Bedding WOW

By I. M. Tired

While you were nestled all snug in your beds, the conscientious and underpaid WOW staff was labouring on a caffeine and pizza high, setting new standards in literary excellence (as much as their spell checker allowed :-). The devoted crew, led by their fearless leader Bashar (nose for) News-eibeh, burned the midnight oil the last three nights for your early morning reading pleasure.

How late, you might ask, did this labour of love last?

Tuesday was easiest, with a 01:30 wrap time. Wednesday, fuelled by cold pizza and red wine, took until 02:45 to crop, layout the photos, and make the columns fit just right. This issue was the most involved, and is still not finished (4am)!! This experience has left all the staff convinced that they made the right choice of professions.

***ICSE-18 WOW editorial team**: Photo by Keng Ng, post-processing by Sabine Lembke. Ken Wong was unavailable for the pose.*

ICSE-18 Window on the World

Editor-in-Chief: Bashar Nuseibeh
(Imperial College London)

Reporters:
Boris Bokowski (FU Berlin)
Steve Easterbrook (NASA/WVU)
Wolfgang Emmerich (City Univ, London)
Christian Kanele (FU Berlin)
Jyrki Kontio (Univ. of Maryland)
Stefan Tai (TU Berlin)
Scott Tilley (SEI)
Will Tracz (Loral Federal Systems)
Ken Wong (Univ. of Victoria)

Photographer: Keng Ng (Imperial College)

Technical Support: Sabine Lembke, Lutz Friedel and **Stephanos Mantoulidis** (TU Berlin)

Technical Coordination: Ingo Claßen (Fraunhofer ISST, TU Berlin)

Not So Small Ad

Tired? Stressed? Can't Sleep at night? Maybe you're the person we need for the WOW editorial team at ICSE-97. Contact Will Tracz (day or night), Senior Programmer, Advanced Technology, Loral Federal Systems, 1801 State Route 17c, Owego, NY 13827-3998. Tel: (607) 751-2169. Fax: (607) 751-6025. Mail Drop 0210. Email: tracz@lfs.loral.com

Distinguished Fellows

...continued from Page 1.

were Lori A. Clarke, Andy Podgurski, Debra J. Richardson, and Steve J. Zeil, for their paper "A Comparison of Data Flow Path Selection Criteria".

In a surprise move, one of the afternoon's awards was held back. Barry Boehm described the search for a recipient for the IEEE Software Process Improvement Award. Previous winners included the Software Engineering Laboratory at NASA Goddard / University of Maryland, and Raytheon Company, Lexington Massachusetts. This year, it was felt that none of the nominees matched up to the standards set by these past winners. Boehm encouraged people to seek out worthy nominees for next years awards. Send e-mail to watts@sei.cmu.edu for instructions on how to make a nomination.

Most Influential Paper Awards – Past Winners

1) Marc J. Rochkind - The Source Code Control System.
2) William A. Wulf, Ralph A. London, Mary Shaw - An Introduction to the Construction and Verification of Alphard Programs.
3) David L. Parnas - Designing Software for Ease of Extension and Contraction.
4) Walter F. Tichy - Software Development Based on Module Interconnection.
5) Mark Weiser - Program Slicing.
6) Sol J. Greenspan, John Mylopoulos, Alex Borgida - Capturing More World Knowledge in the Requirements Specification.
7) David L. Parnas, Paul C. Clements, David M. Weis - The Modular Structure of Complex Systems.

Welcome to Boston

By Rick Adrion

Mark 18-23 May, 1997 on your calendar and call your travel agent. This is the one week of spring in Boston and ICSE will be there to celebrate (actually, spring is a little longer, but the end of May is delightful). Moreover, ICSE-97 has an expanded programme with special activities to attract practising professionals and researchers. With the assistance of the Massachusetts Software Council, the Boston Computer Society, and the local ACM and IEEE/CS chapters, ICSE-97 will have a strong tutorial programme. A first ever commercial exhibit will complement the research demonstrations. A doctoral symposium and a poster session will showcase new ideas. Organizational overviews will provide insight into industry progress and industry realities.

