Requirenautics Quarterly

The Newsletter of the Requirements Engineering Specialist Group of British Computer Society

 Issue 13 (January 1998)

RE-Locations

RE-Soundings

Editorial

Usually by the time I've finished putting RQ together I have very little time left before the printing deadline to write an appropriately witty editorial (what, you've noticed?). This time of course is no different. I was going to write about the perilous state of research papers in our field, a topic that has occupied me quite a lot in the past year, in my involvement in various conference programme committees. It just seems that there's all these papers out there, describing new methods, tools, what have-you. But whenever I try applying any of these in real projects, they bear so little relation to real life that the exercise is pretty futile. Geoff Mullery's columns over the past few years have set out many of the reasons why this is so, and in this RQ he sets more of what he'd really like to see in requirements tools.

But I'm still pondering why there are never any papers that describe a something that didn't work. The lack of published negative results in software engineering in general is, I believe a serious problem. Results are picked up and cited widely, without ever being replicated. That's why for instance, many software engineering books still cite the infamous 1979 GAO report, to show that only 4% of software projects deliver software that can be used as delivered. The study showed no such thing, of course. (To find out why not, read Tom DeMarco's book "Why does software cost so much").

So, to do my part to redress the balance, I'd like to actively solicit negative results, to be printed in RQ. If anyone out there would care to write up on any experience where some method, tool, process, standard, etc didn't work (assuming, of course there was originally some good reason to suppose it would), then I'd be delighted to include it in RQ.

Meanwhile, let not my digressions spoil your enjoyment of this RQ. It is, as usual, packed full of interesting events and articles. Read on...

Copy deadline for the next issue is 27th March 1998

Steve Easterbrook,
NASA IV&V Facility, Fairmont WV

Chairman's Message

The 30th September 1997 was a good day for the RESG. The group organised the First National Requirements Engineering Awareness Day (affectionately known as RE-Day), which over 200 people attended. The tutorial and workshop rooms were always packed, and the exhibition halls were bustling with people watching tool demos, and chatting away over an endless supply of tea and coffee. I really must express my sincere thanks and appreciation to the RE-Day committee, who worked tirelessly for many weeks to make the day the success that it turned out to be.

A sequel is planned for the September 1998 - code named: RE-2-Day. Please tell us what you'd like us to include in the programme.

RE-Day was actually not the last event organised by the RESG in 1997. In November, the group ran an informal afternoon 'debate' addressing the question: Can the RE Process be Quality Certified? Ian Alexander reports on that meeting in this issue of RQ.

The New Year brings another change to the constitution of the RESG Committee. Following his move from Philips Research Labs to Logica a few months ago, Mike Bearne resigned as Secretary to the group at last September's RESG committee meeting. Mike did a great job as Secretary for over a year, and his many contributions included the development and – at great personal effort - the transport of one RESG Soapbox to RE-Day. Apparently, the soapbox has now evolved into a coffee table in Mike's home, but he insists that it is fully customisable and can serve its original purpose should it be needed at future RESG meetings. Thanks Mike and good luck!

I am delighted to welcome Wolfgang Emmerich onto the RESG committee as its new Secretary. Wolfgang is a Lecturer at University College, London, and brings to the committee considerable experience in both requirements engineering and distributed systems. Wolfgang is already busy oranising a meeting on "Distributed Requirements Engineering" for 1998.

Finally, it's that time of the year again when, on behalf of the RESG committee, I would like to thank all of you - the RESG members - for your support during 1997. I hope you will be able to contribute to the group again this year by attending meetings, writing for this newsletter and sending us your comments and suggestions for process and product improvement.

Best wishes for a happy, healthy and successful New Year.

Bashar Nuseibeh,
Imperial College, London

RE-Treats

Forthcoming events organised by the group.

Industrial Experiences in RE

Wednesday February 4, 1998
Location: Department of Computer Science, York University
Time: 2:00pm to 5:00pm
Cost: free to members, £5 to others
Format: Invited speakers.

The activities of Requirements Engineering have been performed successfully - yes, people do actually construct systems that work, despite continued reports to the contrary - for many years. That is not to say that Requirements Engineering is not hard, and that there are not problems that need solving, but merely that such activities have been carried out long before the term 'Requirements Engineering' was 'discovered'. Why then is the question of Requirements Engineering still considered important? Well, it's considered important because there are still many ways that it can be done better - indeed, the last 10-15 years has seen a huge interest (academically) in the topic of Requirements Engineering. Despite this interest, one thing still remains, and that is that many companies are still using Requirements Engineering technology from the 70s and the 80s. Why is this so?

This session will explore the reasons why there is such a technology transfer gulf between the theory of the academic Requirements Engineer, and the practice of the professional. Speakers will try and identify the industrially relevant challenges for Requirements Engineering and explore why the current 'academic wisdom' does not provide sufficient means to address these problems.

For more information about the meeting, please contact:

Dr Andy Vickers, Praxis Critical Systems.

Email: ajv@praxis-cs.co.uk, fax: 01225 469 006

Tutorial: The Unified Modelling Language (UML)

Wednesday March 25th 1998
Location: Room 418, Huxley Building, Imperial College, London
Time: 1.30pm to 5.00pm
Cost: free to members, £50 to others
Format: Half-day tutorial
Tutor: Stephen Morris

Content, duration and scope: This half day tutorial aims to provide a thorough introduction to all aspects of the Unified Modeling Language (UML). The UML helps everyone to take steps towards standardisation of the representations used for object-oriented analysis and design, towards the development of tools and towards easier communication of object-oriented methods and research results.

Learning objectives: The tutorial assumes a general familiarity with object-oriented concepts and techniques. Participants can expect to learn the nature and status of the UML and its documentation, the detailed form of the notations that it includes and their underlying semantics, the likely impact on current working practices based on earlier types of representation and the availability of tools intended to exploit its potential.

Intended audience: The tutorial is aimed at three audiences:

a) practitioners in industry already using object-oriented methods or tools,
b) academics wishing to apply or teach up-to-date approaches to object orientation, and,
c) anyone working in the field of object-orientation who wishes their work to be more widely intelligible.

About the tutor: *Dr. Stephen Morris is a Lecturer in the Department of Information Science of City University, London, UK. He has been working over the past two years developing and using new teaching material based on the UML and incorporating it into existing approaches to object-oriented analysis and design. Dr. Morris is motivated by his research interest in the design and development of multimedia documents.*

How to Use Scenarios and Use Cases in the Systems Development Process

Thursday May14th, 1998
Location: City University, Senate Suite
Time: 1000-1630
Cost: £120+VAT per person
Format: A one-day symposium

There has been considerable recent interest in the use of scenarios and use cases in the systems development process. Defense procurement organisations use complex operational scenarios to acquire requirements for large, integrated systems such as frigates and submarines, and to guide acceptance testing of these systems when delivered. Both the OBJECTORY method and UML approach use use cases to acquire and validate interactive system requirements and to act as a starting point for object-oriented analysis, design and implementation. User interface designers have for a long time used task analysis and scenario analysis to design and evaluate user interfaces. These developments reflect the commonly-held view that scenarios and use cases are powerful and simple-to-use techniques for systems requirements specification, design and testing.

However, there remains a wide diversity in the use and effectiveness of scenarios and use cases, and sometimes even in agreement over the definition of use cases and scenarios themselves. Despite their widespread use there are few methods or software tools available to achieve systematic scenario use, or even to generate useful useful cases scenarios in the first place. A recent analysis of scenario use reveals that current methods and software tools provide insufficient and unsystematic guidance for acquiring and validating system requirements. This has led to considerable debate but few concrete answers.

This symposium aims to provide some concrete answers. It brings together for the first time practitioners, vendors and academics with interests in scenarios and use cases in the systems development process. Invited experts from around the world will explore the diverse uses of scenarios and use cases, examine effective methods and guidelines and propose a common path for future development and application of scenario-based systems development methods.

Who should attend? Practitioners, vendors and academics with interests in scenarios and use cases in the systems development process. However it shall be targeted more at practitioners and vendors.

Attendance: the event shall be limited to 100 attendees. For more information contact Dr Neil Maiden; tel: +44-171-477-8412; e-mail: N.A.M.Maiden@city.ac.uk.

Managing requirements change: a business process re-engineering perspective

Date : June 11th 1998
Format: Full day colloquium chaired by Bashar Nuseibeh and Peter Henderson
Location: IEE, Savoy Place, London.
Details to be announced

RE-2Day

Date : September 29th/30th 1998
Format: Two-day industrial conference

RE-Day is back! A two day extravaganza!!
Look out for more details in the next RQ

Distributed Requirements Engineering

November 98
Details to be announced

Please note: *RESG's policy is that distribution of speakers' slides will be free of charge to those who attend the meeting and specifically request a copy. Those people who do not attend the meeting and request a copy of the slides will be asked to pay £5 to cover photocopying and mailing costs.*

RE-News

SEWORLD Mailing list

A new, non-commercial mailing list for the Software Engineering community is now available at: SEWORLD@cs.colorado.edu.

SEWORLD is intended to serve as a central place for relevant announcements of software engineering conferences, workshops, symposia, special journal issues, calls for papers, research and educational systems, and the like. The list is moderated to avoid spam, duplication, and other misuses. In addition, all e-mail addresses are registered privately to the list, and are not published nor will they be given out to anybody requesting them.

Requests to be added to SEWORLD should be sent to majordomo@cs.colorado.edu

To subscribe, send (in the body of the message)
subscribe seworld <desired e-mail address>
To unsubscribe, send (in the body of the message)
unsubscribe seworld <registered e-mail address>
To find out more options, send (in the body of the message)
help
Please contact "owner-seworld@cs.colorado.edu" if you encounter any problems or have any questions.

