

05

September 2003

THE TESTER

- **How Open-Source and Commercial Software Compare**
- **The Use of Precise Specification in Testing**
- **Adventures in Session-Based Testing**
- **Outsourcing and Risk Management**
- **Use of Mindmaps in Testing**

N E X T C O N F E R E N C E

Quest for the Best Test

Thursday 18 September 2003

FUTURE
SIGiST CONFERENCES

Tuesday 9 December 2003





FROM THE EDITOR

Well, a lovely summer and also signs of more testing business in the marketplace!

As you know, the SIGiST conferences are an excellent way of networking and if you are looking for employment there are nearly always companies who are looking for more testers at our meetings. These are generally verbally advertised in the networking/commercial break session, but you never know who you might be sitting next to at lunch time – could be a new opportunity!

If you are wanting to make an announcement in the networking session but for whatever reason do not want, or will not be available, to present it yourself then I will be pleased to make short announcements on your behalf. I will also place more information on the notice board.

The July session received very good assessments from the attendees, and our featured speaker Ruud Teunissen went down very well. You will note that at the September conference we will have two parallel sessions, one being a workshop necessitating the use of a laptop for more involvement from attendees.

Book now!!

Pam Frederiksen
Communications Secretary
Tel: **01483 881 188** (Leysen Associates)
Fax: **01483 881 189**
email: pam@leysen.com
BCS SIGiST website:
www.SIGiST.org.uk
SIGiST Standards Working Party:
www.testingstandards.co.uk

BCS SIGiST – QUEST FOR THE BEST TEST

Thursday 18 Sept. 2003 – London Marriott Hotel, Grosvenor Square, London W1

08:30	Coffee & Registration, Exhibition opens	
09:25	Introduction and Welcome – <i>Philip Trickey, Chair</i>	
09:30	Featured Speaker The Use of Precise Specification in Testing <i>David Parnas, University of Limerick</i>	
10:15	Coffee & opportunity to visit the exhibition	
10:45	Adventures in Session-Based Testing <i>James Lyndsay, Workroom Productions</i>	
11:30	Featured Speaker Constructing Precise Specifications for Use in Testing <i>David Parnas, University of Limerick</i>	Special Session 1 Getting a Grip on Exploratory Testing <i>James Lyndsay & Neil van Eeden</i> Advance booking see below Ends 12:35
12:20	Book Review	
12:35	Networking session and commercial break	
12:50	Lunch & opportunity to visit the exhibition	
14:00	Use of Mindmaps in Testing <i>Graham Freeburn, Newell & Budge</i>	Special Session 2 Constructing a Testable Specification <i>David Parnas</i> Advance booking see below End 15:00
14.45	Book Review	
15:00	Tea & opportunity to visit the exhibition	
15:30	Tips for Testing	
15:45	Outsourcing and Risk Management <i>Keith Klain, UBS Investment Bank</i>	
16:30	Closing Remarks	

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.

Special Session 1

The parallel session at 11:30 is a workshop with James Lyndsay. This is a hands-on workshop and discussion of exploratory testing methods. This will give participants an opportunity to actually do some exploratory work, rather than just talk about it. Participants will be required to bring their own laptop computer. A few small files will be provided by James on the day (CD, floppy, USB key) that will need to be loaded onto each laptop.

This workshop is limited to the first 20 applicants on a first-come, first-served basis. There is no additional fee. If you would like to take part, then please tick the box on the Special Session section of the enclosed registration form.

Special Session 2

The parallel session at 14:00 is a workshop with David Parnas, our featured speaker. Following on from his two morning presentations on Precise Specifications, David will guide participants to produce a Precise Specification from an informal description of a problem.

This workshop is limited to the first 20 applicants on a first-come, first-served basis. There is no additional fee. If you would like to take part, then please tick the box on the Special Session section of the enclosed registration form.

Signed copies of the Testing Practitioner

We have a few signed copies left of *The Testing Practitioner* by Erik van Veenendaal. The price is £25.00 plus P&P. Contact Claire Mason at the Admin Office if you would like a copy (Tel: 01422 836431).