ICSE-97 is the 19th conference in the series and the first to use the year in its name, following a trend in computing conferences. ICSE-97 will build on the success of the Berlin conference and play on the strength of the computer industry surrounding Boston.

Come to Boston next year for ICSE-97, visit the Computer Museum, walk the Freedom Trail, catch a Red Sox game, view the "Big Dig," and generally enjoy springtime in the "Hub of the Universe." A great conference in a great city – continuing the ICSE tradition.

Letters to the Editor

It seems to me Bashar, that ICSE is a mecca for people with little or no manners to the various presenters within sessions. Have any presenters complained about the mass of audience movement even half way through their talk? Even though ICSE is quite prestigious, I'm glad I'm not talking here – I would be too tempted to act like a demonic schoolteacher and bark at mid-talk interruptions. Your news crew should do an audit to see how that would go down!

Regards,
Paul Fitzhenry
Cambridge University

Dear WOW-Team,

Nice job on the WOW newsletter (again)! Its great to have a sense of some of the ICSE happenings even when one is on a different continent.

Gail Murphy,
gmurphy@cs.washington.edu

Competition Winners

There were twenty entries for the competition to compose the most bizarre title that could be submitted to ICSE 10 years from now.

After a chaotic, non-repeatable, non-defined, but highly optimised selection process, the ICSE-18 WOW editorial board chose Steve Zeil's submission entitled "Implementation Considered Harmful: Avoiding Liability Litigation by Refusing to Deliver a Product". Congratulations, Steve – your champagne is on its way.

The two runners up were Adam Crow's entry "ICSE-2008: Why is Windows '99 so late?", and Will Tracz's proposed submission, "Time Travel Approaches To Maintaining Legacy Software Systems".

Finally, we had to disqualify an anonymous entry entitled "I Was Wrong About Software Engineering" by David Parnas, because the entry was not submitted by Parnas.

Disclaimer

By Guenter Kurth,
QA Manager, Siemens PSE, Austria

Regarding the paper "Engineering an 'Open' Client/Server Platform for a Distributed Austrian Alpine Road Pricing System in 240 Days: Case Study and Experience Report" by F. S. Oberpfalzer, T. Grechenig, and S. Biffl, delivered by Mr. Schönbauer:

- T. Grechenig, S. Biffl, and Mr. Schoenbauer were not involved in the mentioned project. They have no connection to Siemens Austria.
- The opinions expressed in the talk and the paper are their personal opinions and are not authorized by Siemens.

Rebuttal

By Thomas Grechenig

Correct: Mr. Schoenbauer was not involved in the project (but he helped the authors to respect the event). Statements in the talk of Dr. Schoenbauer were for sure his personal opinion and not the official opinion of Siemens

Incorrect: "Dr. Biffl and Dr. Grechenig were not involved at all in the project. They have no connection to Siemens Austria".

Parnas Quotes

On software metrics:

You know a good program when you see it.

On curriculum planning:

You wouldn't want a course in electrical engineering, so why would you want a course in software engineering?

reuse, and quality in various dimensions are increasingly valued.

To deal with these challenges, many of the hard-won lessons of traditional software engineering are still important: modularity, information hiding, early error elimination, closed-loop planning and control, and configuration management. But they need to be re-orchestrated with emerging techniques – (RAD, JAD, QFD, OOA, OOD, spiral models, cleanroom methods, etc.) – and tailored to each organisation's culture, priorities, and boundary conditions.

Here are some examples of issues that are worth rethinking:

– In a COTS capabilities-defining-requirements situation, what is the relevance of the traditional "complete, consistent, traceable, testable" requirements specification? What should replace it?

– What is an effective sizing metric for a project involving GUI builder & extensive COTS integration effort? Neither lines of code nor function points capture the phenomenology very well.

– How does one V&V the behaviour of a bunch of autonomous automated agents roaming around in cyberspace?

There are good many more such issues. Resolving them will require the best from both software engineering practitioners and academics. Bringing these communities together has been the particular strength of the ICSE conferences. I look forward to ICSE-97 and its successors helping us prepare for the big software challenges and opportunities of the 21st century.

