



Requirements Quarterly

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

© 2005 BCS RESG

<http://www.resg.org.uk>

RQ37 (September 2005)

Contents

<i>RE-Soundings</i>	1	<i>RE-flections</i>	12
From the Editor	1	Nuclear Physics for Cleaning Ladies	12
Chairman's Message	1	Special Guest Proverb	13
<i>RE-Treats</i>	2	<i>RE-Implementations</i>	13
<i>RE-Calls</i>	2	RETH: a Freeware Tool	13
Training Courses in October	2	<i>RE-Publications</i>	15
ICSSEA 2005	2	DVD Review: Requirements: Best Practices	15
Mastering the Requirements Process	2	Book Review: Customer-Centered Products	15
CAISE	3	Book Review: Just Enough Requirements Management	17
RE'06	3	<i>RE-Sponses</i>	18
<i>RE-Readings</i>	3	<i>RE-Sources</i>	18
AGM & Tool Vendors Event	3	Books, Papers	18
Annual General Meeting	4	Mailing lists	18
RE'05	6	<i>RE-Actors: the committee of the RESG</i>	19
REFSQ	9		
<i>RE-Papers</i>	10		
Does Object-Oriented Domain Analysis Work?	10		
<i>RE-verberations</i>	11		
Tell Me About Your Mother	11		

RE-Soundings

From the Editor

In this issue, we naturally report on the major Requirements event of the year, the international conference RE'05 in Paris. Associated with that are a whole string of workshops.

In the Papers section, we have some special treats in store for you, while as usual there are some surprising and alternative takes on how to do requirements in the Reverberations and Reflections sections.

Al Davis has come up with yet another excellent book, Just Enough Requirements Management. We also have our first-ever DVD review, of Ivy Hooks' lecture series on Requirements: Best Practices.

In line with this practical industry theme, RQ this quarter reviews a freeware tool, RETH, starting from scratch and ending up with a correct hypertext

specification.

Your committee is infused with new blood and enthusiasm for the next year, and we have some exciting events planned. Come along and enjoy them.

*Ian Alexander,
Scenario Plus*

Chairman's Message

The RESG held its annual general meeting at City University in July. As usual, we had to wave farewell to some past stalwarts of the committee, the sadness only tempered by being able to welcome new members.

The departing members were Carina Alves and Juan Ramil, student officer and membership secretary respectively. Both did a wonderful job and we should all be grateful to them. The new members are Lucia

Rapanotti from the Open University (who has actually been hard at work with us for several months already), Ljerka Beus-Dukic from the University of Westminster, Zachos Konstantinos from City University and Andrew Stone from Lancaster University. A very warm welcome to all of them.

We were fortunate this year to have four companies participating in our tool vendors' challenge that took place following conclusion of the AGM's formalities. You can read about how they rose to the challenge later in this issue.

In addition to our own events, this has been a good RE year for Europe-based folks. The main event, inevitably, was RE'05 which took place in Paris and which you can read all about below.

This only comes to Europe every three years or so. Interestingly, I'm told that when it is held in Europe, attendance is far higher than when it's held elsewhere. That reflects the fact that there's a concentration of RE research activity in Europe that is genuinely envied by researchers elsewhere.

The UK is particularly fortunate in this respect. A glance at the membership section of the AGM report will show that membership is healthy within the ten

Universities that hold corporate RESG membership and which represent the core centers of academic RE research in the UK. Unfortunately, that's the one positive sign in our membership profile, with only five companies holding corporate membership and all other categories of member down. In other words, the UK's active research community might not reflect the same level of sustained interest or investment in the practice of RE within industry. Since the RESG exists to help bridge the research/practice gap, that's a worry.

Now membership profiles tend to go up and down and the underlying reasons are as hard to read as Shane Warne's flipper. However, we would like to hear from members, particularly industrial members, how we might better serve their needs. As the success of creativity workshops shows, there's an active, design element to formulating requirements, and we have invented ideas about how to keep the RESG interesting for members. But we need to elicit requirements from the users too. So that's what I'm doing here. Please have a look at our projected events and think about the mix of services we offer through RQ and the website and, if there's something missing, let us know.

Pete Sawyer
Computing Department, Lancaster University

RE-Treats

For further details of all events, see www.resg.org.uk

Forthcoming events organised by the RESG:

RE for Safety and Dependability:
7th December 2005, Newcastle University

Use Cases and Scenarios Day:
9th February 2006, Northampton Suite, City University, London: 1-day Tutorial, and 4 talks on applications of Scenarios and Use Cases in industry.

Introduction to Requirements seminar:
May 2006, IEE, Savoy Place, London:
<http://www.iee.org/Events/intro-req.cfm>

Problem Frames:

May 2006, Open University, Milton Keynes

AGM and Distinguished Speaker Event:
July 2006, London

IEE/RESG Requirements Days:

2nd October 2006, IEE, Savoy Place, London: 8 talks covering all the essentials of requirements work, followed by a Banquet.

3rd October 2006, IEE, Savoy Place, London: 1-day Introduction to Requirements seminar.

<http://www.iee.org/Events/intro-req.cfm>

Early Aspects:

October 2006, Lancaster

RE-Calls

Recent Calls for Papers and Participation

Training Courses in October

Requirement Specification, 3-4 October; Evolutionary Project Management, 5-6 October; and Agile Inspection, 7 October. 10% discount for BCS & RESG members.

<http://www.testing-solutions.com/gilb>

ICSSEA 2005

18th International Conference on Software & Systems Engineering and their Applications

November 29, 30 & December 1st, 2005, Paris, France

<http://www.cnam.fr/CMSL>

Mastering the Requirements Process

20-22 February 2006, London, presented by Suzanne Robertson, Atlantic Systems Guild

This 3 day seminar & workshop presents a complete process for eliciting the users' requirements, testing for correctness and recording them clearly, comprehensibly and unambiguously.

Delegates will learn to:

- Determine their client's needs, exactly
- Write complete, traceable and testable requirements
- Precisely define the scope of the project
- Discover the stakeholders and keep them involved
- Get the requirements quickly and incrementally

<http://www.irmuk.co.uk/1>

CAISE

5-9 June 2006, Luxembourg
<http://www.tudor.lu/caise06>

RE'06

11-15 September 2006, Minneapolis/St Paul, USA

RE-Readings

Reviews of recent Requirements Engineering events.

AGM & Tool Vendors Event

Wednesday 20th July 2005

The RESG Chairman Pete Sawyer took us through the AGM in relaxed style.

See Minutes below in this issue of RQ.

During the year we ran 5 events: An Audience with David Parnas; Modelling your System Goals: The i* Approach; a PhD Workshop; the Birds-of-a-Feather / Book Launch event; and our regional event which this year was on Defence at MoD Abbeywood, Bristol.

We co-sponsored RE'04 Kyoto in September 2004 (and have similarly sponsored RE'05 Paris this year.)

Members are the RESG's reason for existing. We'd like to hear from members what kinds of event they would like, and what they want to read about in RQ. Members' letters, articles, or other contributions are very welcome in RQ itself.

Membership has fallen slightly: we don't know why as meeting attendance is very healthy, as are our finances. Let us know what you think and what you'd like to see.

Vendors Challenge

Ian Alexander presented the challenge to Vendors (see RQ36), essentially to show how their tools support the core activities of requirements management on an easy-to-explain automated car park project.

