing too much on the envelope, missed the message. For as I noted, the "one basic" or "the basic" assumption referred to is not even an assumption at all.

Myers next notes that "the major assumptions" about Z(t) have been criticized by others, and refers to a recent article by Schick and Wolverton. That paper, in my judgment, is highly questionable, and all of the authors' objections (some of which are invented) are without merit. I have incorporated my rebuttal as part of a critique (41 pages in draft form) which is being submitted to the IEEE Transactions on Software Engineering.

Myers' explanation of the difference between measurement parameters and cause and effect parameters is too deep for me. If a program changes with time, it does seem to me that its reliability changes with time. It is true that if you never fix the errors, the reliability is constant (but very poor).

I now get even with Myers' comment on my journalism. He notes that the different hazard functions were not "seriously proposed." This will be news to his readers; perhaps the next time he writes a book he can identify where he's serious and where he's "only fooling."

Myers hardhat attitude about "plausible arguments and fancy equations...[not] worth the price of a cup of coffee" is going to be hard to change. He might be interested in results recently reported by P. A. Hamilton and J. D. Musa (Proceedings of the Third International Conference on Software Engineering), who illustrate the use of the model on an "operating system," a "spooling system," a "timesharing system," and a "text editing system" (these are not toy programs).

Finally, if the bile and blood expended in our interchange have uncovered a single truth, our wounds will be well worth regarding as badges.

Assistant Professor of Electrical Engineering

Applicants are sought with specialty in one of the following areas: digital communications, semiconductor devices, computer and control. A Doctorate in Electrical Engineering is required. Ability to combine teaching and research is essential. Send application and resume before August 5, 1978, to Dr. Eugene S. McVey, Dept. of Electrical Engineering, University of Virginia, Charlottesville, VA, 22901.

Assistant Professor of Electrical Engineering

A growing Computer Science program in the Department of Mathematical and Computer Sciences seeks to fill three positions. The department currently offers bachelors degree in CS with graduate programs pending. Applicants are sought for August 1978, January 1979 or August 1979. Qualifications include PhD in computer science or a closely related area. Rank and salary will be competitive and dependent on qualifications and level of experience. These are open positions and may be filled at any time. Letters of application including a current vita and a list of at least three references should be sent to Professor Robert Mathis, Math and Computing Sciences, Old Dominion University, Norfolk, VA 23508. AAEEO.

**Computer Engineer**

ASSOCIATE DEVELOPMENT ENGINEER

U.C. Irvine

($1724-$2080 per month)

Specify, design, and supervise the installation and operation of improvements to the computer facilities of the Medical Image Laboratory of the School of Engineering. Requires BS degree in Computer Engineering or Computer Science plus 3 years engineering experience including 1 year of responsible design work plus demonstrated experience in the design and documentation of computer hardware and software. Ability to design and construct and install improvements to computer facilities; ability to carry out research on computer architecture. Experience with computer image analysis and interactive graphics is desirable.


UNIVERSITY OF CALIFORNIA, IRVINE

Personnel Department

Room 152, Administration Bldg.

Irvine, CA 92717

Affirmative Action Employer

Minorities, Women, Handicapped Persons and Veterans Encouraged to Apply.

**Computer Systems Analyst**

To be responsible for the design, purchase, implementation, and software maintenance of a computer network in the College of Engineering. This network will include a Burroughs B-1726 and a PDP-11/34 with extensive peripherals which will support a computer graphics laboratory and control real-time experiments via attached microprocessors. Will also advise and work with faculty in making optimal use of the University computing facilities (Amdeal 470 V/Q and IBM 360/67). B.S. or M.S. in computer-related field and knowledge of the software required for A/D hardware interfacing. Salary: $23K-$25K, generous fringe. Contact P.K. Rol, 5050 Anthony Wayne Drive, Wayne State University, Detroit, MI 48202. (313) 577-3861. Wayne State University is an Equal Opportunity/Affirmative Action Employer.

**Computer Engineer**

**ENGINEERING POSITIONS**

**$15,000—$40,000**

Openings at all academic levels to cover all levels and disciplines. Over 1,000 items represented. To cover all your immediate job possibilities in the "unpublished" market, send your resume with salary requirements now. No fee. Confidential, professional.

**KEY SYSTEMS, NATIONAL HEADQUARTERS**

New Bridge Center, Kingston, PA. 18704

717-222-2196

**Consider these challenging professional opportunities with an International Leader in the Telecommunications Field:**

**MICROPROCESSOR HARDWARE DESIGN & TESTING**

Position requires BS/MSEE with interest in design and testing of microprocessor system hardware.

**MICROPROCESSOR CHIEF PROGRAMMER**

Real-time programming experience. Knowledge of structured programming. BS/MSEE education with experience preferably in communications and/or telephone switching.

**SYSTEMS ANALYST**

Bachelor's degree plus 1-5 years experience in analysis and design of computerized business information systems. Experience in data-base/data communications-oriented applications for a manufacturing company desired. Must have sound knowledge of business functions and have strong verbal and written communications skills.

**ANALYST PROGRAMMER**

You will work in an IBM 370-148 SVS environment. Must have in-depth knowledge of COBOL, JCL, utilities, etc. BS or BA preferred; AA acceptable or related work experience.

GTE LENKURT, Incorporated, a subsidiary of General Telephone & Electronics Corporation (GTE) is a major manufacturer of video, voice and data communications systems. We offer excellent salaries and a comprehensive competitive benefits package.

Please send your resume, including salary history to Kazim Ali, Employment Manager, GTE LENKURT 1105 County Road, San Carlos, CA 94070. (415) 595-3000. An Equal Opportunity/Affirmative Action Employer.

**PUBLICATIONS AVAILABLE**

**PHYSICAL DESIGN AUTOMATION**

System Study and Implementation Plans by Robert J. Smith, II

A unique book on PC board and IC layout software systems: tells how to implement a layout system on your computer. Uses proven techniques for a practical DA system. For the first time, a nonproprietary planning and design document that shows what must be done, and how to do it. Discusses data base, design data entry and verification, placement, routing algorithms, and implementations, and development planning. Also includes alternatives, extensions, references and a glossary. 240 pp.

ORDER THIS IMPORTANT NEW BOOK NOW $100 PREPAID FROM:

V-R Information Systems, Inc.

Box 12051, 1101 Sand Plum Lane, Wichita, KS 67227

July 1978