Errata: *In yesterday's WOW, we misspelled the URL for Barry Boehm's web page on the Win-Win approach. The correct URL is http://sunset.usc.edu/*

Top-selling Books at ICSE-18

By Christian Kanele

The three best-sellers at ICSE-18 from Kluwer Academic Publishers were:

(1) "Software Performability: From Concepts to Applications", by Ann Tai, John Meyer and Algirdas Auizienis.

(2) "Tools And Environments For Parallel and Distributed Systems", edited by Amy Zaky and Ted Lewis.

(3) "Practical SGML", by Eric van Herwijnens. Second edition. [This was also ICSE -17's third bestselling book from Kluwer].

The ICSE-18 Virtual Press Conference

By Wolfgang Emmerich

"The German information highway enables me to attend the ICSE-18 press conference while being in Bonn, but unfortunately it does not allow me to have a beer with my friends that I have known for so long", moaned Prof. Dennis Tsichritzis, chairman of the German institute for mathematics and computer science (GMD).

Tsichritzis and parliamentary secretary Ms. Cornelia Yzer attended the virtual ICSE-18 press conference on Wednesday from a GMD studio in Bonn. The conference amply demonstrated the benefits of new multimedia techniques to about 10 journalists who were no less impressed than I was. Yzer used charts that grew as walls from the ground of the virtual studio. The virtual reality of the studio was mixed with digitised video data and then sent as a number of packets through the ATM connection and was finally projected in Berlin.

Like Yzer's welcome address in Wednesday morning's plenary session, the virtual conference was enabled by a high-bandwidth ATM connection with 34 MBit/sec connecting Bonn and Berlin. This connection was established as part of the federally funded POLIKOM project whose purpose is to establish a communication infrastructure for governmental organisations that will be split between Bonn and Berlin in the next century.

In her presentation, Yzer outlined the current funding programme for software engineering research in Germany. Unfortunately, the facts she presented were less fancy than her way of presentation! The current programme provides no more than 15 Million DM of funding per year. This highlights a serious mismatch between federal funding policy and the strategic importance that Yzer considered software engineering to have for the German economy.

Feast for thought

By Manny Lehman

Though not currently occupying centre stage in academic computer science and software engineering research, process improvement is a focus of interest and investment of software development and maintenance organisations. Many of these have reached the conclusion or have been given to understand that a potential of two or more orders of magnitude improvement should be attainable from new approaches to the software process and its support by improved methods and tools. Improvement, in this context, refers to such characteristics as product quality, process productivity and process responsiveness.

In response to this need, or possibly, the driving force behind it, models and associated procedures of which CMM and ISO-SPICE are the best known, have been developed. Despite serious shortcomings and no convincing demonstrations of return on investment, many industrial activities implementing improvement programs based on such procedures are underway.

Without imputing any judgement of the absolute or relative value of these approaches, this brief note draws attention to a serious weakness common to them all. Feedback clearly plays a major role in global software evolution processes, where evolution includes both ab initio development and so called maintenance. The adjective global is used to include not just technical development steps but all relevant activities of corporate executives, managers, system engineers, marketeers, salesmen and others. All influence the process and its product in one way or another.

The software process is and must be treated as a feedback system. Now a general property of other systems with negative feedback mechanisms is that the impact of forward path changes on externally visible behaviour is reduced in inverse ratio to the gain in the feedback loop. System behaviour is stabilised and the impact of change is much reduced. Should we not expect this to be so for the software process? The FEAST (Feedback, Evolution and Software Technology) project hypothesis suggests that this is indeed so and that the continuing difficulty in achieving major software process improvement may be due to the failure by current procedures to model and treat the process as a feedback system.

Attendees at ICSE-18 and others will recognise that exploration of this hypothesis, its implications and its exploitation potential is an important issue with a potential for major pay off. I commend it to your attention.