Quest for the Best Test

Abstracts

David Parnas
Limerick University



The Use of Precise Specification in Testing

Abstract: Discussions of test results often degenerate into debates about what the project should do. Testers are severely handicapped by the lack of a complete and precise specification. This presentation will discuss what we could do with a precise specification.

Constructing Precise Specifications for Use in Testing

Abstract: Precise specifications can be constructed systematically and help to make many decisions that would otherwise be made by programmers and testers. This presentation will illustrate how precise specifications can be constructed using tabular expressions.

Workshop: Constructing a Testable Specification

Abstract: Participants will be given an informal description of a problem and helped to produce a precise specification of the sort illustrated in the earlier lectures.

James Lyndsay & Neil van Eeden
Workroom Productions



Adventures in Session-Based Testing

Abstract: Session-based testing can be used to introduce measurement and control to unscripted, open-ended test approaches and can form a foundation for significant improvements in productivity and error detection. The techniques are particularly helpful in controlling reactive, fire-fighting test teams, and in bringing agility and focus to exploratory methods.

Using two real-world case studies, this talk looks at the introduction and implementation of session-based testing. It details the session-based methods initially proposed, and notes problems, solutions and improvements found in their implementation – particularly focusing on risk, coverage, test case selection and

prioritisation. The talk also covers tools and simple metrics, recording sessions and the importance of feedback at a number of levels to allow process improvement.

Finally, the talk picks up on common team issues; responsibility and empowerment, coaching and skills transfer, and what it is like to work in a session-based test team.

Workshop: Getting a Grip on Exploratory Testing

Abstract: This is a hands-on workshop and discussion of exploratory testing methods. This will give participants an opportunity to actually do some exploratory work, rather than just talk about it. Participants will be required to bring their own laptop computer. A few small files will be provided by James on the day (CD, floppy, USB key) that will need to be loaded onto each laptop.

Keith Klain
UBS Investment Bank



Outsourcing and Risk Management

Abstract: UBS Investment Bank's (UBS) approach to outsourcing in the Quality Assurance department utilises a component based operational model for test execution, automation, and environment management. The primary objectives of the move towards outsourcing focused on cost control for operating the quality assurance and release management teams as well as meeting the increase in 2003 project requirements. Additionally, the quality assurance team had initiated a test process improvement programme that could be accelerated through the use of a 3rd party with test automation and non-functional testing expertise.

Building off the strength of its IT delivery reputation, UBS took a phased approach to implementation of the model and in the first and second quarter of 2003 several activities were run in parallel to prepare the programmes for remote team management and a flexible approach to staffing. Test process and release management were re-organised to a centralised team as well as knowledge management

for training and project documentation. Smaller, local test teams were created with an increase in focus on programme level test strategies whilst removing environment management as an internal function entirely.

Vendor selection was conducted over four months and included an RFI, RFP, reviews of existing UBS outsourcing vendors and a visit to vendor locations in Bangalore, Chennai, and Mumbai. Knowledge management, training on in-house technologies and applications, as well as specific testing delivery and management processes were included in the service level agreement. Full integration of the test model for all programmes will be completed by Q4 2003 realising a £600k saving including transition costs, whilst expanding test coverage by 60% and maintaining 90% of the permanent staff.

Graham Freeburn
Newell & Budge

Use of Mindmaps in Testing

Abstract: Mindmaps are powerful 'visual thinking' tools that have many applications in learning, personal development, problem solving, etc. Since Tony Buzan first introduced them in his book 'Use your head' in 1974 they have helped millions of people to improve their mental skills and abilities. They have many potential applications in the field of software testing and using examples of mindmaps he has built and some provided from other testers who use the techniques, Graham Freeburn, who has used them for many years; will demonstrate how this powerful technique can be applied.

The presentation will:

- Introduce you to mindmaps – what they are and how they work
- Provide examples of their use in testing – from ISEB revision to Test Strategies
- Demonstrate how excellent software is now available to help the non-artists among you to use them
- With your help, will interactively build a mindmap of the "Top 10 problems of Test Automation" in the session

How Open-Source and Commercial Software Compare: A Quantitative Analysis of TCP/IP Implementations in Commercial Software and in the Linux Kernel

We provide an automated software inspection service that is used by leading commercial software vendors to identify defects and provide metrics regarding the quality of the inspected code. This inspection service is based on a combination of technology and a repeatable process, and enables us to maintain a database of metadata about code quality. This database provides a unique opportunity to independently assess the quality of software.