Visit the SEWORLD web site at:
http://www.cs.colorado.edu/serl/seworld

CaberNet Electronic Newsletter

CaberNet is the ESPRIT Network of Excellence (NoE 21035) in distributed computing systems architectures. The mission of CaberNet is to coordinate top-ranking European research in distributed and dependable systems, to make that research accessible to governments and industries and to further the quality of education concerning such systems. CaberNet addresses all aspects of the design of networked computer systems. These systems can range from embedded systems used to control an aircraft in flight to globe-spanning applications searching for information on the World-Wide Web (WWW).

CaberNews is the electronic newsletter of CaberNet providing up-to-date information on CaberNet activities and related news. CaberNews is freely distributed by email and on the WWW. Contact Nick Cook nick.cook@newcastle.ac.uk to be added to the CaberNews email distribution list or provide input to the newsletter.

Issue 1 of CaberNews can be found at:
http://www.research.ec.org/cabernet/news/issue1.html

COCOMO II Now available

A public version of the software cost estimation tool COCOMO II (COnstructive COst MOdel) is now available at the USC Center for Software Engineering's web site:
http://suset.usc.edu/COCOMOII/Cocmo.html

It includes a Java program and associated manuals.

The 1997 version of COCOMO II has been calibrated to 83 project data points contributed by the COCOMO II Affiliates, primarily large commercial and aerospace firms. Additional data points continue to be contributed; the COCOMO II project plans to issue annual upgrades as the model is calibrated to larger samples. COSTAR, a commercial version of COCOMO II, is also available from Dan Ligett (ligett@SoftstarSystems.com).

Keeping in Touch with Other RESG-ers

Thanks to Andy Wilkes of the BCS, the RESG now has its own email alias. This means that RESG members (well, actually anyone) can send mail to resg@bcs.org.uk and have it distributed to all individuals who are on the RESG electronic email list. This is a trial service in order to see what you RESG-ers make of it.

There are of course other Requirements Engineering email lists already out there and we're not just adding another one for its own sake, rather we're providing a service for RESG-ers to talk to each other - so if you want to use it, please do! For the moment, mail sent to resg@bcs.org.uk will be forwarded to the group's publicity officer, Andy Vickers, who will then forward it to the list.

Formal Methods: A Practitioner's Companion

Quality control via formal methods is a powerful and effective means to assure system safety, and a new report by NASA/JPL writers is worth noting. The report is the result of a group effort headed by Dr. John C. Kelly of JPL (John.C.Kelly@jpl.nasa.gov) reports on experimental use of formal methods of verification on a real-life NASA example. Email to Dr. Kelly may get one of the remaining printed copies, or point your browser to:
http://eis.jpl.nasa.gov/quality/Formal_Methods/

ICSE workshop on SE education

An ICSE workshop on Software Engineering Education will be held on April 25 1998, in Kyoto. (i.e. immediately following the main ICSE conference).

Participation is via a position paper, which should be around 500 words in length and address some aspect of graduate level software engineering education, in other words education offered to individuals having a bachelors degree

or equivalent and hopefully some relevant working experience in a software/system development environment. Of particular relevance/interest are distance learning programmes, studio type programmes such as that offered at CMU and part time programmes for those in full-time employment. Lifetime learning is a term we hear increasingly. Should we redesign our programmes on the assumption that lifetime learning becomes the norm for professional people?

Position Papers should be sent to me by email (as an attached Word document) ASAP and not later than March 28. Acceptance will be notified ASAP after receipt of position paper together with registration details (workshop fee will be 100 US Dollars for IEEE/ACM members)

REFSQ'97 Proceedings

REFSQ'97 (Requirements Engineering: Foundations of Software Quality) was held in conjunction with CAiSE'97 in Barcelona (Catalonia, Spain) - June 1997. The proceedings of the workshop are published by the University of Namur Press and can be ordered by sending an email to Eric Dubois (edu@info.fundp.ac.be). They cost 25 Ecus / US$30 / 1000 BF (Belgium Francs)

The table of content is the following:

Calendar

February 1998

Second International Workshop on Development and Evolution of Software Architectures for Product Families, Las Palmas de Gran Canaria, Spain, February 26 & 27, 1998. *http://hpv17.infosys.tuwien.ac.at/ARES/*

March 1998

Second Workshop on Formal Methods in Software Practice (FMSP98), Clearwater Beach, Florida, USA, 4-5 March 1998. *http://www.bell-labs.com/user/maa/fmsp98*

European Joint Conferences on Theory and Practice of Software, Lisbon, Portugal, March 30 - April 3, 1998. *http://www.di.fc.ul.pt/~llf/etaps98/*

International Conference and Workshop on Engineering Of Computer Based Systems, Jerusalem, Israel, March 30 - April 3, 1998
http://www.ece.arizona.edu/~ecbs/ECBS-98.html

Empirical Assessment & Evaluation in Software Engineering (EASE'98), Keele University, Staffordshire, UK, 30th March - 1st April 1998.
http://www.keele.ac.uk/depts/cs/Announcements/conferences/ease98.html

Requirements Methodology: a 3 day course by QSS Ltd, Oxford Science Park, Oxford, March 17 – 19, 1998.
http://www.qssinc.com/courses/timetables.html

April 1998

Second European Conference on Cognitive Modelling (ECCM-98), Nottingham, UK, 1 - 4 April 1998.
http://phoenix.herts.ac.uk/~rmy/eccm98/programme.html

Third IEEE International Conference on Requirements Engineering (ICRE'98), Colorado Springs, Colorado, USA, April 5-10, 1998. *http://www.cs.technion.ac.il/~icre98/*The 1st UK Colloquium On Object Technology And System Re-Engineering, Oxford University, 6-8 April, 1998.
http://www.cms.dmu.ac.uk/STRL/cotsr/cotsr.html

Ninth IEEE International Workshop on Software Specification and Design (IWSSD9), Ise-shima, Japan, April 16-18, 1998.
http://salab-www.cs.titech.ac.jp/iwssd9.html

The Eleventh Workshop on Knowledge Acquisition, Modeling, and Management (KAW'98), Banff, Alberta, Canada, April 18-23, 1998.
http://ksi.cpsc.ucalgary.ca/KAW/KAW.html

The 20th International Conference on Software Engineering (ICSE'98), Kyoto, Japan, April 19-25, 1998.
http://icse98.aist-nara.ac.jp/

ICSE98 International Workshop on Component-Based Software Engineering, Kyoto, Japan, April 25-26, 1998.
http://www.sei.cmu.edu/technology/dynamic_systems/cbs/icsewkshp.html

International Workshop on Human Dimensions in Successful Software Development, Kyoto, Japan, April 19-25, 1998.
http://btwebsh.macarthur.uws.edu.au/san/hudworkshop

ICSE98 Workshop on Precise Semantics for Software Modeling Techniques (PSMT), Kyoto, Japan, April 20th.
http://www.forsoft.de/~rumpe/icse98-ws/

The 1st IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC '98), in conjunction with The 4th IEEE Workshop on Object-oriented Real-time Dependable Systems (WORDS '98) and The 4th International Workshop on Object-Oriented Real-Time Systems (WOORTS '98), Kyoto, Japan, April 20 - 22, 1998. *http://dream.eng.uci.edu/isorc/*

May 1998

Third International Conference on the Design of Cooperative Systems (COOP'98), Cannes, France, 26-29th May, 1998
http://zenon.inria.fr/acacia/Coop/Coop98/

June 1998

"Modelling and Design". 5th International Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'98). Abingdon, UK, June 5-8, 1998.
http://www.dcs.qmw.ac.uk/research/hci/dsvis98

International Conference On Formal Ontology In Information Systems (FOIS'98), In conjunction with the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 6-8, 1998. *http://mnemosyne.itc.it:1024/fois98/*

Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'98) Pisa, Italy, June 8-9 1998 (Preceding the CAiSE*98 conference)
http://www.ifi.uib.no/konf/refsq98/cfp98.html

Conference on Advanced Information Systems Engineering (CAiSE*98), Pisa, Italy, 8-12 June 1998
http://www.pianosa.cnuce.cnr.it/caise98

IEE Colloquium on "Managing requirements change: a business process re-engineering perspective", IEE, Savoy Place, London, 11th June 1998.

5th International Conference on Software Process (ICSP5), Chicago, Illinois, USA, 15-17 June 1998
http://www.bell-labs.com/user/dep/prof/ispa/icsp5/

IEEE Seventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98), Stanford University, CA, June 17-19, 1998.
http://www.cerc.wvu.edu/WETICE/

Tenth International Conference on Software Engineering and Knowledge Engineering (SEKE '98), June 18-20, 1998 at the Hotel Sofitel, San Francisco Bay, USA.
http://www.ksi.edu/seke/seke98.html

Software Process Simulation Modeling Workshop '98, Silver Falls, Oregon, June 22-23, 1998
http://www.cs.pdx.edu/conferences/prosim98/

The Third International workshop on the Language Action Perspective on Communication Modelling (LAP98), Stockholm June 25-26, 1998
http://www.ida.liu.se/labs/vits/lap98/

August 1998

Sixth International Conference On Conceptual Structures (ICCS'98), Montpellier, France, August 10-14, 1998
http://www.lirmm.fr/ICCS98

September 1998

HCI'98, Sheffield Hallam University, 1-4 September 1998
http://www.shu.ac.uk/hci98

The 5th International Conference on Object-Oriented Information Systems (OOIS'98), Paris, France 9-11 September 1998.
http://panoramix.univ-paris1.fr/CRINFO/OOIS98

IFIP 13.2 Working Conference on Designing Effective and Usable Multimedia Systems, Stuttgart, Germany, September 9-11, 1998.
http://www.swt.iao.fhg.de/deums98

3rd Annual International Conference on Software Process Improvement - Research, Education and Training (INSPIRE '98), University of Sunderland, UK, 10th-11th Sept. 1998.
http://osiris.sunderland.ac.uk/INSPIRE98/