Agony Column

Dear Crabby,

Tom DeMarco told us to concentrate on the essence, not the accidents. I tried to explain that to the police lieutenant whose car I rear-ended, but he didn't understand. What should I do now?

Saddened in Spandau

Dear Saddened,

You are a prime candidate for process improvement. Driving lessons are a key process area at your level of maturity.

Crabby

Software Engineering and ICSE Futures

By Barry Boehm

The next century may well be called “the software century”. Organisations competing in product lines, services, or national defence will find that the excellence of their software engineering efforts will be one of their most critical success factors.

Meanwhile, the twin paradigm shifts of COTS (commercial-off-the-shelf) software and cyberspace are shaking traditional software engineering methods to their roots. COTS software is causing a 180 degree shift in the traditional software development cycle: from requirements-determining-capabilities to capabilities-determining-requirements. Cyberspace is changing the nature of software applications, and their development, from individual-oriented activities to networked-group activities.

Concurrently, competitive pressures are re-prioritising software objectives. Maximising functionality per dollar is no longer adequate. Rapid cycle time, product line-based

Industrial Authors at ICSE

By Jyrki Kontio

Many conference attendees have been impressed by the high number of industrial participants at ICSE-18. Many industrial participants we talked to pointed out that ICSE is a good general conference where it is easy to keep up with recent trends and developments in the field.

While the number of industrial participants is high, there are much fewer industrial authors at the conference. The lack of industrial contributors has sometimes been criticised as a sign of ICSE's "deviation from reality".

We talked to Marvin Zelkowitz, the conference programme co-chair. There were several industrial paper submissions but many of them had to be rejected. "Some papers from industry seemed to have been written as advertising brochures, they repeated the company name whenever possible and lacked a critical discussion on the proposed approach", Zelkowitz said. Also, industrial papers made few references to other similar work, they seemed to assume that their approach and results were unique without proper justification and they rarely cited the origins their work was based on. It seems that industrial authors are not familiar with the scientific writing and style required for ICSE. "Clearly, the rewards structure in industry works against publishing papers", Zelkowitz said, "people working in industry have different priorities, their careers do not depend on papers". At the same time, industrial experiences benefit the academic community in important ways, they highlight issues that are important in practice and may give examples or ideas to other industrial participants. It is important for ICSE to encourage good quality industrial papers. Zelkowitz, summarised what industrial authors need to do with a few rules of thumb: "relate your work well with the state of the art and other similar work, do not reinvent everything but build on the work of others, and do not repeat your company's name too often, nor overuse company specific acronyms."

Nineteen industrial (co-)authors still made it through the review process. We tracked down three of them, who actually gave a presentation, to find out their motivation for an ICSE submission. It turned out that they all worked in industrial research centres or units and publishing of results is considered a part of their jobs. "Publishing papers is built into the reward structure in my organization" said Jean Mayrand of Bell Canada. Frances Paulisch of Siemens pointed out that her company expects her to publish papers. "Our paper also represented an opportunity to work together over organisational boundaries on an interesting topic", she added. Prem Devambu also cited as similar reasons that AT&T Bell Labs expected him to publish papers.

It is a good sign that some industrial organisations are encouraging publishing papers, but the lack of papers from "front line" software engineers is alarming. It seems that people in software development projects do not have the time nor motivation to contribute to ICSE. Fortunately this problem is being addressed. Industrial papers are sometimes "mentored" through the publishing process and industry representatives are invited to give less formal presentations.

ICSE-18: An Industrial Perspective

By Volker Gruhn

The ICSE-18 programme looks familiar to ICSE participants at first glance. It seems to be dominated by academics presenting research results to each other. A closer look, however, reveals that several successful steps towards industrial involvement have been achieved. Case study presentations on industrial experience sessions are a step forward.

An even more substantial change is that almost 50% of the participants were from industry. What are we industrial people doing at ICSE? Are we trying to sell our systems and services to each other? Are we looking for immediately applicable methods and tools? My impression is that this is (fortunately) not the case.