MKS (said Ryan Lloyd) handles the information structure with templates. Import is from Word or other tools, though MKS supports direct editing. Word requirements are tagged (with attributes in the document) and then managed directly by a tool integration. When requirements change, the impact can be traced via an Explorer-type hierarchical list with top-level requirements decomposing into several product features (child objects). These features are flagged as suspect if their parent changes.

Tracking of status changes is also provided in the form of graphs summarising task breakdown by risk, trends over time, and task assignment to development. Equally MKS can produce HTML reports on how the requirements have been decomposed.

Validating completeness through traceability is via project reporting, complex querying, and a relationship view – again, hierarchical, with columns to show related text. The project report (based on a query) is organised into chapters and sections, eg the control of entrance to the car-park is analysed into detailed (specification) tasks, a Work Breakdown Structure, and its contribution to the Project Schedule.

Bashar Nuseibeh asked how change was controlled down to requirement level. MKS can drill down to the source code (even for this exercise, code skeletons had already been created, we saw) that relates to each requirement; this is more than a typical Change Management tool can do.

Alistair Mavin asked if a feature could trace back to multiple requirements. MKS can do this; it appears twice in the display, but is only stored once.

Stakeholders can be handled simply as attributes of the requirements, or in a separate section. As regards stakeholders within a project, electronic signatures can be attached to authorise changes; people can equally be notified of changes by email.

The MKS Integrity Suite enables seamless management of the software lifecycle from requirements capture through to final deployment. It's a client-server architecture. People can work remotely via a proxy.

Project issues are recorded alongside the requirements. You can also deal with requirements that span multiple projects, eg for product families.

Telelogic's DOORS (said Toby Dupont) imports documents by integration from Word and other tools. Requirements are represented as database records within document-like 'modules'. Status, priority etc are represented as attributes of requirement objects. Traceability is by links between requirements; it can be displayed in columns and navigated via coloured triangular indicators (pointing left for incoming, right for outgoing). References to dictionary items, etc, is also handled by traceability.

Change can be displayed by a suspect links mechanism. Change is controlled both at individual requirement level and by baselining at module level (and indeed for groups of modules also). Filtering makes it easy to identify untraced or unvalidated requirements, etc.

Reporting is by filtering (on queries), and by export to Word, HTML etc. HTML export creates a fully-navigable set of specifications.

Use case and other UML diagrams can be drawn within DOORS modules; there are also integrations with many diagramming and software design tools such as Rose.

IBM's Requisite Pro (said Tony Kesterton), like all vendors, faces the challenge of actually getting people in industry to recognise requirements at all. So you need something of 'low impact' for them to work with. Requirements go right from the business process to executable code.

ReqPro handles requirements, stakeholder lists, dictionary, references etc in any number of documents in Word (though HTML is also possible). They are held in a database, ie there is a Word-centric view of the world. Requirements can also be created and edited directly in the database.

Priority, status etc are handled as database fields or attributes. Kesterton gave his USB drive to the Telelogic team to get their requirements to import, but it crashed their machine. He said he wasn't surprised (laughter, and suggestions of Vendor rivalry). So he typed in some requirements and stakeholders instead.

Documents can be based on templates, eg MIL-STD or RUP. Then requirements can be typed in directly as usual in Word, the only visible difference being an extra toolbar. Behind the scenes are hidden Word attributes; marked-up requirements appear in a changed colour and style in square brackets.

Traceability is by means of cross-reference links between requirements. A trace matrix can be viewed directly in a compact grid. Anything can be linked to anything.

Predefined attributes are provided for Status, Priority (and Release Number – including 'none'), Stability, Cost and so on. User-defined attributes can be added (as with MKS and DOORS).

Suspect traces can be indicated when linked requirements change. Baselines can be created and compared to handle change on a larger scale.

Test Cases can be set up, based on Use Cases. UML models can be maintained in the background.

People can share the database for editing, but only one person can edit a document at a time. Offline working on documents is also possible.

Objectiver (said Robert Darimont) is not yet another complex text-based requirements tool. Its mission is to bridge the gap between business and IT development. It's based on the KAOS goal modelling approach: requirements are just a part of that.

Inputs are from documents, interviews, etc followed by simple text editing. (Graphical) Outputs can be to HTML, etc. Models are made of goals (parallelograms), obstacles (red parallelograms sloping

backwards), expectations, domain properties (houses), agents (hexagons), entities (UML rectangles), associations (UML diamonds), and operations (ellipses).

Objectiver can start from scratch or from templates. Traces can be up to source documents, down to requirement reports, or "inwards" between model elements such as goals.

The source document (the challenge published in RQ) can be marked up (annotated) and linked to model elements such as access control, car park exit, barrier, payment machine.

But requirements, expectations, stakeholders and goals are all modelled as KAOS elements, linked to each other; there is no mass of textual documents. The model has a goal (deter theft and non-payment) which is refined into subgoals and solutions (eg CCTV). Asking WHY moves up to the top-level goals; asking HOW moves down towards solutions. Operations like 'Open barrier' operationalise (line with blue circle) goals like 'Exit barrier opened only when paid-for vehicle is in slot'. Other relationships exist between other element types, eg Agents have responsibility for Requirements.

Attributes handle things like Priority.

Change is supported with facilities for reviewing (models read-only, only annotations allowed). Links can be followed for impact analysis. A query language OQL is embedded, allowing search for all requirements not traced to the model, or all unrefined goals, etc. Changed documents are produced by updating the model and simply regenerating the documentation.

The dictionary/glossary is produced from the model's definitions, so it should be a well-structured UML-style glossary.

Requirements documents are based on the IEEE-830 standard.

"And the winner is Objectiver + your favourite RM tool", said Darimont: Objectiver wasn't a direct competitor of the other tools.

All the vendors very kindly kept to the challenge, showing how their products would handle the stated problem. The level of questioning and audience attention showed that everyone was extremely interested in what they had to say.

Kathleen Maitland ran a raffle to choose someone to have the RESG's sponsorship ticket to RE'05. The lucky winner was Celeste Belon of MKS. Congratulations to her; for the rest of you, turn up to the AGM next year for your chance!

Annual General Meeting

Date: 20th July 2005 at 1.00pm.

Location: City University, London

Pete Sawyer welcomed us to the 11th AGM of the RESG

Minutes of last AGM:

These (published in RQ33) were approved with no matters arising.

Events (since last AGM, July 2004)

An Audience with David Parnas: 25th May 2005, Imperial College, London

Modelling your System Goals: The i* Approach: 20th April 2005, City University, London (open session); 21st and 22nd April 2005, City University, London (closed session)

PhD Workshop: 8th December 2004, University College London

Birds-of-a-feather and book launch event: 12th November 2004, City University, London

Defence Event : 19th October 2004, MoD Abbeywood, Bristol

Related/Co-sponsored events:

RE'04, 6-10 September 2004, Kyoto, Japan

Events scheduled for the coming year:

Tool Vendors' Challenge: 20th July 2005, City University, London

RE for Safety/Dependability: 7th December 2005, Newcastle University

Use Cases and Scenarios: 9th February 2006

Problem Frames: May 2006, Open University, Milton Keynes

AGM and Distinguished Speaker Event: July 2006

IEE/RESG Requirements Day: 2nd October 2006, IEE, Savoy Place, London

Early Aspects: October 2006, Lancaster

Newsletter: Requirements Quarterly (RQ)

Editor: Ian Alexander

As hinted at by the name, four issues have been produced during the last year:

RQ33: September 2004

RQ34: December 2004

RQ35: March 2005

RQ36: June 2005

The newsletter's name has changed from *Requireonautics Quarterly* to *Requirements Quarterly*. (Steve Easterbrook, the first RQ Editor, remarked that the old name was guaranteed to be found at once if Googled, but we aren't sure that's a requirement.)