Third Northern Formal Methods Workshop, Ilkley, UK, September 14-15 1998
http://www.comp.brad.ac.uk/research/nfmw/

IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'98), Heraklion, Crete, Greece, September 14 - 18, 1998.
http://iihm.imag.fr/EHCI98

5th International School and Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98) Lyngby, Denmark, Sept 14-15 (School) and Sept 16-18 (Symposium) 1998.
http://www.it.dtu.dk/~ftrtft98

Sixth European Workshop on Software Process Technology (EWSPT-6), near London, UK, September 16-18, 1998.
http://www-dse.doc.ic.ac.uk/~ban/misc/ewspt98.html

October 1998

The Third International Conference on Practical Software Quality Techniques (PSQT'98), St. Paul, Minneapolis October 5-7, 1998.
http://tcqaa.org/psqt/index.html

International Workshop On Current Trends In Applied Formal Methods, Boppard, Germany, 7-9 October, 1998.
http://www.dfki.de/vse/fm-trends/

13th IEEE International Conference on Automated Software Engineering (ASE'98), October 13-16, 1998, Honolulu, Hawaii, USA
http://www.ics.uci.edu/~ase98

Workshop on Industrial-strength Formal specification Techniques (WIFT'98), Boca Raton, Florida USA, October 21-24, 1998.
Info: chengb@cps.msu.edu

November 1998

METRICS 98: Fifth International Symposium on Software Metrics, Bethesda, Maryland, November 19-21, 1998.
http://aaron.cs.umd.edu/metrics98

September 1999

FM'99: Formal Methods 1999, The World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, Late September 1999.
http://www.it.dtu.dk/~db/fm99/

RE-Readings

Reviews of recent Requirements Engineering events.

RE-Day Keynote Address

Report by **Ian Alexander**

Suzanne Robertson of the **Atlantic Systems Guild** gave the keynote address, entitled '**Requirements: Rough Sketch or Precise Specification?'** With 20 years' experience as what is now called a requirements engineer, she was an inspirational speaker.

She began by showing a slide of a subtly lit painting by James Tissot, depicting a ball at Cowes week. The painter started from a blank canvas, and continued until he felt he had 'finished' according to his own evaluation criteria. He

had proceeded from a rough pencil or charcoal sketch, by negotiation with the structure he was creating, increasing precision bit by bit, locally, until he had achieved what he wanted.

She works in the same way, within certain constraints such as purpose, goals, time, and money. The subject is sketched out with functional and non-functional requirements, using words, pictures, first-cut models, and prototypes. The sketches build up an increasingly precise picture, as well as posing questions. Sketches can be of many kinds, such as video tape, or any of the representation techniques so well described in ACRE.

The sorts of questions that may be put are, it seems, rather similar to those that the essayist must famously put when trying to write about a bear (What? Why? Which? When? Who? How?). Robertson suggests asking:

- why is this important?
- what are the domains of interest?
- who cares about this?
- is this actually several requirements?
- how will we know if a solution meets the requirements? (She calls this the 'fit criterion', following the guru architect Christopher Alexander.)
- what assumptions should we make?

Listeners were advised to look out for mismatches in internal meanings held by different people (such as groups of users); to identify precisely the boundary of the project, and to define the interfaces. All of these are frequent causes of acrimonious failure: ill-defined projects cannot end happily.

Sketches have to be captured in a rough state. That does not mean a vague state though: Robertson is emphatic that even crude sketches contain identifiable and measurable statements, using a 'shell' or template to describe who is involved, what the product is to be used for, how it can be most effective, and what the product is to be.

The idea is that the requirements engineer keeps a sort of reporter's notebook; each page documents a requirement with a number, a use case number, a one-sentence description, a purpose, a source (who), a fit criterion (test), dependency, supporting materials (references), history (date), and a couple of feelings on a scale from 1 to 5: customer satisfaction if the requirement is met, and dissatisfaction if it isn't.

Once written down, each requirement has to pass a quality check, which Robertson calls a 'gateway'. There must be a fit criterion, ID, and cross-reference. It must be relevant in its context, viable given the project's constraints, of value to the 'customer' (or did she mean the users?), state a requirement not a solution, not be a gold-plated 'Japanese toothpick', be complete, and use defined terminology. There also has to be a consistency check between requirements. Any cases of doubt are recorded as risks to the project.

She then presented a detailed template. At the top level, the template organises the requirements by identifying global 'constraints' (what one might call the introduction or context) such as system purpose, customer, users, naming and terms, facts, constraints (in some narrower sense), and assumptions. There follow functional requirements, non-functional requirements (also known as constraints!), and project issues.

This emphasis on issues, on precisely what is not yet sorted out in the project, is perhaps Robertson's key contribution. She suggests keeping lists of open issues, COTS components, new problems, tasks, 'cutover', user documents, and a 'waiting room' especially for rough sketches of ideas which are still outside the project boundaries.

Her credo is 'Formality enables Precision', and she bravely showed an amazingly complex entity-relationship diagram of the entire requirements engineering process, to gasps and laughter. There is, in effect, a creative tension between temporary acceptance of rough, incomplete and possibly conflicting sketches, and the relentless pursuit of completeness, consistency, and precision.

Robertson ended her address by stressing that each project's context must be wide enough to include the real-world-out-there 'system' into which the little bit we are building will fit, to ensure success.

Questions from the floor followed.

Geoff Mullery, one of the controversial soapbox speakers, offered sympathy, but said it wouldn't work for political reasons.

Robertson agreed that yes, the core was people.

Anthony Finkelstein said that specification was often over-precise. The real work was to strip out needless precision.

Linda Macaulay asked what the role of the requirements engineer was: facilitator or gamekeeper?

Robertson suggested the role of a helper who identified stakeholders.

If this is all about people, someone asked, how is this engineering? Robertson said she didn't know.

Alistair Sutcliffe asked whether the template would not need to be different for projects that had to handle legacy systems.

Robertson agreed, and said that the template was available on the web and was updated regularly.

John Dobson, another combative soapbox speaker, suggested that what we did was not like engineering as the requirements engineer could not own the requirements, unlike a civil engineer who owns the correctness of, for instance, the design of a bridge.

Godfrey Draper asked if one could fix (freeze) requirements.

Robertson said no, change just happened. It was a Darwinian world in which things evolved all the time.

Bernie Cohen asked what a specification really was.

Robertson replied that a specification was a living organism.

Laurence Brooks asked whether the template was a set of guidelines, or a means of enforcing consistency.

Robertson asked if he was asking about reuse of experience, and explained that the template was generic but tailored each time.

Bashar Nuseibeh (chair) referred interested listeners to a recent paper by Steve Easterbrook on "Living with Inconsistency" which appeared in the 8th IEEE International Workshop on Software Specification and Design (IWSSD-8)

Linda Macaulay asked whether the problem was not now a thing of the past. Nowadays people just issue a rough and ready product and iterate.

Robertson answered that it was all down to risk analysis. Sometimes a rough product was fine, but many projects failed.

David Anderson asked how much detail was needed.

Robertson suggested far enough to find the fit between COTS and the users' needs, and referred to the work of the CREWS project (at City University).

Bashar Nuseibeh said that Anthony Finkelstein had worked on package selection.

Fraser Michaelson asked whether there were not conflicts between stakeholders, or whether there was vagueness?

Robertson replied that she tried to show what each responsibility was, and to make what each stakeholder said come alive so that all the stakeholders came into the loop. This could be encouraged by use of modelling techniques.

Alan Cooper asked whether a repository for business requirements could serve as a knowledge base for a whole organisation.

Robertson thought this a wonderful concept, but it would be a large effort. For example, an architecture project in Houston had captured requirements on cards.

Readers can find out more by looking on the Web at *http://www.atlsysguild.com.*

RE-Day: The Tutorials

Report by **George Spanoudakis**

Tutorial 1: UML for Requirements Engineering

The first tutorial was on the Unified Modelling Language and how it can be used for requirements modelling. The tutorial was given by Dr. Jim Arlow a trainer and consultant who is involved in object-oriented methods since 1990 and has won an award for his work on object-oriented development environments and transfer of object technology to large organisations at the Object World exhibition in Frankfurt.

The tutorial focused on those aspects of UML that may be effectively used in requirements engineering. In particular, it discussed the artefacts generated in the analysis phase of an object-oriented system lifecycle, including use-cases, scenarios, class diagrams, sequence diagrams and collaboration diagrams. Of particular interest to the audience was the discussion on how high level system requirements can be expressed by use cases and how use-cases can be mapped down to individual system classes. Further discussions focused on the semantics of various structural relations between use-cases and the relationship between use-cases and scenarios.

Tutorial 2: Requirements engineering for systems

The second tutorial discussed requirements engineering for very large systems and focused on the implications that monitoring products and maintaining their benefits throughout long product development lifecycles has for the requirements engineering process. The tutorial was given by Dr Thomas Docker. Dr. Docker has been actively involved in practical and research work on project management and product delivery approaches concentrating on the realisation and preservation of business benefits.

The tutorial addressed issues involved in managing requirements in relatively long development life cycles. In such settings both the profile of the claimed product benefits and the requirements themselves are likely to change. As a result requirements need to be identified, structured and mapped to adequate solutions, in a way that ensures continuous monitoring of the products and their benefits. A scheme for expressing and monitoring requirements that supports this monitoring was presented. This scheme structures requirements around problems and solutions and involves a number of different types of constraints.

Subsequent discussions focused on the use of the scheme in practical settings. More specifically, there were questions about the structural expression of problems and solutions and the implications of using the different types of constraints. Thomas Docker indicated that constraints may be used to restrict the space of the feasible solutions to a problem gradually and that the use of different types of constraints depends on the willingness of the stakeholders to compromise on them if necessary.