Instead of doing so, many people actively participate in bridging the gap between industry and academia. We know the phenomenon that industrial people complain about "their" problems not being tackled by current research, while academic researchers complain about industrials not understanding the benefits of "their" research. ICSE-18 provides a forum to establish and/or strengthen relationships between researchers and technology users. Their personal contact could form a basis for the technology transfer we urgently need. Whatever means are taken to make technology transfer happen, the personal relationships between researchers and industrial software developers are *the* key success factor for technology transfer.

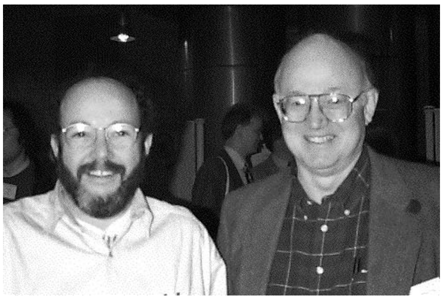
To sum up, I appreciate the ICSE-18 step towards support of communication between software engineering research and practical software engineering. Of course, ICSE must remain an event with a strong technical focus, but the approach to involve industry seems promising for both, researchers (because they get in touch with today's problem in industrial development) and industrial software developers (because they have the chance to benefit from recent trends in software engineering research).

Easterbrook, DeMarco, and AT&T

By Dewayne Perry and Larry Votta

First, Easterbrook's reporting should be relegated to the editorial page or perhaps the paper should have a yellow tinge! Neutral reporting would be more appropriate. DeMarco's talk was provocative enough without the reporters over zealousness for controversy.

Second, the comments on process by DeMarco are consistent with the goals of good process work and research. Not only that, his comments actually point up the necessity and importance of process work. Yes, the easier things have been automated (the accidents, if you will), leaving the harder parts (the essentials) to be done by the people. These hard issues are precisely where process is essential: the methods, techniques, and heuristics to manage the inherent complexity of software systems as they are developed and evolved. That we are not doing that job as well as we might means that we have more work to do, rather than that we should fold up our tents and go home. Yes, processes are often inflexible. That means we have done a poor job defining those processes. Yes, organizations are often averse to risk. But, that was true long before the advent and current emphasis on software process. Organizations, especially large ones, are often extremely conservative. Just try injecting some research results into an organization that is overworked, understaffed, and has to meet unrealistic deadlines. Especially since development management is often rewarded on its ability to meet schedule, not on its ability to take risks. We will also grant that the aversion to risk is driven by the logic of expediency (skipping process steps so you can get the product out fast) is a serious process problem that needs to be solved BUT, it is not primarily the result of process work. Yes, processes often become ends in themselves. There is nothing new in that (you need only look at your local government organizations). But, process work, like any other discipline is a means to an end. In the case of process it is the end of producing software according to cost, quality and interval goals established by management and marketing strategies. Process is an essential ingredient in the effective engineering of software systems.



***Votta and Perry**: Process is essential.*

Third, AT&T's breakup (for a second time) is undoubtedly a stressful situation, especially for those at the boundaries of the split. AT&T as of January 1, 1997 will be three companies; AT&T (telephony services), Lucent Technologies Inc. (systems and technology), and NCR (a computer company). Indeed there are two classes of people at those boundaries: those who are no longer wanted and those who are wanted by all sides. But for the majority of people it has been business as usual. However, we can attest to the stress of being at the boundaries of the breakup. Fortunately we (and all software researchers) are in the class that is wanted by both sides, so the stress has been more the result of uncertainty and the need to make career decisions with little information. In the end we should have two viable research organizations instead of one and thus two research organizations with reality based software engineering research.

Myths, Methods, Rituals and Software Engineering

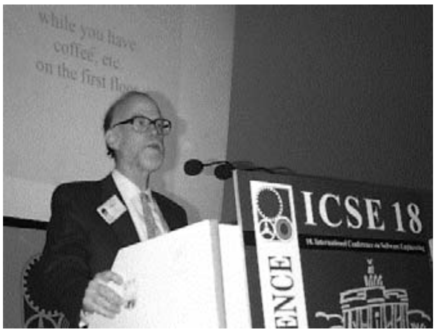
By John Musa

I hope that as we search for solutions to problems facing the software engineering profession, we take a broad cultural and historical perspective. Whether or not you agree with Tom DeMarco's thought-provoking insights, I think his most important message is that we look outside of ourselves to world history and culture for possibly instructive analogies.