Due to increasing requests to receive RQ electronically and the cost of printing, the default distribution mechanism will be by email w.e.f. RQ37.

The audience was invited to submit articles, letters, proverbs, etc. to the Editor.

Back issues of RQ are available on the RESG website.

Industrial Liaison

A major part of the RESG's mission is to serve our industrial constituency so almost all RESG events bring together researchers and practitioners. The RESG committee includes an Industrial Liaison Team.

The RESG in the regions

The past year has seen only one event held outside London; in Bristol. An event was scheduled for Lancaster but had to be cancelled.

During the coming year, however, events are scheduled for the OU (Milton Keynes), Newcastle and Lancaster.

Membership

Last year's downward trend in membership has continued, from 348 in 2003 to 294 in 2004 to 258 in 2005.

This breaks down as follows:

Individual members (2004 figure):

- BCS/IEE members: 62 (83)
- Non BCS/IEE members: 17 (32)
- Students: 19 (36)

Corporate members:

- Industrial: 45 (60) in 5 companies
- Academic: 114 (83) in 10 universities

The reason for the decline is unclear and Pete invited members to let the committee know if the service offered by the RESG could be improved to make membership more attractive.

The membership goals for 2005-6 are to:

- Increase industrial membership
- Increase student membership
- Encourage corporate membership renewals

Publicity Report

In contrast to the decline in membership, subscribers to the RESG's email distribution list have increased from 982 last year to the current figure of 1071. 813 of these are non-members.

Pete summarised the policy for advertising non-RESG events on the mailing list, explaining that only genuinely RE-related events are advertised since the RESG certainly does not wish to increase the volume of spam received directed at members' mailboxes. The website is used to advertise more tangentially RE-related information but even this must satisfy the publicity officer, Will Heaven, that it is of some interest to members.

Treasurer's Report 2005

Pete reported that total reserves stood at £31890.61 (to be validated by the BCS). This comprised:

EXPENDITURE: Various items: meeting, newsletter and i* event costs: £3208.22

INCOME: Various items: membership, i* event fees, accounts interest: £4530.77

SURPLUS for 2004/5: £1322.55

BANK ACCOUNT AT 01/05/05: £7068.71

SAVINGS ACCOUNT AT 01/05/05: £24821.90

Election of RESG Executive Committee

Before asking the meeting to approve the appointment of the new executive committee, an on behalf of the RESG membership, Pete thanked the outgoing student officer, Carina Alves, and membership secretary, Juan Ramil, for all their work for the RESG.

The committee for the coming year comprises:

- Patron: Michael Jackson
- Chair: Pete Sawyer
- Vice Chair: Kathy Maitland
- Secretary: David Bush
- Treasurer: Neil Maiden
- Membership Sec: Lucia Rapanotti
- Publicity Officer: William Heaven
- Newsletter Editor: Ian Alexander
- Newsletter Reporter: Ljerka Beus-Dukic
- Regional Officer: Stephen Armstrong
- Student Officers: Zachos Konstantinos, Andrew Stone
- Immediate Past Chair: Bashar Nuseibeh
- Industrial Liaison Team: Wolfgang Emmerich, Alistair Mavin, Suzanne Robertson, Gordon Woods

RE'05



Victor Hugo gazes at RE'05 in the Sorbonne, Paris

The Thirteenth IEEE International Requirements Engineering Conference was held in the incomparable splendour of the Sorbonne University, Paris. Outside, under bright blue skies, the beautiful buildings of Paris' 5me district showed off their architecture. Within the quadrangle, Victor Hugo and Louis Pasteur offered their literary and scientific encouragement to rational thought. Inside, in the main amphitheatre, the sculpted gods of Athens (on the right) and Rome (on the left)

looked down on the speakers, while Paris sat at the back, her ancient sources of inspiration on either hand, urging the conference to higher things.

The first keynote talk was given by **Daniel Jackson** of MIT. He spoke fluently and entertainingly, with something of the manner of his father, on Dependable Software: An Oxymoron? Certainly software is important, but it rarely kills, contrary to myth. On the other hand, far more of it is critical to business and to safety than is commonly recognised: perhaps there is a looming time-bomb in the shape of incautious integration of COTS products into critical systems. The USS Yorktown found itself dead in the water, its entire network down, after a simple divide-by-zero error. Perhaps a network-enabled battlefield would be a very vulnerable force?

So far, software has killed far fewer people than automobiles have, or tired doctors for that matter. Other myths include the familiar blaming the operator - all too easy in aviation, when the pilots are usually dead. Failure is little understood; it is normal (where correctness is not); and accidents usually have many causes. Hindsight bias does not help, either. As far as software propaganda is concerned, requirements cannot be complete, and even if they were, that would not ensure safety. Correctness does not mean dependability, and testing doesn't either (it doesn't even mean correctness, of course). Certification is mainly a tail-covering, after-the-fact exercise; it does little to improve systems, and often doesn't even use the same documentation as development. Feel like flying, anyone?

The talk provoked an interesting discussion (not a common event after a keynote). Dan Berry observed that over-automation caused deskilling, which worsened over time. Similarly, certification's value decayed as circumstances changed.

Alistair Sutcliffe was sceptical of safety figures like 10^{-5} failures per hour. We needed rather to think about the complicated causality of accidents.

RQ readers may like to look at the dependable software study's website:

http://cstb.org/project_dependable

It's always impossible to know which papers to hear, or to decide which are the most interesting.

Svetinovic et al spoke on the problem of concept identification in Object-Oriented Domain Analysis (OODA). The paper was subtitled 'Why some students just don't get it'. The symptoms of OODA are becoming familiar: the models are under-specified; different students (and practitioners?) make "drastically different" models of the same system. Eg ten published specifications of the hoary old elevator/lift problem show very little overlap. There were in all 44 concepts identified, but in three typical case studies there were respectively 6, 19, and 6 concepts actually used.

That led to the Fundamental Difficulty of OODA: it's hard (even on small problems like the elevator). Why?

Because the concepts are tightly interrelated: most of them work together as a set of sub-activities; worse, many concepts share in numerous activities. To control this complexity, analysts often ignore passive concepts, and vainly seek tidy 1:1 mappings of concepts to reality.

Instead, Svetinovic argued, you needed to postpone and constrain OODA, using other RE techniques such as statecharts first (RE readers may think plenty of other techniques might precede statecharts, too). RE does not consist of object modelling, he said.

If OODA were easy, Svetinovic argued, all students would get the same results, the same models. **Shamal Faily** asked why, in that case, use OODA at all if it wasn't suited to the problem domain? See his article discussing the impact this paper made on him, in this issue of RQ. This was an Emperor's New Clothes paper, if ever I heard one.

Steve Easterbrook spoke as entertainingly as ever on whether Viewpoints led to better conceptual models. Viewpoints are independent sub-models, representing the total points of view of different stakeholders. These are nice to trace and make it easy to explore conflicts. The idea is that you construct partial, inconsistent models (tolerating inconsistency for a while) and then somehow merge them into one complete and correct model. The theory of viewpoints says that comparing models will yield useful new insights in the form of: better traceability; improved readability; better capture of divergent/minority opinions; and better teamwork as people work on separate parts of the modelling task.