Tutorial 3: Rapid and Requirements Engineering

The third tutorial examined the principles and practice of rapid and evolutionary systems development methodologies. The tutorial was given by John Crinnion, a senior lecturer at the City University in London, who has worked extensively as a consultant on RAD projects of all sizes and is the author of the book 'Evolutionary Systems Development' (published

by Pitman in 1991). The tutorial focused on how rapid and evolutionary systems development methodologies address the requirements engineering aspects of business applications and what are the implications of these methodologies for the theory and practices underpinning requirements engineering. The tutorial was supported by videos illustrating the use of RAD methodologies in practice. The discussions that followed also focused on problems encountered in the practical application of RAD methodologies.

RE-Day - The Requirements Tools

Interviews by **Ian Alexander**

With a large room full of the latest requirements tools, RE-Day participants would have had more than enough to do just to look at all the tools on offer. For those who were not there, or who spent the whole day listening to the talks in the other rooms, here are some quick highlights and unique selling points of some of the different tools that were on display. The tools are in alphabetical order.

Cradle

IA: What are the unique features of Cradle?

Malcolm Boyack of Structures Software Systems Limited (3SL): Cradle is a full life-cycle management tool, not a single-phase product. It can integrate any other applications. It is both multi-user and multi-site. It has built-in Configuration Management. It supports risk analysis via a user-extensible database and checking.

IA: How does Cradle do that?

MB: At the centre is Cradle-PDM, which links up all the other tools. There is a document editor, a charting tool, a performance modeller, a system modelling tool, and a source document manager. The idea is that we manage systems all through their life-cycle.

IA: From cradle to grave?

MB: That's where the name comes from. Once you have requirements in the database you can do analysis and design with dataflow diagrams, OMT, state diagrams, use cases, object interactions and so on.

IA: How do you trace requirements?

MB: The Cradle database lets you cross-reference any information to anything else. It also creates its own cross-references, such as from source documents to requirements.

DOORS

IA: What are DOORS' unique selling points?

Amanda Haisman-Baker of Quality Systems & Software (QSS): DOORS has a very strong method basis. It is very easy to use. There is a very short learning curve. It offers unique navigation of requirements via its fisheye hierarchy view.

IA: How does it achieve that?

AHB: DOORS uses an object-oriented technique to organise any number of inter-linked requirements. You can add as many user-defined attributes as you like to each requirement. You can switch instantly between text, outline, and graphical views to see how your requirements relate to each other.

IA: What about other information like pictures and graphs?

AHB: You can insert pictures directly, or you can use OLE to link in image editors or Excel charts or whatever.

IA: How do you handle traceability?

AHB: There is a choice of matrix or back-to-back tree views. As well, you can put in traceability columns to see links to or from other documents.

IA: What happens if requirements change?

AHB: DOORS keeps a full history. You can baseline modules any time. The traceability allows you to see directly the impact of any changes.

Orion

IA: What are the unique features of Orion?

Mike Dawe of Machine Reasoning Limited: We do modelling with intelligent objects. We elicit requirements by building a model without a full or clear understanding at first, allowing for conflicting statements. We search for gaps, such as where there are no methods to calculate a result. Orion is new to the RE market, it comes from the world of modelling and machine intelligence.

IA: How does it work?

MD: Its high level language is close to users' knowledge, such as the languages of logic, sets, and algebra. Expressions can be inverted automatically, so if anything is missing it can be found from the rest.

IA: How do you get requirements from users?

MD: Orion is not a database or hierarchy of sets of textual statements. It is at the formal end, but it allows for a range of users with different views or parts of a model. It is fast enough to use interactively to capture ideas, good for inconsistencies, gaps in knowledge. You could import text and use it as hypertext with traceability links to the model.

PC Pack

IA: What are the unique selling points of PC Pack?

Patricia Hughes of Integral Solutions Limited (ISL): Well, it's cheaper! At £1200 per seat, less for academics. PC Pack is a collection of integrated tools, some of them third-party. It is method-centred, using the GDM (Generalised Directive Model) based on the older KADS method. Its home was the knowledge acquisition techniques from the artificial intelligence research of people like Nigel Shadbolt.

IA: What is the software then?

PH: It uses its own editor and a custom-made database. There will be ODBMS and so on. There is a card sorting tool for acquisition and refinement. Our policy is to develop and refine requirements in any tool to suit the specific style and circumstances. As you can see, most of the tools are very visual.

RDD-100

IA: What are the special features of RDD-100?

Brian Hardwick of Ascent Logic: The prime differentiator is RDD's tight integration between a set of requirements and the behavioural model. The system architecture is represented in our own notation but this can be modified, e.g. to DFDs (Dataflow diagrams). The model is executable in the style of a simulation.

IA: How can you look at these models and requirements?

BH: All views are editable. You can filter by components or relationships: you need not see Objects unless you want to. Requirements fit into the framework of system engineering. Text alone is not sufficient for requirements. RDD provides an outliner that indents the requirements according to their depth in the hierarchy.

IA: How can you modify the structure, such as to add an attribute?

BH: You have to save the data and empty the database, exit and view the class hierarchy. Then you load the attribute editor, update the database schema, reload the database and import the data. You can add an enumeration attribute without programming.

Requisite Pro

IA: What are the key selling points of Requisite Pro?

Alastair Aitken of Rational Software Corporation: It provides the capability to track requirements through the project life-cycle. You can generate use cases and test cases via Rational Rose and the SQA Suite (a Windows GUI testing tool). Export to Rose at low level allows you to create UML (Unified Modeling Language) and ultimately code skeletons. The real benefit is that it gives you the overall picture, to let you actually see coverage of the initial requirements model.

IA: How do you do all that?

AA: The tool is based on Microsoft Word for text editing, and Access for the data. Requisite Pro controls Word and Access, letting you select text in your requirements document for immediate inclusion in the structure. Images and other OLE items are pointed to, outside the structure.

IA: How do you tell if someone has edited an image...?

AA: Requisite Pro checks any changes to things it points to. There are many (hidden text) attributes. A 'Project' keeps track of Word files and so on. You could edit those files just with Word but you'd lose the extra information. Instead, you double-click on a requirement and it opens the Word document. The tool keeps a history of changes.

IA: How do you trace from requirements to models?

AA: With a tree or matrix. The tool checks for circularity at any depth. If there has been any change it marks the links as 'Suspect', a useful feature.

SOMATiK

IA: What are the unique selling points of SOMATiK?

Mark Lewis of Bezant Object Technology: It's the strongest requirements capture tool. No object-oriented (OO) analysis at the start! On Day 1 you capture the Mission Statement, from scratch. The tool is for requirements capture, analysis, design, and code generation, for OO in a RAD (Rapid Applications Development) environment.

IA: How do you do all that?

ML: It is like structured word-processing. There is a tabbed user interface ideal for a laptop. You can store the data in a repository and share it over a LAN. Reuse is possible at specification level. There are three models: a context (top-level to identify externals, actors, and messages); a task object model (user-facing); and a class object model (code-facing). You start by not worrying about boundaries, whether the system does something or not. Our unique selling point is that SOMATiK is 'anti-CASE' – it does all the drawing for you!

IA: How do you trace back to requirements?

ML: A rule set captures the semantics of business rules. You can check that the class model is necessary, sufficient, and self-consistent. Metrics such as our task-point metric of complexity are produced automatically.

TMA Toolset

IA: What are the special features of TMA?

Paul Mellon of Systems Engineering & Assessment (SEA): TMA has a Word front end in a Microsoft Office environment. It is specially tailored for the tender assessment process in the procurement cycle. We have extensive experience of Ministry of Defence requirements, tender assessment and acceptance procedures.

IA: How does the tool work?

PM: There are actually three tools, Tracer, Marker, and Acceptor. We have Tracer on show here today. Tracer is for requirements capture. You edit the source in Word, use Access for analysis, and then issue your documents.

IA: You use database reports to time-stamp them?

PM: Yes, that sort of thing. We link requirements with Word Bookmarks.

IA: What will make people buy the tool?

PM: Ease of use, familiar interfaces. It is a simple tool, you can learn it in one day. It is well tailored to tender assessment and acceptance, especially in a defense environment.

A Visitor's View

IA: How are you finding the exhibition and events today?

Claire Clayton: Well the timetable is a bit difficult! I enjoyed the tutorial on training. I used to be in process improvement and came along today for help on requirements engineering and its relation to process modelling. It's interesting but I'd need more time to see everything.

Can the Requirements Management Process be Quality Certified?

The final RESG event of 1997 was a debate on Wednesday 26 November 1997, at Imperial College, London.

Report by **Ian Alexander**

This last meeting of '97 took the form of a debate, with lively involvement from the floor, between the three speakers: Adrian Gill, Pete Sawyer, and Richard Stevens.

Adrian Gill, now a software process improvement consultant with Xerox International Software Operations, spoke first, stressing the Software Engineering Institute's Capability Maturity Model (CMM). The basic question was why engineer requirements? The answers were professionalism; quality (Crosby's conformance to requirements); and profit. A business objective, whether to make a product or, more likely, to make a profit, always meant following an opportunity. One needed to measure that one was meeting requirements and the ultimate business objective, so as to ensure that the problem was solved and to be able to repeat successes (repeatability being a stage in the CMM). This called for a formal process such as that described in the report SEI-91-TR-30 on the SEI's website. Requirements management was the starting point in the CMM for process improvement, so perhaps RE could be certified as part of the overall systems process; or perhaps one should try to certify the product of RE, namely the resulting analysis.

In any case, said Gill, it was questionable whether RE could be certified in isolation from, say, the ISO 9001 or TickIT standards for system engineering. There would have to be an external frame of reference, widely accepted. So, he argued, there was no benefit to be obtained from RE certification as such: one should certify in the large.