This particularly struck me in a visit to Pergamon museum. I was noting the problems that forced societies 2000 to 2500 years ago and how they dealt with them when I realized some of the parallels we are using (perhaps unconsciously) today. Although those societies had high levels of intelligence and cultivation, their accumulations of scientifically-based knowledge were rather small (sound familiar?). They were confronted with many serious problems that cried for immediate solution.

In this environment, anyone who could offer a structured model of how things worked (read "myth" if you are an archeologist) was welcomed with immediate belief and even worship, which quickly led to a ritual. How many times today do we jump to a model of a pressing problem without critically examination, testing, experimentation, or measurement? What we think of as methodologies are actually mythodologies. Mythodologies will not stand up to scrutiny, hence they must quickly be supported by ritual or they will die and will be faced with the uncomfortable realization that we have a continuing unsolved problem.

It might be a refreshing exercise if we ask ourselves what our rituals and mythodologies are.



***Tony Hoare**: They didn't need it.*

ing requirements analysis. So the most successful applications of formal methods have been to requirement specification and capture, which then provides a basis for proving correctness of design and code. These techniques should be supplemented by reviews and inspections, instilling a reliability culture and identifying opportunity for further use of formal techniques to explore possibilities and assure completeness.

Hoare went on to say that his second important contribution has been effective testing. Testing is not like insect control: as you detect and remove initial bugs, the rate of discovery of new bugs goes down quickly, perhaps depending on a software system's execution paths.

Software engineers sometimes over-engineer their code to ensure that it will be reliable. For example, they clone new code, use defensive programming, and check the plausibility of data.

These three engineering facets – management, testing and over-engineering – are supplemented by software research that have been highly successful. Structured programming is now an accepted practice, as are the use of data types, strict type checking, and information hiding. These techniques have a basis in formalisation and abstraction.

According to Hoare, it is important for researchers to stay well ahead of current practice. The duty of the researcher is to prepare the basic understanding that may be needed to deal with the unexpected challenges of any possible future development. It is the practitioner's responsibility to identify relevant research and adapt it to widespread application. Part of the reason for the large gap between research and practice that is preventing this technology transfer is that many practitioners studied neither formal methods nor computer science; this educational gap could be filled by in-service education. Additional needed talents are understanding of the application area and marketplace, intuitive knowledge of the customer, a deep commitment to quality, and more, plus toolsets to support them.

Hoare concluded that pure curiosity should guide future research. He has sought to unify the study of many different kinds and styles of programming. He believes that theory should be simpler than any paradigm it encompasses; as a result it will be too simple for direct application.

Thus, there are many strategies: improved management, testing, over-engineering, methodology, languages and tools. Their choice and mixture depend on cost and practicality.

Reactions to Hoare

A number of delegates expressed strong views on Hoare's keynote presentation, but refused to go on record. Chairman of the ICSE steering committee David Notkin expressed his disappointment that this was the case.

Nevertheless, WOW *did* manage solicit some feedback from ICSE delegates. Prof. Herbert Weber expressed his reservations about some of Hoare's arguments. He felt that the relationship between scientific work and its industrial application was and should be stronger than that suggested by Hoare. He also asserted that many scientific contributions required significantly less than twenty years to find applicability in industry.

Other anonymous commentaries included:

- "I felt the talk was anodyne thoughts; there was some brief spark in the discussion of the technology transfer gap" – *anon*
- "For me, the highlight was the measured rebuttal of the use of experimental method in software engineering" – *anon*
- "Good talk" – *anon*
- "I was surprised, really surprised that he gave that talk. I didn't expect him to say what he said, but he did". – *anon*

- "I liked what I heard but I expected more" – *Wilhelm Schäfer.*

Agony Column

Dear Crabby,

I've heard that fly-by-wire aircraft use a lot of software. What is it that makes them fly?