The study used just two teams (a statistically-valid experiment being out of the question, a scientific case study approach was taken). One made a single 'global' i* model which became very large. The other made many smaller i* models, which were very difficult to merge: it was labour intensive, and was never completed. It was valuable, however, as the sense-making was done twice, proving good for understanding differences.

The result was that the traceability and teamwork claims for Viewpoints were supported, but not the other two. The cost/benefit effect was not measured.

Liaskos et al talked about configuring personal software. We are familiar with the idea of making software somewhat adaptable, but too often the Options... switches are so obscure that nobody really uses them: and are they the right choices anyway. Mozilla apparently has over 80 billion combinations so presumably most of these have rarely been tested. The presentation documented some of the choices with "Huh!" and "Err, OK, Whatever". The proposed remedy to the undoubtedly tricky situation is to make i* softgoal models and to propagate "labels" (ie attribute values of users' goals). This would mean that software would have to be aware of small details of user preferences, ie that it would have to model preferences mathematically. There are obviously many unsolved problems here before such an approach could become practical.

Alistair Sutcliffe (with **Steve Fickas** et al) also looked at personal and contextual RE, and won the **Best Paper** award for it. The background is of "Stakeholders" as opposed to individuals, user interface research into the idea of the "user", and adaptive interfaces (shades of the Expert Systems of the 1980s: is anything really new?). The idea is to add new layers below the generalised stakeholder layer: user characteristics, and personal goals. Each layer contains two dimensions: changes in space, and changes in time. Eg needs vary in mobile applications, and privacy varies with place, yielding implied requirements such as to monitor where the user is. Not all such goals can be satisfied, so trade-offs are needed. These can be handled by arguing in Non-Functional terms: how well do Functional Requirement choices contribute to people's Non-Functional Goals? Cost-Benefit analysis clearly needs to be extended to deal not only with system issues but effects on people like loss of privacy from monitoring. Again, this paper opens up a whole avenue of RE research. Over coffee the audience feeling was that this was practical in niche markets such as the one chosen by Sutcliffe (caring for the disabled) where individual differences are important.

A panel of **Dan Berry**, **Daniela Damian**, **Don Gause** and **Eric Simmons** looked at why, if the RE payoff is so good, more companies are not doing it? The panellists first proposed various parts of a solution individually. These came across as somewhat piecemeal.

- Berry argued for formal methods, claiming that safety-critical software can be made reliable if done formally. He'd used Parnas' 4-variable model: "to date no errors".
- Damian pointed out the rich interaction between RE and other processes. The main way RE helped was through better collaboration between people in different areas.
- Gause said it was no good arguing with management, you had to make a solid case using scenarios with and without RE; and you had to manage expectations.
- Simmons said there was really no need to construct a case as the argument is so overwhelming: "Failure to do this is just ridiculous". He offers "free" support to projects within his company (Intel), focussing on improving business results (ie dollars). The payoff is a Return on Investment (RoI) so large that the actual ratio is irrelevant, even on first application. His team is two people in a firm of 19,000 so his work is knowledge transfer; the trained-up teams do their own RE.

So what were the barriers to RE?

- Cultural barriers included lack of a process improvement champion; large chaotic companies; lack of awareness of project failures; lack of RE training and education; geographical distribution (within projects); intolerance of planning before

doing (ie a desire to get coding before thinking); aversion to risk-taking; and pride, the belief that needing to improve is a flaw.

- Structural barriers included companies (or profit margins) too small for process improvement; and, especially, heavy layers of middle management who opposed all changes whether from above or below.
- Within RE itself, there are what we could call process barriers: low emphasis on the -ilities, which are after all the key requirements; lack of prioritisation (now chic, see the review of Al Davis' book in this issue); and not least the competing fashions within the RE business, eg scenarios versus Agile. Fashion seems to be surprisingly important, with the consequence that people who saw the last initiative fail are sceptical about the latest one.

This is a big pile of obstacles to get around. Perhaps it's surprising that RE gets going at all. What to do about it? Ah, that's the work of a future panel.

Jörg-Michael Barth spoke on getting from requirements to production on the Airbus A380. This was the first ever use of a new set of tailorable Standards (Aerospace Recommended Practice ARP 4754, 1996) on an aircraft programme.

Jean-Pierre Corniou spoke on IT within the Renault Group. His aim was to bring wisdom into practice. His business goals were the usual faster, better, cheaper, which he attained using a Deming-like cycle: Plan, Acquire, Do, Monitor. Cars were made in collaboration with many other companies; processes were aligned using the COBIT framework to achieve economies of scale.

Ian Alexander gave a report of a survey carried out with **Suzanne Robertson** and **Neil Maiden** on what influences the requirements process in industry. (Couldn't help being at that talk. Ed.) People consistently reported that the most important influences on their choice of techniques were standards, their organisation's own processes, and (paradoxically) that they knitted their processes themselves from first principles, consulting with experienced colleagues. Perhaps this meant that people felt reasonably free to choose approaches within a familiar framework, but that in fact they were strongly influenced by what they collectively knew, eg the waterfall model's 'do something called user requirements before specifying the system' etc. There were striking differences between industries, eg Telecomms rated websites as very influential but nobody else did, presumably because they have to deal with rapidly-changing protocols. In the City (finance, banking, insurance) it seemed to be easy to throw money at problems, so they were happy to buy books and training courses. In Defence, tools such as Doors were important, helping presumably to impose practices such as traceability.

Suzanne Robertson gave an inspiring Keynote Talk on Exemplars for Better Requirements. Like Isaac

Newton, we see a little further by 'standing on the shoulders of giants'. RE includes many practices known to work, and the best of these can themselves be thought of as exemplars, or perhaps as patterns. One of Suzanne's favourites is 'Rooms with Walls', ie large clear areas where notes, post-its, reminders, whiteboards, action lists etc can be put up and left up to encourage discussion, to co-ordinate actions, etc. A digital camera makes it easy to record diagrams and configurations of post-its. She praised the German habit of providing a proper Moderatorenkoffer, a box full of facilitator's goodies.

Exemplars could be sought in experience, in lessons learnt, in books and articles, in other people's success. Keeping a notebook was a simple way of benefiting from these things. Time to reflect was also vital. She challenged the audience to come up with their own Exemplars.



Champagne, Canapés and Petit Fours in the Sorbonne

The conference was superbly organised by **Colette Rolland and her team**, with the most astonishing Reception of real Champagne and elegant canapés among marble colonnades and sweeping staircases.

The Banquet was held (apparently at immense expense) in the Museum of Fairground Arts. Requirements engineers old and young (as young as 6) frolicked on the carousel of hand-painted wooden horses, played a variety of antique pinball-like games, or rode the manège, a dangerous circle made of interlinked fixed-wheel bicycles, at breakneck speed.

The hotels were as delightful as could be imagined: we breakfasted in a barrel-vaulted stone basement of great age and beauty. The place of the conference canteen was taken by half-a-dozen local restaurants, creperies, and cafés which contrived to serve delicious food within the allotted time.



In the Musée des Arts Forains

And of course Paris itself tempted most of us out for some sightseeing at the Louvre, the Rodin Museum, or the old Orsay Station, now both an architectural treasure and the home of many impressionist paintings. This was a conference to remember.

© Ian Alexander 2005

REFSQ

11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'05)

13th – 14th June 2005 Porto, Portugal (Co-located with CAiSE'05)

<http://www.refsq.org>

REFSQ was established in 1994 as a complement to the ACM international Symposium on Requirements Engineering and the IEEE International Conference on Requirements Engineering (which have since been fused into the annual IEEE International Requirements Engineering Conference – the RE series). REFSQ has emerged as arguably the most important RE workshop series in the world. Although always held in Europe, since it is co-located with CAiSE, it has a genuine international profile.