What is Open Source and why might it be better?

Most commercial software vendors distribute their products in the form of executable or object code. Their customers do not acquire a licence to use the source code, so they cannot change or extend the functionality of the executables, except by specific arrangements with the vendor. They are generally prohibited from redistributing a changed or extended version to others. With few exceptions, customers of commercial software vendors must rely on the vendor to make changes and extensions.

Open source software represents a fundamentally different way in which software is developed, sold, and maintained. For example, the source code can be modified by many people without the need for those people to be employed by the same software vendor.

Open source proponents believe that, for important pieces of software, the open source model encourages several activities that are not common in the development of commercial code:

- Many users don't just report bugs, as they would do with commercial software, but actually track down their root causes and fix them.
- Many developers are reviewing each other's code, if only because it is important to understand code before it can be changed or extended. It has long been known that peer review is the most effective way to find defects.
- The open source model encourages programmers to organise themselves around a project based on their contributions. The most effective programmers write the most crucial code, review the contributions of others, and decide which of these contributions are incorporated into the next release.

- Open source projects don't face the same type of resource and time pressures that commercial projects do. Open source projects are rarely developed against a fixed timeline, affording more opportunity for peer review, and usually offer extensive beta testing before "release."

For these reasons, open source enthusiasts claim that the open source model produces better quality software than commercial software development.

Software inspection

Software inspection – the process of examining source code to identify defects – is a standard practice in development organisations and is widely recognised as the best way to find defects. Inspection is hardware-independent, does not require a "runable" application or a suite of test cases, and does not affect code size or execution speed. But until recently, it has been a manual process – very slow, and very costly – or tools-based and hard to implement effectively.

The majority of code inspections are performed manually. Although a human reading the code line-by-line can theoretically uncover the greatest number of defects, the process is slow, painstaking, and fraught with inconsistency. Also, this approach does not scale to handle today's multi-million line applications. As a code base grows, the cost of a complete manual inspection becomes prohibitive and the volume of code is intimidating to developers. In practice, manual inspections are only performed on subsets of the source code.

Inspection tools are able to perform only a portion of the inspection process, requiring significant further manual review. The inspections tools generate a large volume of defect 'warning messages' many of which are, in fact, false positives. The inspection tool "thinks" it has found a defect, but a deeper manual analysis of the context shows that the reported issue is not actually a defect. This false positive problem is very severe. Frequently, the rate will exceed 50 false positives to each true positive; in other words, only 2% of the warning messages represent defects.

Reasoning's automated software inspection service

Our automated software inspection service provides many of the benefits of a manual code review in significantly less time and at dramatically lower cost than manual inspection or internal use of inspection tools. With the service, in-house resources are not diverted from current development projects. We identify defects that cause application crashes and data corruption, and provide actionable reports. The error classes in C and C++ include:

- Memory leak: *Reference to allocated memory is lost*
- NULL pointer dereference: *Expression dereferences a NULL pointer*
- Bad deallocation: *Deallocation is inappropriate for type of data*
- Out of bounds array access: *Expression accesses a value beyond the array*
- Uninitialised variable: *Variable is not initialised prior to use*

The output of the inspection is a set of reports that:

- Make defect analysis fast and simple by identifying the location and describing the circumstances under which the defects will occur;
- Identify the parts of the code with the greatest risk, enabling the development organisation to focus QA and testing resources where they are most needed;
- Compare the customer's code quality with a benchmark (related to other inspections done by us).

The Study and Methodology

Of the thousands of open source applications available, we chose the Linux, operating system. This general-purpose operating system has been under development for nearly a decade, is widely used and is actively maintained and enhanced by a community of thousands of programmers.

However, comparing the quality of several entire operating systems is a challenge, primarily because the size, scope and goals can be so different. Instead, we chose a common function implemented by all the projects in our study, the TCP/IP network

protocol "stack". There were several reasons for this decision. This protocol is usually in the operating system "kernel", the lowest level software in the system; thus defects can have a major impact, including inability to communicate, system crashes, network outages, and security violations.