Pete Sawyer, Lancaster University, (REAIMS Esprit project) spoke next. He stated that the industrial partners on his project were mature and used both ISO 9000 and CMM with tight regulation, but still experienced many problems of cost overrun and quality. There were several domains to consider, such as customer- versus market-driven; software-intensive versus software-only; RAD versus fully-planned development. He pointed out that developers can already choose from a range of QA standards (ISO, CMM, SPICE, …); lifecycle standards (PSS-05, ISO-12207, …); documentation (IEEE 830, …); and domain-specific standards, such as the military DefStan 00-55. All of these touched on RE in some form: for instance CMM Level 2 refers to the need to allocate resources to RE. The question was, which good practices should be common?

One might try to distil the best from all these standards, deriving an ideal documentation standard describing properties, language, and structure; or a requirements standard for CMM, covering tracing, configuration management, verification plans, and the review process. There were various possible intentions behind such approaches, such as to identify conflicting requirements, or to gather as many viewpoints as possible.

REAIMS had, between 1994 and 1996, produced a set of guidelines and a process improvement framework called a Process Maturity Model (not the same as CMM!) with the steps:

1. Initial;
2. Repeatable;
3. Defined;
4. Measured.

"Nobody has yet reached level 4" said Sawyer.

He concluded that there were already enough standards to raise the average practice. It was possible to certify at least to level 2, repeatability. New standards could be designed explicitly to do this, even if one did not go so far as to award certificates.

Richard Stevens, Quality Systems & Software, gave a characteristically amusing speech, starting "It may well work in practice but the theory is quite unsatisfactory". He referred to the Standish corporation's report, showing that 5/8 of system problems can be put down to requirements; the remaining 3/8 to management. QSS was now the largest RE company in the world, having sold upwards of 6000 licences, with 100 staff. But despite this, he at once said "Tools are a waste of time. You have to get the process right". The basic job was always to get the right product to market on time; then one had to inject new versions as competitors came in.

A certificate, he said, was a neutral test against a standard, such as ISO-15288. As the slogan over the doorway of the testing museum in Southwark, London has it, "Facts, not Opinions". One could conform to the ESA's PSS-05 software engineering standard, or the generic high-level approach of ISO-9000's eight pages, or the arbitrary set of questions posed by the Capability Maturity Model.

One could apply the concepts of RE to the development of a certification process: write a User Requirement for it, define a structure for the standard, and build an Architectural Design for the process itself. The basic requirements were few and simple: it must be developed systematically; it must be neutral with respect to any good RE process; it must meet the business case; and it must add value, based on real experience of large practical projects. "I happen to have a requirements tool handy", said Stevens, and he presented a list of requirements and constraints. The question was, should we do it? Perhaps we were still a bit weak on repeatability, let alone on full definition of a prescribed and measurable method. Every project still needed its creative and inventive "Italians" along with its disciplined and rigorous "Germans" (he apologised for the stereotypes).

The debate moved to the floor, where the topics covered included: the need and possibility of requirements metrics; whether systems included software or vice versa; whether one could standardize or certify; and whether certification

applied to the process or to the requirement engineers (laughter).

The feeling was that RE could probably be certified, but that perhaps it was not quite ready for this. Until then, there were quite a few attributes that could be measured, and we could follow a few basic rules - such as ISO 9000's - which would make a big difference in quality at low cost. Whether we called the goal "Customer Satisfaction" and "Quality", "Service Development", or "Mature Capability", there was a clear need to continue to improve, and the way ahead was reasonably clear.

RE-Papers

RPI: Requirements Process Improvement the REAIMS Way

By Pete Sawyer, Lancaster University.

Many organisations have made a commitment to software process improvement (SPI) based upon, for example, the SEI's Capability Maturity Model (CMM) for Software [1]. Many more organisations have taken steps to comply with the ISO9001-3 quality standard which shares many of the same aims as SPI. It must be admitted that there is a fair degree of cynicism about the motivation of some organisations seeking accreditation under the various SPI or quality standards. However, there is mounting evidence that, given sufficient organisational commitment, an SPI programme can yield real benefits [2] for delivery scheduling, cost control and product quality.

SPI emerged during the late 1980s and its influence on the industry's practice, particularly in North America, has been growing ever since. Requirements engineering (RE) has also received much industry attention and research funding during this period. However, the impact of this on industrial practice compares poorly with that of SPI. This prompts the question: why is there a difference? Admittedly, customers are able to wield the refusal to award contracts as a stick to drive SPI. Nevertheless, we don't think that this completely explains the disparity since few organisations are complacent about RE.

The question interested the REAIMS project (Requirements Engineering Adaptation and IMprovement for Safety and dependability). Our conclusion was that we, as RE researchers, consistently underestimate the effort needed to adopt a new technique and integrate it into an existing process. We seldom recognise the risk involved in selecting and applying even a mature RE technique on real projects. Inevitably, as schedules are squeezed, even long-recognised, measures such as requirements tracing become vulnerable to neglect [3]. The result is that, for too many organisations, RE continues to be the most risky and intractable aspect of system development. No software process, whatever its "capability", can keep delivery times, costs and product quality under control if the requirements are poorly formulated or unstable.

Clearly, these were problems that would also face the ultimate acceptance of the novel techniques and tools developed in REAIMS. Our solution was to address the basic process problems impeding organisations' ability to exploit otherwise useful techniques. If the principles of SPI could be applied to RE processes, they would offer a practical means to substantially improve the state-of-the-practice (This insight was not unique to us - the forthcoming CMM for Systems Engineering will also address RE).

An obvious question is: why can't existing standards be used to the same effect? Existing SPI "standards" don't solve the problem since they provide relatively little explicit guidance on RE. This is because SPI has had to strike a balance between: addressing crucial process areas; restricting support to those areas where best practice has been established; and addressing their brief to focus on software development processes.

RE poses a number of difficult problems in these terms. It isn't a discrete activity but rather is a complex set of activities that feed into software development and, in many cases, are enacted concurrently with software development. Consensus about what constitutes best practice in RE is relatively hard to arrive at (one reason why we prefer to talk in terms of good practice). Additionally, in many application domains, rather than being a software engineering activity, RE is a top-level phase of the overall systems engineering process (and there is a strong argument that this should be true for all domains, even software-only applications).

The consequence of these problems is that, while RE's importance is recognised, it isn't treated in much detail. The CMM, for example, mandates that the software requirements are allocated, analysed, documented, and managed but doesn't specify the activities which result in allocated, analysed, documented and manageable requirements.

However, this doesn't mean that guidance on RE good practice is unavailable. A number of software and systems engineering standards cover RE processes. For example, the European Space Agency's ESA PSS-05 [4] lays out a number of fundamental practices, documentation requirements and planning activities for the elicitation, analysis and management of customer requirements.

There is therefore a substantial amount of readily available wisdom and experience on RE. All that was missing was help for an organisation to tailor it to its own needs. The REAIMS view was that it was timely to exploit the industry's acceptance of SPI to present known good RE practice within a framework which allows organisations to identify the weaknesses in their existing RE processes and plan appropriate phased improvements.

The Requirements Engineering Good Practice Guide (REGPG) [5, 6] is the result. The REGPG contains 66 RE good practices, a set of recommendations for assessing an existing RE process and guidance for selecting and applying practices to address the identified weaknesses.

We were concerned to make the REGPG compatible with existing SPI models, so the REGPG has a 3-level (Initial, Repeatable and Defined) maturity model based loosely on the CMM's 5-level model. It is fairly straightforward to characterise RE processes in terms analogous to the CMM's bottom three maturity levels. However, what we know about the current state-of-the-practice makes us doubt if any RE processes exist which, in CMM terms, could be characterised beyond Defined (level 3). Certainly, if requirements processes exist which are Managed (CMM level 4) or Improving (CMM level 5), we do not know how to distinguish them.

The three REGPG maturity levels are characterised in terms that will be familiar to anyone used to the CMM. Initial RE processes are performed ad hoc. It is hard to estimate and control costs as requirements have to be reworked and customers report poor satisfaction. The processes are not supported by planning and review procedures or documentation standards. They are dependent on the skills and experience of the individuals who enact the process.

Repeatable RE processes have defined standards for requirements documents and have introduced policies and procedures for requirements management. They may use tools and methods. Their documents are more likely to be of a consistent high quality and to be produced on schedule.

Defined RE processes are based on good practices and defined methods. The need for rework is much reduced. Organisations can make objective assessments of the value of new methods and techniques. The maturity level is determined by the practices employed and the degree to which their use is standardised. We derived the REGPG's good practices from existing standards, reports of state-of-the-practice (e.g. [7]) and the experience of REAIMS partners. Within this list, we have had to recognise that while consensus exists on the genericity and utility of many practices, the value of others are more project, organisation or application domain-dependent. Similarly, while some practices are within the scope of immature organisations, some practices must be underpinned by other measures or require specialist expertise. To reflect this, we have classified practices as Basic, Intermediate or Advanced:

Basic practices represent fundamental measures that underpin a repeatable process by, for example, defining documentation and basic management standards. The adoption of basic practices should normally have the highest priority.

Intermediate practices are typically more complex but help make the process more systematic by, for example, using methods for conceptual modelling. Intermediate practices usually have to be underpinned by basic practices in order to be effective.

Advanced practices are practices which often require substantial experience and expertise or which support continuous improvement. They include practices that are of most benefit in specialist domains.

Clearly, the classification we have chosen for some of the practices will be controversial. However, we have tried to counter what we believe is the natural tendency to underestimate the difficulty in applying many seemingly straightforward practices.

To help with evaluating the good practices, they are presented as guidelines. Each guideline provides a qualitative assessment of: the key benefits of the practice; its cost of introduction (training, support tools, etc.); and its cost of application (level of resourcing needed for their support).