Twenty-eight people from eleven different countries attended REFSQ '05 in the beautiful city of Porto. Despite the heat and a room only just large enough, the usual strengths of REFSQ were on show. There was a healthy mix of people ranging from grizzled veterans like Dan Berry, Sjaak Brinkkemper and Neil Maiden, to students in the initial stages of their doctoral research. Many among the participants were either currently or recently employed in industry or had a foot in both academia and industry. As usual at REFSQ, the workshop benefited greatly from their rooted-ness in real problems. The carefully structured discussion sessions delivered a lively workshop over two days of intense activity.

The workshop was organised in four thematic sessions: *Understanding and Improving the RE Process*, *RE in Different Domains*, *Requirements and Quality*, and *Changes, Dependencies, Composition*. From the papers and associated discussions, we observed three distinct themes:

1. A comeback of quality features

2. New approaches to known problems
3. Fresh problems for RE research.

We'll look at these in turn.

Comeback of quality features

There was a comeback in the sense that an interest in quality issues appears to be cyclical at REFSQ; our attention may shift elsewhere from time to time, but we keep coming back to the relationship between RE and software quality.

Many of this year's papers elaborated on scenarios and use cases, linking them to coverage of needs and of tests – and thus, to the general quality attribute of *completeness*. These led to discussion of empirical issues, the importance of context (i.e., situated processes), and reflection on the scope of RE as a discipline.

A switching of focus from *consistency* to *completeness* within RE research was seen as a positive sign of a maturing discipline. However, the “good enough” question was seldom addressed – i.e., taking budget limitations into proper account to decide when to stop requirements engineering as well as testing. The subject has been studied extensively in testing, so reusing or at least considering those techniques for requirements work was posited as an important area of research.

The relative importance that industry assigns to completeness vs. consistency is interesting. The point was made that, in order to make the volume of requirements manageable, industry frequently *does not want* too many requirements. The implication of this is that stop criteria are not just linked to constraints on the implementation of the requirements, but also to constraints on the *management* of the requirements. However, the availability of tools changes the threshold for such trade-offs. A better understanding of stop criteria would also help in justifying the cost of upfront RE, by placing a cap on the time and financial costs of the activity.

New approaches to known problems

Several papers reported novel approaches to real industry-focused problems such as release planning. Nevertheless, there was an undercurrent of anxiety that we sometimes invest effort in developing new approaches for non-problems. This led to a lively debate of the nature and value of research driven by existing industry priorities versus more speculative research.

Ten out of the workshop's eighteen papers were about capturing, analysing and modelling requirements. Doubt was expressed that this reflected the real priorities of the industry since in industry (or at least some sectors of the industry) these just weren't considered problems. Testing against requirements, by contrast was posited as an example of a more valuable line of research. In response, it was acknowledged that the community could do more about identifying real problems, but we also needed to be innovative rather

than responding only to currently recognised problems. Modelling was offered as an example. Despite industry's initial failure to recognise its utility, modelling has been proven to hold great general value for understanding.

It was also noted that industry is heterogeneous and the problems they face are similarly heterogeneous. Limiting the foci for RE research would potentially impede our ability to address either the full range of problems faced now, or the so far unanticipated problems to be faced in the future.

There was a more general discussion on why, as a community, we have been unable to convincingly demonstrate the value of RE. To put a more positive spin on this, it was suggested that there was an opportunity for us to prove its economic benefits. One way to do this is by stealth: to focus on the things that are clearly acceptable to industry but which serve the wider purpose of RE; things like doing cost estimation and writing test cases. In addition, we should seek to lower the cost to developers of adopting RE tools. For example, RE tools can be implemented as plug-ins for open development environments such as Eclipse, permitting integration with coding, testing, CM and other tools in the developers' armoury.

Less positively, we concluded that, despite one paper's brave attempt to do so, it was not possible to correlate RE activity with macro-economic performance. To do

so would require the collection of data that we simply don't have at this time.

Fresh problems for RE research

A number of papers explored how the emergence of new solution technologies are challenging orthodox views of RE. In particular, service-oriented systems promote a model in which RE is performed directly by end users, since, in principle, service-based applications are configured from a marketplace of services. Even more radically, it was proposed that autonomic systems, able to dynamically adapt to changes in their environment, do their own RE. While it is engineers that set the parameters for adaptation, it is the systems themselves that decide how to adapt at run time. These ideas led to much debate about the nature and probable evolutionary path of RE. As an interim conclusion we agreed that a conventional description of RE as "deciding what the system should do" is inadequate in all these contexts in which RE is performed by unconventional actors.

Two days of talking had left everyone ready for a refuelling break before either flying home, or the start of CAiSE the following day. Before leaving, however, Erik Kamsties, the retiring co-chair was warmly thanked for all his hard work and leadership over the last three years.

© 2005 Erik Kamsties, Pete Sawyer &
Vincenzo Gervasi

RE-Papers

Does Object-Oriented Domain Analysis Work?

Shamal Faily, LogicaCMG Space & Defence Division, shamal.faily@logicaacmg.com

The Rational Unified Process (RUP) [5], the Rapid Object-Oriented Process for Embedded Systems (ROPES) [2] and other OO approaches propose Use Case driven analysis as a mechanism for capturing requirements and deriving object models.

B. Douglass' *Real Time UML* [3] describes how "later analysis decomposes the system into ... objects". Beyond discussion of a number of common object identification strategies, such as noun-phrase underlining and key concept identification, only lip-service is paid to what has been described as the Fundamental Difficulty (FD) of Object Oriented Domain Analysis (OODA).

The Fundamental Difficulty was defined by Svetinovic et al (at RE'05 in Paris, see the reports in this issue) [7] as the difficulty of identifying system domain concepts as Objects (which some might think pretty central to the OODA enterprise - Ed.). A study of undergraduate projects, documented by Svetinovic using these techniques, found that object models of the same system often differed drastically in terms of

concepts identified, while software concepts were often specified at inconsistent abstraction levels.

These observations raise the concern that OODA may be incorrectly applied by many practitioners. Microsoft's Steve McConnell [6] argues that most practitioners neither have the benefit of a Software Engineering education, nor do they have ready access to evaluations of the myriad of available tools and techniques.

Managing the abstraction problems

Software Engineers will appreciate the difficulty of working at multiple levels of abstraction whilst evaluating or developing Object Oriented software, even for solutions within a well-known problem domain. Could the results presented by Svetinovic indicate cognitive overload, caused by the combination of trying to understand the problem space and conceptualising this within the object metaphor?

A cost-effective remedy might be to focus on key Object-Oriented concepts, in practice, using an OO Programming Language such as Python or Smalltalk. Many people, myself included, commonly use Python to prototype concepts and ideas which can be eventually implemented in other languages, such as C++.

By using OO Programming to test one's own ideas and explore how idioms and patterns can work

together, one can appreciate some of the beauty and elegance of solving problems with the object metaphor. Students and practitioners alike should then be more confident and better prepared to abstract the problem space more accurately.

A Reality Check

If transforming domain concepts to objects were purely a visualisation task, a good argument would still exist for OODA. Unfortunately, the more one looks at the grey area between the problem and solution spaces, the less good the argument seems.

For example, Concept Identification by noun-phrase analysis, using scenarios, depends on the quality of the scenario text. But even if the scenarios were initially well-written, what may they be like after repeated and possibly biased editing?