Each project was inspected using our standard automated software inspection process.

The commercial projects

We have conducted inspections of five different commercial TCP/IP implementations, including implementations from both general-purpose operating systems and embedded applications.

Four of the five implementations are considered mature, having been in commercial use for over ten years (although the TCP/IP code is under active development). The fifth

is relatively young: it was started about three years ago.

The size of these projects ranges from 64 KLSC to 269 KLSC. For reasons of client confidentiality, we cannot disclose further information about these projects.

The Linux inspection

We inspected the TCP/IP implementation in version 2.4.19 of the Linux kernel. We chose this version because it was the latest "stable" release at the time of the study. The TCP/IP code was inspected in isolation from the rest of the kernel, using the exact same process we use for customer projects.

The open source TCP/IP implementation includes 166 source files with just under 82 thousand lines of source code (KLSC) in size, not including user include files, header files, blank lines and comments. We found 8 defects, resulting in a defect density of 0.10 defects/KSLC.

Comparison Results

The table below summarises the results for the five inspection classes.

Error class:	Commercial:	Open Source:
Memory leak:	43	1
NULL pointer dereference:	128	3
Bad deallocation:	0	0
Out of bounds array access:	9	3
Uninitialized variable:	132	1
Total:	312	8

Note that there are no bad deallocations. However, since the applications are generally fairly mature and all are written in C rather than C++, this is not particularly surprising. Bad deallocations that occur in C are generally beginner's mistakes (much more so than the other defect classes), and tend to happen on the path the code is intended to take, so there



ICSTEST®-UK 9th-10th September 2003 Café Royal, London

The largest and best-attended UK conference on Software Testing and Quality Management looks to the future by examining the latest trends and issues - with tracks on Testing Mobile Applications, Testing in the Real World, Test Management and Non-Functional Testing.

Presenters from world leading companies will share real-life experiences and latest techniques, including:

- Microsoft
- Nokia
- Symbian
- Sony Ericsson
- EDS
- Virgin Money
- Vodafone
- IBM
- Deutsche Bank
- National Software Testing Labs
- Royal Bank of Scotland
- University of Hamburg
- ABN Amro
- Marks & Spencer

Day **1**
Day **2**

Day one of the conference opens with a keynote from John Noakes, .NET advisor from Microsoft and will cover 'Quality and Testing in Converging Technologies'.

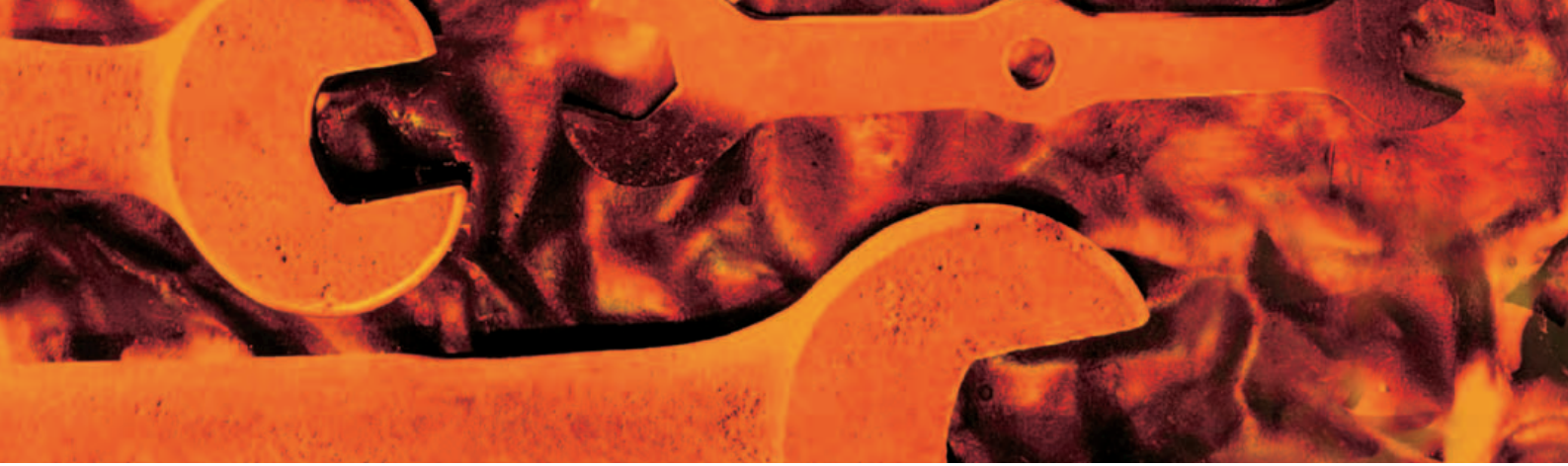
Day two opens with a keynote from Christa Menke-Suedbeck, CIO of Deutsche Bank covering 'Smart Sourcing', a new way of managing and delivering software testing.