The list of guidelines is partitioned into the RE process areas (products or activities) to which they apply. These are:

1. The requirements document.
2. Requirements elicitation.
3. Requirements analysis and negotiation.
4. Describing requirements.
5. System modelling.
6. Requirements validation.
7. Requirements management.

REGPG maturity levels do not focus on selected process areas to the exclusion of others. We don't mandate, as the CMM does, which process areas should be addressed in order to achieve, for example, a repeatable process. Sometimes it makes sense to concentrate resources on a particularly weak process area but if there are weaknesses across the RE process, prioritising of improvements needs to be more flexible.

As an example of an REGPG process area, the practices recommended for requirements management are:

1. Uniquely identify each requirement. (Basic)
2. Define policies for requirements management. (Basic)
3. Define traceability policies. (Basic-intermediate)
4. Maintain a traceability manual. (Basic)
5. Use a database to manage requirements. (Intermediate)
6. Define change management policies. (Intermediate)
7. Identify global system requirements. (Intermediate)
8. Identify volatile requirements. (Advanced)
9. Record rejected requirements. (Advanced)

As part of an improvement plan, we would recommend that an organisation with an Initial process should consider implementing the first 4 practices. Starting from a low base, adopting basic practices from other areas will usually offer a better return than practices 5-9.

In order to evaluate an existing RE process and identify where its weaknesses lie, the requirements practices in use can be assessed against a checklist of the REGPG's good practices. Each practice in the checklist is rated as:

standardised (score 3), in normal use (score 2), used at the discretion of the project manager (score 1) or never used (score 0). The assessment should reveal particular areas of weakness where few basic practices are in standard or normal use, and the level or maturity. For example, a score of below 55 for the basic practices indicates an initial process.

Of course, this assessment scheme is not a precise instrument and an organisation should not react to its results without careful analysis of the results. They may well have good reasons for rejecting some practices and favouring others. Where a more precise measure of the process maturity is required, other schemes could be adopted (e.g. [8]). However, our scheme is deliberately quick and dirty because we are not confident that requirements metrics are sufficiently well developed to repay more systematic analysis. The merit of our scheme is that it can give a relatively fast "ball park" indication of an organisation's process maturity and, in so doing, reveal a large class of process weaknesses.

Unfortunately, some organisations have such chronic RE problems that a process assessment would reveal nothing but weaknesses and present an unhelpfully large number of potential improvement measures. For these organisations, we have selected a "top ten" good practices which we think represent the fundamental foundations of a repeatable process. These are predominantly concerned with documenting and managing the requirements and are relatively inexpensive. Unsurprisingly, they correspond closely to long-established good practice. For example, eight of the practices are very similar to recommendations that appear in ESA PSS-05.

In conclusion, the REGPG doesn't pretend to provide revolutionary solutions. However, we think that there are many RE practitioners who could use something that distils basic industry knowledge and helps them place it in context. Background work on REAIMS suggests that the RE process maturity of almost all organisations is still at the Initial level. Most organisations have pockets of good practice but their benefits are often diluted by weaknesses elsewhere.

The REGPG is intended to fill a gap in the current support available to organisations intent on resolving persistent RE-related problems. These need not be actively engaged in SPI programmes. However, the REGPG borrows from the philosophy and principles of established SPI models to add value to what is provided in existing standards. Few organisations can afford to radically change their existing RE processes. The advantage of the REAIMS approach is that it helps incremental improvement by matching the most effective measures with the most pressing problems.

Information on the REGPG can be found on:
http://www.comp.lancs.ac.uk/computing/resources/re-gpg/

Information on REAIMS can be found on:
http://www.comp.lancs.ac.uk/computing/research/cseg/projects/reaims/

References

1. Paulk, M., Curtis, W., Chrissis, M., Weber, C. 1993. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, Software Engineering Institute, USA.

2. Diaz, M., Sligo, J. 1997. How Software Process Improvement Helped Motorola, IEEE Software, 14 (5).

3. Davis, A. 1995. Tracing: A Simple Necessity Neglected, IEEE Software, 12 (5).

4. Mazza, C., Fairclough, J., Melton, B., De Pablo, D., Scheffer, A., Stevens, R. 1994. Software Engineering Standards, Prentice Hall.

5. Sommerville, I., Sawyer, P. 1997. Requirements Engineering A Good Practice Guide, John Wiley and Sons.

6. Sawyer, P., Sommerville, I., Viller, S. 1998. Requirements Process Improvement Through the Phased Introduction of Good Practice, Software Process, Improvement and Practice (to appear).

7. Lubars, M., Potts. C., Richter, C. 1993, A Review of the State of the Practice in Requirements Modelling, in Proc. IEEE International Symposium on Requirements Engineering (RE93), San Diego, Ca., USA.

8. El Eman, K., Madhavji, N. 1995. Measuring the Success of Requirements Engineering Processes, Proc. 2nd IEEE International Symposium on Requirements Engineering (RE95), York, UK.

CORE - Blimey!

A regular column by Geoff Mullery, of Systematic Methods Ltd.

What Can I Say?

In this contribution I make reference to things I said in the July and October Newsletters. If you want to find what I said and you do not have a copy of the Newsletters, email me and I can send you a copy of the article.

In the October 97 Newsletter I wrote about use of database systems and other forms of specification support tool. I mentioned seven key features needed in a support environment in order for it to satisfy all the needs of specification production. The first three, Capacity, Inter-Relationship and Distribution are primarily served by current database systems, though not entirely satisfactorily in my opinion. The fourth, Analytic Power, is supported by specialist languages (like Prolog and Lisp) but they are poor at supporting the first three features.

The last three, Autonomy, Flexibility and Immediacy are badly supported by all the current tools and tool

combinations I know of. In this contribution I try to describe desirable characteristics of a support environment which caters for these three. I emphasise that this is in the context of specification of requirements for a system whose stakeholders include various and often competing organisations, projects, resources, cultures, capabilities and priorities.

In my opinion this precludes several fondly held beliefs on the subject of requirement specification. A single "unifying" language to be used by all stakeholders is an impractical dream in all but small environments. A single non-redundant, consistent and/or complete logical database is a damaging over-constraint in any but a tight-knit single culture environment. A single authority in control of who can say, see or release what during specification is impractical and can block achievement of real communication, understanding and agreement.

I propose below a number of criteria that I believe must be satisfied by a requirement specification support environment if reliable communication, understanding and agreement are to be achieved. It is my proposition that satisfaction of these criteria is necessary, but not sufficient for success. The criteria proposed here are:

1. Anyone involved in the specification process can describe their views from multiple perspectives. In the remaining criteria I shall use the word ***Role*** to refer to a specific person specifying a given perspective. This allows several people to give their (possibly different) versions of the same perspective and any one person to give their views of several different perspectives.
2. Any Role can (as far as is technically feasible) say whatever it feels is needed in whatever way (notation) it feels able to use. The absolute minimum it must say is the name of the person who created it and the name (a characterising phrase) of the perspective it describes.
3. Any Role can retrieve anything it has said, in exactly the form it originally said it.
4. A Role can constrain the ***visibility*** of each thing it has said, on the basis of which other Roles, and locations can see it and over which intervals of time it can be seen. The default state is for all statements to be visible at all times to all other Roles (in other words, restriction of visibility requires a positive act). The absolute minimum visible information about a Role is its person/perspective pair.
5. No Role can change what another Role has said.
6. No Role can say something as if it was being said by another Role.
7. Any Role can make a (possibly named) ***relationship*** between anything it has said, whether or not it is externally visible, and any visible statement (or part thereof) of another Role.
8. Any Role whose statements have been referenced in a relationship (or via a chain of relationships) by another Role can see at least which other Role has made the reference and can see each referencing statement which is visible. Mixes of visible statements and just the existence of a relationship are possible where a chain of relationships exists.
9. The existence of a relationship (or chain of relationships) between statements by different Roles is visible to all Roles which can see any of its linked statements and to no other Roles. The statements in a relationship (or chain of relationships) which are visible to a Role can be seen in their correct position in the chain by that Role, with non-visible statements in the chain being indicated at the appropriate points in the chain as existing but not accessible.
10. Any Role can a search for visible statements constrained by the location and other Roles to search and by the time interval over which the request is to be applied. The time factor means that the result of a search may be incremented over the specified duration, if new information satisfying the other constraints becomes visible in that time.
11. Any Role can ***change or delete*** something it has said if and only if it has not been referenced in a relationship by another Role. This implies that statements kept wholly private can always be deleted.
12. Any Role wishing to change or delete something it has said must first get all references to it by other Roles deleted by those Roles.
13. Any Role can ***recant*** (assert to be incorrect) anything which it has previously said, even if it is visible to other Roles and has been referenced in a relationship by one of more of them.
14. For any statement that has been referenced in a relationship by another Role, if it is later recanted, the recanted state is visible to all referencing Roles.

Satisfying these criteria supports autonomy, flexibility and immediacy, but leaves open the possibility of anarchy. Hence, for a given proposed system or family of systems there is a need for specialised Roles to identify what forms an agreed and/or contractual position. This requires the following criteria:

15. Roles can be grouped by specifying a single ***Representative Role*** as a "spokesman" for the group. A Representative Role may itself be grouped with other Roles. The person responsible for a Representative Role makes statements and sets up relationships with statements made by other Roles in the group. The only special thing about a Representative Role is the way it is set up and interpreted. This occurs in either of two ways described below.
16. A ***Subscriber Representative Role*** is one voluntarily joined (subscribed to) by two or more other Roles, who agree on a person delegated to enter statements on behalf of them all. This allows associations (for

example political or technical or cultural groups) to declare matters on which they agree even though they do not necessarily agree on everything.