Tool-support can help; promising approaches such as Metamorphosis [1] are likely to evolve to support OODA.

But Open-Source tool support in this area is scarce; COTS tool support is costly; and research approaches have had only a limited outing within industry. Worse, the combination of domain and engineering knowledge necessary to make OODA work remains scarce.

Since few people in industry will want to volunteer their own projects as a proving ground even for suitably risk-mitigated OODA approaches, we may have to accept that progress will be slow.

A way forward?

Fortunately, there are other ways to explore the problem space besides OODA, including traditional approaches like context analysis, and modern ones like goal modelling.

Also, because of the increasing ubiquity and maturity of OO and its applications within the solution space, Software Engineers are well positioned to apply Objects in the problem space.

Hawthorne [4] has suggested that continuing trends such as the move of software development off-shore to the lowest-cost location of the day and growing reliance on COTS/Open Source components will force

Software Engineers to apply their architectural expertise within problem-space areas, ie in the domain of the Requirements Engineer.

We should not necessarily abandon OODA; many industrial projects have used it successfully. But perhaps we should drop OODA for now, until we can better mitigate the concept identification risk?

And, what about a head-to-head comparison between OODA and other approaches? Such a study could help determine where the Fundamental Difficulty lies.

In conclusion

I hope that this short discussion of OODA will provoke reflection on the value of applying the object metaphor within Requirements Engineering.

At the least, OODA needs a health-warning on its cellophane wrapper.

References

- [1] Diaz, I., Pastor, O., Matteo, A., *Modeling Interactions using Role-Driven Patterns*, Proceedings of RE'05, Paris, IEEE
- [2] Douglass, B., *Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns*, 1999, Addison-Wesley
- [3] Douglass, B., *Real Time UML*, 2004, Addison-Wesley
- [4] Hawthorne, M. J., Perry, D. E., *Software Engineering Education in the Era of Outsourcing, Distributed Development, and Open Source Software: Challenges and Opportunities*, Proceedings of ICSE 2005, ACM
- [5] Kruchten, P., *The Rational Unified Process*, 1999, Addison-Wesley
- [6] McConnell, S., *Code Complete*, 2004, Microsoft Press
- [7] Svetinovic, D., Berry, D. M., Godfrey, M., *Concept Identification in Object-Oriented Domain Analysis: Why Some Students Just Don't Get It*, Proceedings of RE'05, Paris, IEEE

© 2005 Shamal Faily

RE-verberations

This section is for items of news that have a bearing on requirements work. RQ would like to hear of such things from its readers.

Tell Me About Your Mother

The May 2005 issue of Scientific American (www.sciam.com) contained an Insights column 'When Medicine Meets Literature'. The summary runs:

"Writing and humanities studies produce better physicians, Rita Charon argues,

because doctors learn to coax hidden information from patients' complaints"

This sounded interesting – a professor of medicine with a PhD in English! When interviewed, she explained:

The goal is to make doctors more empathetic by getting them to articulate and deal with what they feel and to develop sophisticated listening skills, ears for the revelations hidden in imagery and subtext.



Dr Rita Charon

Charon herself named the field 'narrative medicine': quite a different use for story-telling than scenarios in engineering:

For instance, your doctor might ask 'How long have you had shortness of breath?' You say 'Since I divorced my husband.' The next question typically might be 'How long ago was that?' [Followed, presumably, by reaching for the prescription pad.] In contrast, a Rita Charon would then say, 'Tell me about that relationship.'

I'm reminded of the wonderful old story of the man with incurable eczema. He had been to one skin-doctor after another, and had tried every cream and potion known to science. Eventually a dermatologist thought to ask 'When does the eczema come on?' and the man confessed that it was when he dressed up in his wife's

clothes, terrified he'd still be in her finery when she came home. A prescription for an alarm clock permanently cut the drugs bill to zero.....

Leaving aside the pleasingly radical overtones of all this, there are some practical implications for requirements work.

Firstly, listen, listen, listen. The stakeholder isn't you.

Secondly, don't assume you have the answer (a technological fix, the database you want to install, the prescription pad) ready and waiting.

Thirdly, don't assume that things said are out of scope. You don't know the scope of the problem until you've heard the stakeholders! Requirements scope is much less well-defined than medical scope: the cardiologist listening to a patient talking about chest pains and shortness of breath is 'supposed' to deal only with heart problems, isn't she? Well, no. She's supposed to be there to help the patient, and you're there to help the stakeholders, not to sell databases or whatever.

Fourthly, reality is often much stranger than you expect.

Finally, if a stakeholder keeps on coming back to a problem (and tells it to you six times in half an hour, in slightly differing ways), guess what? It's probably the problem he or she is worried about. Even if it isn't what you were hoping to hear.

© Ian Alexander 2005

RE-flections

Nuclear Physics for Cleaning Ladies

Once upon a time in an earlier millennium, at the end of a long day of hard work, a fearfully clever young man was putting the finishing touches to his PhD in nuclear physics in Sir Ernest Rutherford's famed Cavendish Laboratory in Cambridge. Most of the dons had left, but the young man toiled on, eager to get some ideas in order on paper.

The cleaning lady, for in those days it was taken for granted that privileged young men did intellectual things, while poor uneducated women cleaned their laboratories, came into the room and saw him still at work. Curious, for she had an inquiring mind, she came up and said,

Excuse me, I was just wanting to ask you a question.

The young physicist looked up, dizzy with equations:

Yes? Ask away.

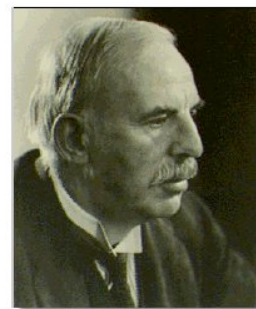
Well, she said, I was wondering if you could explain to me what it is that you do here.

Gosh, said the young physicist. It's really rather complicated.

And that would have been the end of the conversation. But, unnoticed by either the researcher or the cleaning lady, someone else was in the room and listening intently to the unusual conversation.

Young Man, if you can't explain it to her, you can't explain it to anyone.

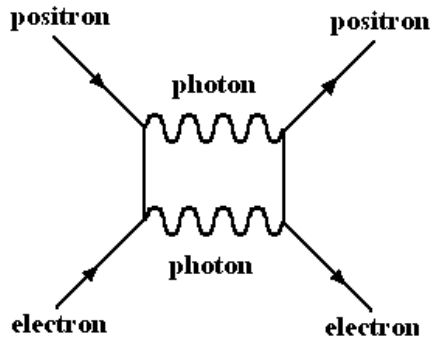
boomed the voice of the head of the department, Professor Rutherford himself.



Sir Ernest Rutherford, 1871-1937

And, one might add, you probably don't understand it yourself. There is a long and honourable tradition among physicists of explaining the most complicated things in terms that everyone can grasp. Take Maxwell's Demon; Einstein's Tram; Schrödinger's

Cat; Feynman's Diagrams, or Hawking's *Brief History of Time*, to name just a few.



A Feynman Diagram, showing in easily-understood form a complex set of related interactions of subatomic particles

Rutherford himself explained his astonishment at his discovery of the nuclear structure of the atom like this:

It was almost as incredible as if you fired a fifteen-inch shell at a piece of tissue paper and it came back and hit you.

He had (in 1909) fired alpha-particles at a thin metal foil, and found that some of them were deflected through more than 90 degrees. They had passed very close to a heavy, highly charged nucleus, and had been strongly repelled. Or at least, if you can think of a better explanation than Rutherford's for what happened, now's your chance to speak up.