The conference also has four tutorials delivered by leading specialists on Monday 8th September covering:

- Test Strategy and Planning
- Centralised Testing
- Non-Functional Web Testing
- Effectively Managing User Acceptance Testing

...If you can only attend one conference this year - make it this one...

**Register online: www.icstest.com
or e-mail: katharine.smith@icstestuk.com or Tel: 01483 733 109**



is a large likelihood they get caught quickly.

In light of the relative maturity of the commercial applications and the open source model of the Linux code, perhaps one should be surprised that any defects remain at all. Given the amount of testing that all the code bases had undergone before the inspections, this also confirms that testing is not enough: inspection finds defects that escape testing.

Of course, the five commercial applications together contain much more source code than the one open source application. Therefore it makes much more sense to look at defect densities.

Feedback from the developers on Linux inspection

We submitted the details to people on the kernel networking list and have received the following feedback so far:

- The memory leak is a real defect. Independently of this inspection, it has been fixed in version 2.4.20.
- The out of bounds array accesses are not real defects, because the kernel would not work if they were.
- The uninitialized variable is not a defect. This is code implementing a tiny interpreter, and the uninitialized variable represents variables in the interpreted language. These variables have random values when the interpretation starts, and it is the responsibility of the interpreted program to initialise the variables before they are used.
- We have not received definitive feedback on any of the null pointer dereferences.

In summary: one defect is real, 4 defects are not real, and 3 are undecided.

Defect repair comparison

Since those most familiar with the application are best able to determine the need to repair any individual defect, the most reliable metric is which defects need to be fixed according to the developers or maintainers of the code.

The table below reflects the reported defects, the repaired defects, and the defect density (defects/KLSC where KLSC stands for Kilo Lines of Source Code) for the commercial projects and the open source project. Since we have not yet received feedback on many of the defects reported to the Linux kernel maintainers, the real number for the open source code may be higher.

	Reported:	Repaired:	Size (KLSC):	Reported/Size:	Repaired/Size:
Commercial:	312	235	568	0.55	0.41
Open Source:	8	1	81.9	0.10	0.013



Leading the way in specialist testing

Consulting

Through early involvement in testing programmes our experienced consultants have a proven track record in delivering significant business benefits to our clients.

Recruitment

A tailored flexible service for all your test recruitment needs. Our expert team take care to understand your exact requirements to ensure we provide the right people at the right time.

Education

We provide ISEB accredited courses, specialist industry workshops and seminars to motivate, retain and develop testing staff, thereby enhancing skills and productivity.