17. An ***Enlisting Representative Role*** may enlist (nominate) a set of other Roles. Its statements reference the statements of the enlisted Roles that it deems to be valid, even if the people responsible for the enlisted Roles do not agree. This permits for example several people separately to propose conflict resolutions, managers to assert decisions organisations to identify contractual commitments – each referencing the same basic information distributed among normal Roles. Even though they may not necessarily reach the same conclusions it is possible to choose unambiguously which view prevails (explained in comments below).
18. A subscriber to a Subscriber Representative Role can ***cancel*** its subscription (e.g. because of a fundamental disagreement with the group view).
19. An enlisted Role can enter a ***refutation*** of (but not delete) an enlistment to indicate that it believes the enlistment is inappropriate. A refutation can be cancelled only by the enlisted Role that entered it.
20. Any Role can get a list of all Representative Roles with which it is associated, together with the cancellation and refutation state of itself and each other associated Role

It is a matter of contractual agreement between users of the support environment to define which Enlisting Representative Roles are those whose statements are used as the basis of a contract for development of the system being specified. Similarly it is a matter for organisational discipline among users of the support environment to ensure that people do not frivolously create Roles.

In a support environment that supports multi-organisation contracts it is not necessarily possible or desirable to have one controlling authority ruling on which Roles can be set up and how each is to be interpreted. The one key thing that must exist external to the support environment is a contractually binding agreement on which Roles are to be taken as contractual and at what date/time such Roles' statements are to be frozen for contractual purposes. This implies one further criterion for a viable support system:

21. It is possible to identify the date/time at which each statement of each Role is made.

Practical realisation of a support environment satisfying all the above criteria requires combinations of tools derived from tools which currently exist. In particular, current database systems and special purpose languages (like Prolog and/or Lisp) need to be made mutually compatible, which I contend they currently are not, even though they may be used to exchange information.

They need also to be extended to support interwoven threads of relationships similar to those I described in the July Newsletter contribution: Darn Those Threads. That interweaving requires an extension of what appears to be

RE-Bites...

[couldn't resist this one – for some reason it made me think of the LAS – ed.]

An Ambulance Down In The Valley
Anon

Twas a dangerous cliff as they freely confessed,
Though to walk near its edge was so pleasant.
But over its edge had slipped a Duke,
And it fooled many a peasant.

The people said something would have to be done,
But their projects did not at all tally.
Some said, "Put a fence around the edge of the cliff,"
Others, "An ambulance down in the valley."

The lament of the crowd was profound and loud,
As their hearts overflowed with pity;
But the ambulance carried the cry of the day,
As it spread to the neighboring cities.
So a collection was made to accumulate aid,
And dwellers in highway and alley,
Gave dollars and cents not to furnish a fence,
But an ambulance down in the valley.

"For the cliff is all right if you're careful," they said,
"And if folks ever slip and are falling;
It' not the slipping and falling that hurts them so much,
As the shock down below when they're stopping."

And so for the years as these mishaps occurred,
Quick forth would the rescuers sally,
To pick up the victims who fell from the cliff,
With the ambulance down in the valley.

Said one in his plea, "It's marvel to me,
That you'd give so much greater attention,
To repairing results than to curing the cause;
Why you'd much better aim at prevention.
For the mischief, of course, should be stopped at its source;
Come friends and neighbors, let us rally!
It makes far better sense to rely on a fence,
Than an ambulance down in the valley."

"He's wrong in his head," the majority said,
"He would end all our earnest endeavors.
He's the kind of a man that would shirk his responsible work,
But we will support it forever.
Aren't we picking up all just as fast as they fall,
And giving them care liberally?
Why a superfluous fence is of no consequence,
If the ambulance works in the valley."

Now this story seems queer as I've given it here,
But things oft occur which are stranger.
More humane we assert to repair the hurt,
Than the plan of removing the danger.
The best possible course would be to safeguard the source,
And to attend to things rationally.
Yes, build up the fence and let us dispense,
With this ambulance down in the valley.

currently possible via HTML and Java on the Internet – and the extension must be integrated with use of the improved database and special purpose tools mentioned above.

What I have left unsaid here is anything about the mechanics of how some of these things can be achieved. That is partly because I am not competent to put the whole lot together, but partly because the things I can say would take up too much space. My intention is to say more about the mechanics of it in a later contribution.

Geoff Mullery
geoff_mullery@dial.pipex.com

RE-Calls

Recent Calls for Papers

Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'98), Pisa, Italy, June 8-9 1998.

http://www.ifi.uib.no/konf/refsq98/cfp98.html

Purpose

The ultimate measure of software quality is the degree to which user requirements are fulfilled by a system, its fitness for use. Early elicitation and correct definition of requirements prevents costly rework during later development stages and provides the foundation for building high quality systems. Therefore, requirements engineering is considered as a more and more crucial part of the system life cycle.

During requirements engineering the users and engineers have to find a way from an initially opaque and diverse system understanding to exact, reconciled and at least partially formalized software specifications. A multitude of methods from software engineering, ethnology, social sciences and psychology have been adapted to support this process, and to improve the requirements specification as a foundation for higher software quality. Most of these methods are relying on adequate specification languages that are expressive and formal enough so that the represented quality requirements can be verified or validated.

At the REFSQ'94, REFSQ'95 and REFSQ'97 workshops researchers and practitioners from various disciplines presented approaches to improve the definition and implementation of quality requirements. The success of the earlier REFSQ's has encouraged us to provide a follow-up workshop REFSQ'98 as a stage for the discussion of quality-related problems in requirements engineering as they have developed over the last year. In particular, we like to encourage people from the requirements engineering, software engineering and information systems fields to present their approaches to higher software quality and to discuss how requirements engineering can contribute to it.

Goal

The main goal of REFSQ'98 is to bring together people working in the fields of requirements engineering, software engineering and information systems focussing on the

- specification of quality requirements;
- their traceability back to user needs and forward to the design;
- their realization by SE and IS development methods;
- the measurement of their achievement, as well as
- consolidating the achievements of the three previous REFSQ workshops.

Themes

REFSQ'98 invites contributions from research and industry within the following four main themes:

Managing the quality of RE processes: Relevant topics include: change management, procurement, RE for COTS, process modelling and monitoring, RE project organisation, quality models of RE and the RE process, environments for supporting RE processes, etc.

RE methods and techniques.: Relevant topics include: RE using scenarios, goals and agents, RE with multiple viewpoints, traceability, etc.

Quality assurance and RE: Relevant topics include: models for quality assurance, considering quality assurance in RE, software quality and RE, specification of software quality requirements, measuring the quality of requirements, etc.

Mapping requirements specifications to software architecture and design: Relevant topics include: transformation and mapping methods, the interplay between requirements and software quality features, formal representation methods, etc.

Note that the list of topics for each theme is not intended to be exhaustive. High-quality papers on other topics within any of the four themes are also welcomed.

Papers

Papers of three types can be submitted to the workshop:

Full papers should emphasize what is new and significant about the chosen approach and adequately compare it with similar work. Integration of the contributions with mainstream SE and RE methods and products (like SA, OMT, ER, and the like) are especially encouraged. The maximum length of a full paper is 6000 words.

Position papers should state the author's research position with respect to view of current RE practice, relations between current RE practice and RE research, and/or research methodology and ontological assumptions. Papers should emphasise which topics that are of particular concern to the RE community at present, and why. The maximum length of a position paper is 2000 words.

Industrial problem statements: People from industry are especially encouraged to submit problem statements. Industrial problem statements should focus on perceived mis-matches between current RE practice and research and/or on emerging areas of concern for RE practitioners. An industrial problem statement may be about one page long, but more comprehensive statements will also be considered.

Workshop Format

The workshop will be an interactive forum. Attendance will be limited to 30 people and all participants must have a paper accepted for the workshop. Authors who do not attend the workshop will not have their papers published in the proceedings.

The workshop language is English.

The workshop will be organized in conjunction with the CAiSE*98 conference, and all workshop participants must also attend the main conference.

The accepted papers will be made available electronically to all workshop participants before the workshop, so that presentations can be kept short. Each presentation will be summarized and commented on by a discussant, on behalf of all the other authors in the same session. The discussant will be followed by a plenary discussion of the paper. In addition, there will be a plenary discussion at the end of each session.

At the end of the workshop there will be a general discussion, including a brainstorming session about areas or topics the participants would like to see more RE research on.

Email sessions between the participants will be organised before the workshop to prepare for fruitful and focussed plenary discussions and brainstorms. For more information, see the workshop preparation kit at:
http://www.ifi.uib.no/konf/refsq98/kit98.html

Instructions for Authors

Send your full paper (max. 6000 words), position paper (max. 2000 words) or industrial problem statement (approx. 1 page) by e-mail or via normal post before March 9th (arrival date) to:
REFSQ'98
c/o A L Opdahl
Department of Information Science
University of Bergen
N-5020 Bergen
Norway

Email submission is encouraged (to Andreas.Opdahl@ifi.uib.no), as the accepted papers will later be distributed to the other participants electronically.

Papers will be published in the REFSQ'98 workshop proceedings, and the papers will be made available electronically for the participants before the workshop. We also intend to provide paper preprints at the beginning of the workshop, but this is dependent on the funding we receive.

Important Dates

Submission deadline: March 9th 1998
Acceptance notification: April 9th 1998
Confirmation of participation: April 24th 1998
Camera ready paper due: May 9th 1998

Organization

Eric Dubois
Dept of Computer Science
FUND Namur
Namur D-52074
Belgium
Tel. +32-81-72 41 11
Fax. +32-81-72 49 67

edu@info.fundp.ac.be
Andreas L. Opdahl
Dept of Inf Science
Univ of Bergen
N-5020 Bergen
Norway
Tel. +47-5558-4115
Fax. +47-5558-410
Andreas.Opdahl@ifi.uib.no

Klaus Pohl
Informatik V
RWTH Aachen
Aachen
Germany
Tel. +49-241-80-21512
Fax. +49-241-8888-321
pohl@informatik.rwth-aachen.de

Sixth European Workshop on Software Process Technology (EWSPT-6), September 16-18, near London, UK

http://www-dse.doc.ic.ac.uk/~ban/misc/ewspt98.html

Sponsored by: ESPRIT BRWG PROMOTER (Process Modelling Techniques: Basic Research)

The software process community has developed a wide range of process modelling languages, process modelling tools, and mechanisms for supporting the enactment of software processes. The focus of this workshop is on extending the focus of this research to the application of software process technology in practice.