Fear of Flying

Dear Fear,

A great deal of software supports a fly-by-wire aeroplane, much of it developed using formal methods. But it is an informal method that is actually responsible for the flight. The passengers, not having much confidence in formal methods, fasten their seatbelts and pray that the plane will take off. Their faith is uplifting, and it is that that causes the plane to be airborne.

Crabby

Small Ads

Lost: All Faith in Formal Methods. Last seen near Oxford.

For Sale: Kids Toolkit. Suitable for 8-12 year olds (in 20 years).

For Sale: Almost complete set of Petri Nets. Only a few tokens missing.

– ICSE-18 WOW! –

18th International Conference on Software Engineering

Window on the World

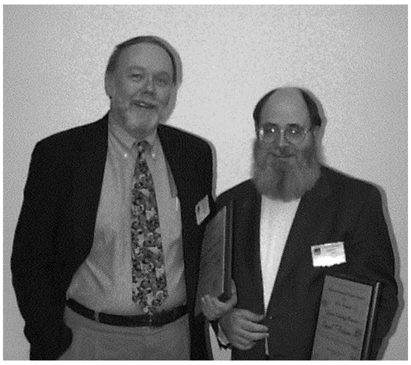
A Fraunhofer ISST Sponsored Publication **Volume 2 Number 3** March 29, 1996

ICSE Honours Distinguished Fellows

By Steve Easterbrook

At a packed plenary session yesterday afternoon, three distinguished researchers were recognised for their long-standing contributions to software engineering. Peter Freeman and Tony Wasserman were recognised IEEE fellows, while Rick Adrion received the ACM Service Award.

The ICSE tradition of awarding a best paper prize ten years on was continued by awarding a prize to the most influential paper from ICSE-8. The prize was awarded to Samuel T. Redwine, Jr. and William E. Riddle for their paper “Software Technology Maturation”. Marvin Zelkowitz, in presenting the award, described a difficult choice between this paper and the eventual runners up, who



***Most influential paper**: William E. Riddle and Samuel T. Redwine, Jr.*

...continued on page 8

Hoare Reflects

By Bashar Nuseibeh and Shari Lawrence Pfleeger

In the second keynote of ICSE-18, Prof. Anthony Hoare reflected on his contributions to computer science research during the last twenty years, within the context of software engineering practice.

Twenty years ago, Hoare asserted, it was reasonable to predict that the size and functionality of requirements would be severely constrained by reliability requirements. The software engineering community responded by directing its research at program correctness based on mathematical proof. In telecommunications, aircraft, and compilers, millions of lines of code run reliably and satisfy customers. As with many branches of engineering, software engineers have continued to learn from their failures. They apply many techniques that have proven effective in other engineering disciplines. Although formal methods and proofs play a very small role in large-scale programming, they provide a conceptual framework for current practice and future improvement.

Nearly all failures and cancellations of software projects are due to bad management, particularly dur-

...continued on page 2

Reminder:

Daylight saving time begins on Sunday. Clocks go *forward* by 1 hour at 2am. If you’re travelling on Sunday, don’t forget to adjust your alarm clock!!

Farewell

By Dieter Rombach

I hope you all enjoyed your participation in the 18th ICSE in Berlin. I believe that the wide range of presentation topics and views, as well as the widespread backgrounds of attendees have stirred up many useful controversies and stimulated many constructive discussions.

The purpose of any flagship conference is, after all, not just to converse with your peers with whom you have agreed with for years, but to get exposed to other members of the community and their views. Immature disciplines fight ideological wars between opposing views; mature disciplines use them to create synergy. I personally believe that with ICSE-18 we have made a small but important step into the direction of unification.

I want to thank all involved in making this conference a success, from organizing to local arrangements committees. Special thanks to the newsletter team for having put together ICSE-18 WOW.

I wish all of you a safe trip home and hope to see you again in Boston.

CONTENTS

■

CHECK OUT the *Window on the World* on the **World Wide Web: http://www.cs.tu-berlin.de/~icsewow**