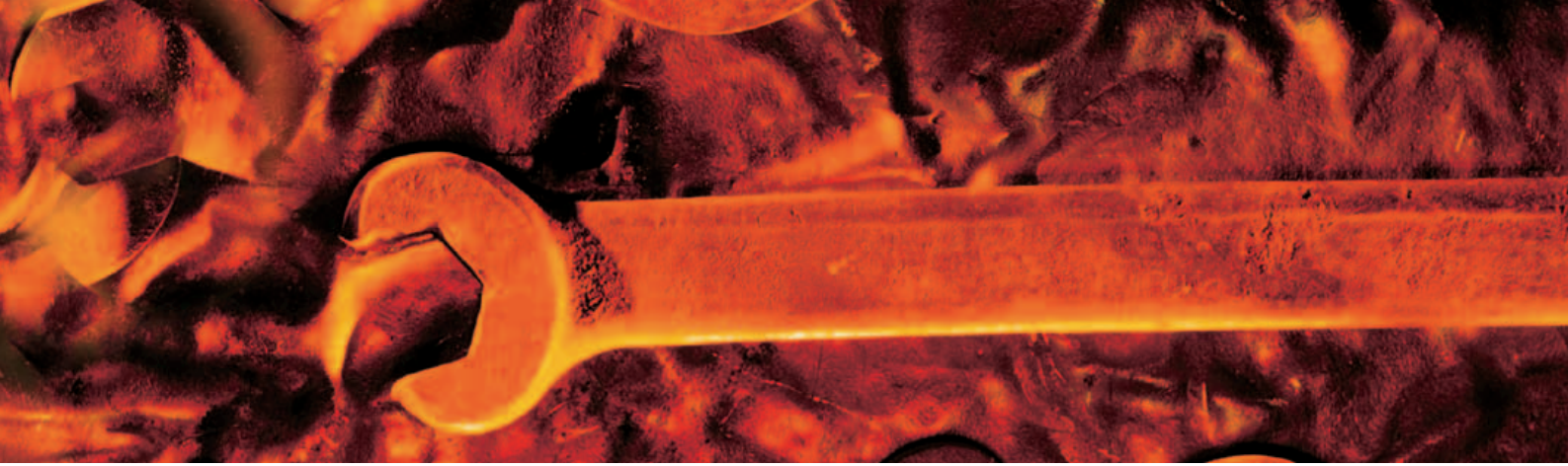
Managed Services

By combining our capabilities we offer a unique solution to the requirement for flexible levels of skilled testing resource. This enables improved planning and cost reduction.



For more information please contact: T: 01293 44 00 22 E: info@missiontesting.com W: www.missiontesting.com

Mission Testing is part of The Capita Group Plc.



On average, both the reported and the repaired defect densities are higher for the commercial implementations compared to the open source implementation.

Conclusions

This study compares five commercial implementations of TCP/IP, the fundamental protocols underlying the Internet, with the TCP/IP implementation in version 2.4.19 of the Linux kernel, an open source general purpose operating system.

The open source implementation of TCP/IP in the Linux kernel exhibits significantly lower defect density when compared to the five commercial applications and falls within the "Best Third" of source code projects inspected by Reasoning.

About Reasoning Inc

Reasoning is a leading provider of automated software inspection services that helps development organisations reduce the time and cost involved in finding software defects. The company's business is focused on organisations that develop C and C++ applications. Reasoning is headquartered in Mountain View, CA, USA. The full Linux inspection report can be downloaded from URL <http://www.reasoning.com/downloads/inspectionreport.html>. The full comparison paper is available at <http://www.reasoning.com/downloads/opensource.html>. For further discussion about these results and/or the Reasoning service, please contact Rix Groenboom (rix.Groenboom@reasoning.com).

Membership of SIGiST

At its meeting in February, the SIGiST Committee decided that we would no longer run a membership scheme. If you are on our database, then you will automatically receive details of our events and regular copies of The Tester. If you do not receive The Tester you can add yourself to the database by going to <http://www.sigist.org.uk/cgi-bin/register.asp>. You may also change your details or unsubscribe at this site.

If you would like to receive information by post please contact the Admin Office at SIGiSTregs@aol.com heading up your email 'Mailing List'. Thanks.

This change means that there is now one flat rate for the SIGiST conferences with concessions only for academics and students. Everyone on the database will receive all other benefits hitherto associated with membership – access to the SIGiST library for instance.