To achieve this, EWSPT-6 is soliciting three kinds of papers:

Position papers: which present concise arguments about an area of software process resaerch or practice (in less than 2000 words). Position papers should not be incomplete versions of full papers.

Full papers: which describe authors' novel research work (motivated, presented and evaluated in less than 6000

words). Full papers must be original contributions, not published, accepted or submitted for publication elsewhere.

Industrial reports: which describe real-world experiences in managing software processes (less than 6000; short papers are also welcome).

To emphasise the broadened focus of the workshop, its organisation will incorporate a variety of new kinds of sessions. These include:

- Academics on trial: In sessions of this type, academics will attempt to "sell" their research to practitioners, who, in turn, will demand economically usable technology.
- Industrial presentations: In sessions of this type, practitioners will explain their requirements and experiences of process technology.

The programme committee will select a subset of accepted papers for different kinds of presentations at different workshop sessions.

Proposals for panel sessions are also solicited.

Attendance will be limited to 40 people. Invitation is based on paper submissions. The workshop language is English. The proceedings will be published by Springer-Verlag as part or their Lecture Notes in Computer Science series.

Please send 4 copies of your submission to:
Volker Gruhn
Informatik 10
University of Dortmund
D-44221 Dortmund
Germany
Phone: +49 231 7552782, Fax: +49 231 7552061
Email: gruhn@ls10.informatik.uni-dortmund.de

Submission Deadline: 1 March 1998
Acceptance Notification: 5 May 1998
Camera ready paper due: 2 June 1998
Workshop: 16-18 September 1998

Organisation

General Chair: Bashar Nuseibeh, Imperial College, London, UK

Programme Chair: Volker Gruhn, University of Dortmund, Germany

Programme Committee:
Nacer Boudjlida, CRIN, Nancy, France
Jean-Claude Derniame, CRIN, Nancy, France
Gregor Engels, University of Paderborn, Germany
Alfonso Fuggetta, CEFRIEL and Politecnico di Milano, Italy
Bertil Haack, WBRZ, Berlin, Germany
Carlo Montangero, University of Pisa, Italy
Bashar Nuseibeh, Imperial College, London, UK
Lee Osterweil, University of Massachusetts, Amherst, USA
Brian Warboys, University of Manchester, UK
Vincent Wiegel, COSA Solutions, The Netherlands
Alexander Wolf, University of Colorado, Boulder, USA

13th IEEE International Conference on Automated Software Engineering (ASE'98), October 13-16, 1998, Honolulu, Hawaii, USA

http://www.ics.uci.edu/~ase98

Paper Submission: May 8, 1998 (email abstracts by May 1, 1998)

The IEEE International Conference on Automated Software Engineering brings together researchers and practitioners to share ideas on the foundations, techniques, tools and applications of automated software engineering technology. Both automatic systems and systems that support and cooperate with people are within the scope of the conference, as are computational models of human software engineering activities. ASE-98 encourages contributions describing basic research, novel applications, and experience reports. The solicited topics include, but are not limited to:

- Architecture
- Automating software design and synthesis
- Automated software specification and analysis
- Computer-supported cooperative work, groupware
- Domain modeling
- Education
- Knowledge acquisition
- Maintenance and evolution
- Process and workflow management
- Program understanding
- Re-engineering
- Requirements engineering
- Reuse
- Testing
- User interfaces and human-computer interaction
- Verification and validation

The ASE Conference, formerly called the Knowledge-Based Software Engineering Conference, has for the past decade provided a forum for researchers and practitioners to discuss the application of automated reasoning and knowledge representation to software engineering problems. In conjunction with the name change a year ago, the scope of the conference expanded to encourage international participation and to reach other scientific communities concerned with formal methods, partial evaluation, process support, human-computer interface support, requirements engineering, reverse engineering, testing, or verification & validation.

All accepted papers will be published in the proceedings. In addition, several of the highest quality papers will be selected for a special issue of The Journal of Automated Software Engineering (Kluwer). ASE-98 will also include invited talks, tutorials, panel discussions, and project demonstrations. Separate calls will appear for participation in some of these activities.

Submission Information

Papers should not exceed 6000 words in length, with full page figures counting as 300 words. Papers will be reviewed by at least three members of the program committee according to: technical quality, originality, clarity, appropriateness to the conference focus, and adequacy of references to related work. Papers that exceed the length restriction will not be reviewed. Application papers and experience reports should clearly identify their novel contributions and lessons learned.

Six copies of each submitted paper should be sent to Dr. David Redmiles at the address below. No fax or electronic submissions will be accepted. The submission deadline is May 8, 1998. A paper's title, authors, and abstract should be submitted by May 1 through electronic mail to ase98@ics.uci.edu.

General Chair

Alex Quilici
Department of Electrical Engineering
University of Hawaii at Manoa
2504 Dole Street
Honolulu, Hawaii 96822, USA
Tel: +1 808 956-9735
Fax: +1 808-956-3427
Email: alex@wiliki.eng.hawaii.edu

Program Chairs

Bashar Nuseibeh
Department of Computing
Imperial College
180 Queen's Gate
London SW7 2BZ, UK
Tel :+44 171 594-8286
Fax: +44 171 581-8024
Email: ban@doc.ic.ac.uk

David Redmiles
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
USA
Tel: +1 714 824-3823
Fax: +1 714 824-1715
Email: ase98@ics.uci.edu

RE-Sources

For a full listing of books, mailing lists, web pages and tools that have appeared in this section in previous newsletters, see the RQ archive: http://research.ivv.nasa.gov/~steve/resg/

Web Pages

The BCS RESG home page can be found at:
http://porta.cs.york.ac.uk/bcs/resg/

Back issues of Requirenautics Quarterly:
http://research.ivv.nasa.gov/~steve/resg/

Compendium of Software Engineering Tools:
http://www.methods-tools.com

Books

Biren Prasad "Concurrent Engineering Fundamentals. Volume II: Integrated Product Development". Prentice Hall, 1997. ISBN # 0-13-396496-0

Mailing lists

Software Requirements Engineering Mailing List

To subscribe to the Software Requirements Engineering (SRE) mailing list, e-mail listproc@jrcase.mq.edu.au, with the only line in the body of the message:
subscribe SRE your-first-name your-second-name

Articles to the SRE mailing list should be sent to SRE@jrcase.mq.edu.au.

Tools

TicketOStar is a computer-aided Requirements Engineering tool. Version 1.0 provides method, process and product guidance to for RE. Please try it and and feel free to distribute it and assess it. Return comments, remarks and suggestions to christian.clercin@syfed.refer.mg

http://www.refer.mg/madag_ct/edu/fianar/eni/ticketos.htm

RE-Actors

The committee of RESG

Chair: Dr. Bashar Nuseibeh, Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ. ban@doc.ic.ac.uk, Tel: 0171-594-8286, Fax: 0171-581-8024

Treasurer: Dr. Neil Maiden, Centre for HCI Design, City University, Northampton Square, London EC1V OHB. N.A.M.Maiden@city.ac.uk, Tel: 0171-477-8412, Fax: 0171-477-8859

Secretary: Wolfgang Emmerich, City University, Department of Computer Science, Northampton Square, London EC1V OHB. W.Emmerich@cs.ucl.ac.uk, Tel: +44 171 504 4413, Fax: +44 171 387 1397

Membership Secretary: Dr. Sara Jones, School of Information Sciences, University of Hertfordshire, Hatfield, AL10 9AB. S.Jones@herts.ac.uk, Tel: 01707 284370, Fax: 01707 284303

Industrial Liaison Officer: Dr. Orlena Gotel, Systems and Software Engineering Centre, Defence and Evaluation Research Agency, St. Andrews Road, Malvern, Worcestershire, WR14 3PS. olly@hydra.dra.hmg.gb, Tel: 01684 894674

Publicity Officer: Dr. Andrew Vickers, Department of Computer Science, University of York, Heslington, York Y01 5DD, andyv@minster.york.ac.uk, Tel: 01904 434727, Fax: 01904 432708.

Events Officer: Carol Britton, Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, UK. AL10 9AB, c.britton@herts.ac.uk, Tel: 01707 284354, Fax: 01707 284303

Newsletter Editor: Dr. Steve Easterbrook, NASA/WVU Software IV&V Facility, 100 University Drive, Fairmont, WV 26554, USA. steve@atlantis.ivv.nasa.gov, Tel: +1 (304) 367-8352, Fax: +1 (304) 367-8211

Newsletter Associate Editor (Features and Book Reviews): Dr George Spanoudakis, City University, Department of Computer Science, Northampton Square, London EC1V OHB. gespan@cs.city.ac.uk, Tel: 0171 477 8000 ext. 3701, Fax: 0171 477 8587.

Newsletter Associate Editor (Features and Event Reviews): Ian Alexander, 2 Dornton Road, London SW12 9ND. iany@easynet.co.uk. Tel: 0181 675 6915

RE-Creations

How to contribute to RQ

Please send contributions to Steve Easterbrook (steve@atlantis.ivv.nasa.gov) before the publication deadline. Submissions must be electronic copy, preferably plain ASCII text. A list of the kinds of contributions we welcome can be found in the January 1996 newsletter, or on the web at:

http://research.ivv.nasa.gov/~steve/resg/rq5/ReCreations5.html

	Copy deadline
Issue 14 (April)	27th March 1998
Issue 15 (July)	26th June 1998
Issue 16 (October)	25th September 1998