Nobody really famous in physics seems to be above simple explanations. Or perhaps it's the case that only

really good physicists understand things well enough to put them simply.

Have you ever tried explaining your really rather complicated system requirements to your admin or cleaning staff? Yes, it's quite hard at first. But try it, stay with it – you can use any kind of text or diagram. You may find things get a bit clearer. After all, your requirements probably aren't harder than nuclear physics. And of course, once you've succeeded in saying what you want so everyone gets it, you may then want to rewrite your specifications before someone seriously misunderstands them.

© Ian Alexander 2005

Special Guest Proverb

After all that physics, this issue's proverb has to come from a physicist, and the most famous of them all is the author of the one equation that everyone knows.

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

Albert Einstein, quoted by Fritjof Capra in *The Tao of Physics*, 1975 (4th edition, 2000)

RQ hesitates to rephrase this as 'As far as requirements refer to reality, they are not precise; and as far as they are precise, they do not refer to reality', but you may like to draw your own conclusions.

RE-Creations

Send contributions to Ian Alexander (ian @ scenarioplus.org .uk). Submissions must be in electronic form, preferably as plain ASCII text or rtf. Deadline for next issue: 15th December 2005

RE-Implementations

RETH: a Freeware Tool

Hermann Kaindl developed a prototype RE tool, RETH (Requirements Engineering Through Hypertext) to explore how requirements could be handled and traced in the familiar manner of a hypertext. The prototype is available free, subject to signing an agreement with his former employers, Siemens.

The easiest way to begin is from a blank window, selecting the Guide, and pressing the New button. (That was after trying to load a Word "parent document" and discovering that only .RTH documents would do.)

The tool offers a choice of English and German. It then automatically creates a document structure, displays two window-panes, and puts the document structure (Domain Model, Goals, Requirements) on the

left in an explorer view. On the right are the contents of the selected part of the document.

The Guide then prompts one to step through a sequence of instructions. These are quite technical, with classes and instances, scenarios and goals, associations and tuples. (I hadn't done tuples since the days of Prolog and Hope.) However the instruction sequence is actually quite easy to follow, especially as the one you are meant to do is highlighted in white while the others are greyed out; and there's even a 'Perform Instruction' button at the bottom that does most of the step for you!

I created a scenario instance and copy-and-pasted the scenario text in from Word without trouble. Then I was prompted to link this to my Goals which didn't exist yet. I was prompted to display and then cancel a window (a blind alley) but the guide, noticing I was stuck, asked if I wanted to create a Goal. That led to

another Guide sequence and I suddenly found I could create goals quickly and easily.

The approach, the direction of working was unfamiliar but I could see what the point was, and was quite rapidly starting to understand how RETH would work.

I then guessed that Requirements too would be created as Instances of the (sub)classes Functional Requirement and Quality Requirement. So I tried it and it worked: there is a simple consistent paradigm of Class - Instance - Instance title - Instance Text (and other Attributes like Further Explanation, Source, Priority, Reasons, and it looks as if you can define your own, too).

Now equipped with some goals (Portable, Reliable, etc) and a Scenario I was ready to try out the Hypertext part, the H in RETH.

I followed the Guide through the task of connecting the scenario to the goals. This consisted of creating "Instance Connections" (links to you or me) between the "Achieves Goals" section (looking much like one of the attributes) of my scenario to several of my goals. Or in RETH terms, between a scenario instance and several goal instances.

I then spotted that there was also a "Constrained by Quality Requirements" section, and guessing that this would be easier if I had some, copied and pasted a few from my Word document as, wait for it, instances of the "Constraints on the System to be Built" class. The alternative was the snappily-titled "Constraints on the Development and Maintenance Process" class, revealing a clear and logical structure for the non-functional requirements, by the way.

At this point I felt it a little clunky navigating to the Create menu to add requirements, and was pleased to find that Control-I would do the job. I added requirements (instances) with names like Size, Weight, Independence of Mains Power. It is quite nice having names for all the requirements but it is certainly a slight additional burden.

I then clicked on my new constraint "Size" and found to my surprise that its text was inherited from that of my Scenario, which was of course in a separate subtree. I didn't expect inheritance to go sideways. However editing the text worked normally and I was able to paste in the shall-statement from Word as desired.

At this point I discovered that I could simply double-click on "Statement" and then paste in my text. On display in the right-hand pane is a button showing "+" if detail is concealed, "-" if all is revealed; the type of attribute, in this case "Text" shown diagonally in grey, the attribute name "Statement", and the status, "new", "inherited" or "overwritten", as well as the actual text itself.

If you want a comparison, the feeling is something like working in Outline mode in Word, something that I notice not many people like to do, with the paragraph style displayed.

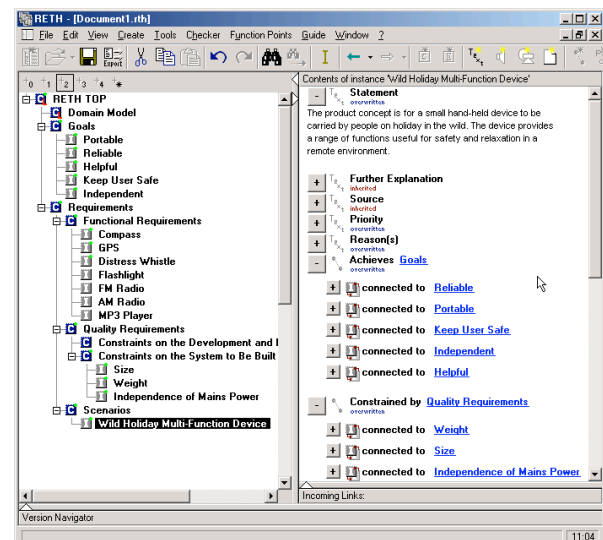
I moved to the functions and quickly now created and pasted in several of them. It's quite logical: to add a function you select "Functional Requirements" and give an Insert command (like Control-I). This feels very unlike the familiar carriage-return to add a paragraph in a Word document, and of course has to be done from the equivalent of a chapter heading, not from a normal paragraph. You can't insert several requirements in rapid succession because each one has to be named, so you have to insert, name, and then insert again. I found myself putting in names of features, and omitting the text with the associated shall-statements for the time being: not necessarily a bad thing.

I then tried filling in the "Priority" attribute. This was free text where I'd have expected a list of values like "Essential" and "Desirable", but you can't have everything in freeware. However RETH does support aggregation as well as association, so the clever can set up a hierarchy containing a list of values if they want to (it's something like linking to values in a separate table).

RETH seems to do quite a lot of checking behind the scenes: a popup window warned me there was no text in some of my functional requirements. However with a fully-linked hypertext, it was time to try navigating.

Within RETH, one can see and simply click on blue underlined links between scenarios, goals and requirements in any direction.

Hypertext, in fact, suddenly emerges as a core part of RETH, and a wholly natural element in requirements work, even in what is effectively a single but very well-structured document.



Exporting to HTML (one of several options) took only a moment. The tool wisely prompts for a folder to export to (though without a browse button), and creates a new one on demand. My RETH document became 37 files including GIF images for embedded icons. Each Scenario, Goal, and Requirement gets its own HTML page: a simple choice, but one that does make seeing the big picture more difficult. The title page is

effectively a hierarchical table of contents, offering a basic overview. However the subsection pages such as "Constraints on the System to Be Built" do not contain links to their contents (eg Size, Weight) which would have been nice.