Software Testing Courses
from UNICOM

ISEB Foundation Certificate in Software Testing*
23-25 September, 9-11 December 2003

ISEB Practitioner Certificate in Software Testing*
Dates on request

Software Testing Essentials
22 September, 8 December 2003

Risk, Testing and Project Intelligence
23 September, 8 December 2003

User Acceptance Testing
22-23 September, 11-12 December 2003

Systems and Acceptance Testing of UML-Based Systems
24 September, 10 December 2003

Software Quality and Testing for Project Managers
25 September, 11 December

Managing a Testing Project
22-23 September, 15-16 December 2003

Testing in a .NET Environment
3 October, 16 December 2003

Integration Testing Techniques
22-23 September, 2-3 December 2003

www.unicom.co.uk
For more information please call UNICOM on 01895 256 484 or email info@unicom.co.uk
*ISEB courses are presented by ISEB accredited trainers ElectroMind; e-Testing Consultancy, TSG

Advertising In The Tester

The Tester is distributed to the database of the Specialist Interest Group In Software Testing (SIGiST), which is part of the British Computer Society (BCS). With over 2,500 professional testers and IT professionals, we are the largest group of specialist testers and they need to know about your products and services. Additionally the The Tester is downloadable from our website www.sigist.org.uk and is accessed by IT professionals looking for testing related information.

The Tester can now offer you the opportunity to place your organisation in front of these specialists at a very affordable price as we can allocate 2 extra pages of the magazine to you.

The costs are £300 for a half page advertisement and £200 for a quarter page. To book your space for the next issue, please contact Claire Mason on 01422 836431 or email SIGiSTregs@aol.com. The advertising space will be allocated on a first come first served basis. The closing date for confirmation is 26 September 2003 and the artwork will be required by 3 October 2003.

If you require any help with artwork, this can be provided for a small charge. Technical details for the material will be provided when your space is confirmed. All costs exclude VAT.

For any other information please do not hesitate to contact Claire.

SPECIALIST INTEREST GROUP IN SOFTWARE TESTING

Next conference:

Quest for the Best Test

Thursday 18 September 2003 – London Marriott Hotel, Grosvenor Square, London W1

see page one for Conference Agenda

Registration Form

PERSONAL DETAILS

You may register by

Fax 01422 836 096 or 01422 839 472

Post SIGiST Conference Registration,
Marshwood Events Management,
PO Box 445, Triangle, HX6 3YF

Tel 01422 836 431

Email SIGiSTregs@aol.com (giving all details required below)

Title _____

First Name _____

Family Name _____

Invoice and Joining Instructions to be sent to (please indicate company name):

Company _____

Address _____

Postcode _____

Tel _____

Fax _____

Email _____

If you haven't heard from us by 11 Sept, please contact us on 01422 836 431

PARALLEL SESSIONS

Please indicate with a tick, one parallel session for each time slot
(see programme for further details):

11.30	14.00
<input type="checkbox"/> Dave Parnas	<input type="checkbox"/> Graham Freeburn
<input type="checkbox"/> James Lyndsay	<input type="checkbox"/> Dave Parnas

FEES

Including morning coffees, afternoon refreshments, luncheon, full set of presentation materials, and entry into the tools and services exhibition.

	Ex Vat	Inc VAT
Ordinary Delegates	£195.00	£229.13
Full Time Student* and Academics	£85.00	£99.88

*please inc copy of student ID (VAT @ 17.5%)

PAYMENT

By cheque made payable to 'BCS SPECIALIST INTEREST GROUP IN SOFTWARE TESTING', by bank transfer (await details on invoice) or by credit card.

VISA Mastercard Access Switch

Name on card _____

Expiry date _____

Issue number (Switch only) _____

Card Number _____

Amount _____

Billing address if different from first column: _____

Signature: _____ Date: _____

PURCHASE ORDERS

Does your company use Purchase Orders? Yes No

If so, please put the Purchase Order number here so that we can process your registration more quickly. If you tick the box above but do not know the Purchase Order Number we shall wait until we have one before processing.

Purchase Order Number: _____

CANCELLATIONS

Cancellations must be received in writing prior to 11 September to qualify for refund of fees (less £10.00 administration charge). No-shows are liable for the full cost of fees. Substitutions may be accepted at any time.

VEGETARIAN MEALS/SPECIAL DIETARY REQUIREMENTS

I am a vegetarian

I cannot eat: _____

CPD

The meeting is worth 5 hours CPD
(Continuous Professional Development)

Vat Reg No GB 618 1687 24. In the unlikely event of cancellation, our liability will be limited to the refund of fees.

Please tick this box if you DO wish to give permission for your name and address to be passed to a third party for mailings on related matters