RETH is a remarkable software package. It does a lot that many expensive tools do, and several things that few commercial tools even attempt (such as handling both goals and scenarios as fully-valued requirement items). The language used is quite technical, assuming that users are software engineers familiar with object

jargon. But it works extremely well, and the learning curve only takes an hour or so to climb.

There do not seem to be any plans to commercialise RETH just yet, so it will probably remain as a freeware prototype for now.

Contact: RETH@siemens.at

© Ian Alexander 2005

RE-Publications

DVD Review: Requirements: Best Practices

A series of lectures on requirements from Compliance Automation: <http://www.complianceautomation.com>

I looked at "Rules for Good Requirements", the second of five 15-minute lectures in the second of three modules. Ivy Hooks talks over bulleted slides with a background of, variously, swirling coloured ripples, rotating transparent globes, and meshing cogwheels. Various comments like "vague and ambiguous" swish past lightly in a delicate typeface to emphasize the voice-over. All the while, very soft rock music pounds away in the background. To give an automobile analogy, it was more of an upholstered Sports Utility Vehicle than a rugged Humvee.

The content is essentially traditional list-of-shall-statement requirements work: "shall", "will", and "should" are all considered correct and indeed necessary for different purposes, but "must" is prohibited. "Should" is said to mean a "goal", a non-mandatory provision.

Ambiguous terms to be avoided include "etc.", "including, but not limited to", "robust", "minimize", "easy", "ultra-low power". They can't be verified. As in Hooks' book (see below), the commentary can be witty: "The system shall support the operator" – how much does the operator weigh, then?

"And" and "Or" are not forbidden, but "And / Or" certainly is. "Be capable of" is mostly just noise; if it has "some subtle meaning" then we ought to avoid that: "we don't want to be subtle in requirements". It sounds much better with a Texan drawl, by the way.

More interesting is the implementation requirement "the aircraft shall have 3 engines". What performance was really wanted here? Asking why revealed "the aircraft shall be able to fly with an engine out". That's the "magic of why".

The lecture was easy to follow, clear, practical, based on experience, and would certainly result in better-written requirements.

Other lectures cover Scope, Operational Concepts, Rationale, Prioritisation, Requirement Attributes, Finding and Fixing Defects, and Managing Requirements Change.

The full set is available on 3 DVDs, compressed from a face-to-face seminar into "less than four hours" of viewing time. It's also available in web-based form for company intranet use, but that's considerably more expensive.

© Ian Alexander 2005

For some reason, RQ did not review Ivy Hooks' excellent book back in 2001. For readers who haven't already seen it, the occasion of the launch of the DVD provides a more than adequate excuse to review it now.

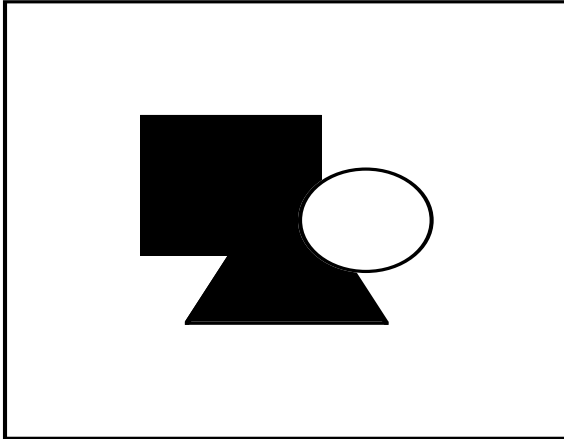
Book Review: Customer-Centered Products

Creating successful products through smart Requirements Management

Ivy F Hooks and Kristin A Farry,
Amacom, 2001

Well, there aren't many requirements books advertised on their covers with the words 'Drawing on their 50 combined years of real-world product development experience ...', so this one must be special. Ivy Hooks is one of the world's most experienced requirements engineers, with a distinguished career with NASA. She now runs Compliance Automation. Kristin Farry is 'an engineer and pilot... and a space shuttle flight controller'. Together they have written 'A Book on Requirements Especially for Managers'.

The book is physically unusual - a hardback, printed on slightly crinkly porous paper in rather big elegant type, with almost an Art Nouveau feeling to the section headings. And it is illustrated with absolutely no UML and virtually no other diagrams - barring one waterfall model which, apart from the nine steps to requirements, is dismissed in a line and a footnote on other development models. The illustrations are cartoons, one or two graphs of the cost of putting off fixing the requirements, and some useful checklist-style tables. In short, this is a practical, lively, sharp and businesslike book, more like a 'Fix the Problem Before Your Competitor Does' exhortation to business managers from the airport bookshop than a discourse on the merits of sequence diagrams over flowcharts.



The authors' backgrounds give them one huge advantage – they have wonderful stories to tell. The cost growth graph speaks volumes about the dire lack of requirements to keep project budgets and timescales under control back in the 1980's. Here is firm evidence that a stitch in time saves nine: compare the well-run and modestly-overrunning Ulysses which invested ten percent of the effort 'pre-design', i.e. in requirements, with the 160% cost-overrunning TDRSS with only five percent invested. The graph also shows that after about 12 or 15% the returns from additional specification effort diminish to almost nothing – requirements are best done quickly but not too quickly.

The book is very readably organized, with chapters on, for instance,

- Creating a Shared Vision (on project scope),
- One Day in the Life of a Product (on operational concepts/scenarios),
- Theirs But to Reason Why (on recording requirements rationale),
- But Will it Work (on thinking ahead to verification).

Story-telling (shades of Schank) is in a way the theme of the book. It extends both to a firm commitment to writing operational scenarios (use cases, stories, does it matter) before writing [system] requirements; and to a stack of grey-boxed anecdotes, humorous and presumably true stories about everything under the sun. The military and the space agencies are of course a rich vein of tales that would just be funny if they were not so sadly wasteful. Daily life is also used to teach the basic principles – Hooks and Farry are evidently lively public speakers. For instance, Farry discusses the topic of restroom doors. Men never see any problem:

"They just walk in the stall, close and latch the door behind them, hang their jacket on the conveniently placed hook above the latch, and use the toilet.

The scene is very different in the women's restroom!"

Aha, a woman's viewpoint on requirements engineering coming up...

"Women then hang coats and purses – both heavier than a man's sports coat – on the hook on the back of the door, because there is no other place besides the dirty, wet floor to put them. If a woman is blessed with small children, the diaper bag and children's survival gear also go on that hook. Is it any wonder that the hinges soon sag? To make matters worse, while Mom is supervising child one through toiletry 101, child two and child three are running about, trying to push open every door and swinging on those that they succeed in opening."

Now that's what I call a vivid operational scenario. Obviously 'moving the hook off of the door and occasionally [providing] a shelf for bags' is a good idea, but it takes someone who actually uses the woman's loo to know it. Farry gives an object lesson in requirements elicitation in inimitable style.

Trendy requirements engineers may not agree with everything in this book.

"Operational concepts are .. important.. but they are not the requirements. For example 'The operator shall be able to turn the machine on and off.'

This description of .. what the operator will do .. is not a requirement on the system. The real system requirement is 'The system shall provide a manual on/off switch.'"

But is it? A manual switch is a choice, which might depend on several operational factors, such as whether the operator's hands are free: a voice-operated switch might be needed. And does a microswitch that actuates a relay count as a manual switch? Perhaps the 'real' requirement is 'The system shall enable the operator to turn the machine on and off.' – which is pretty close to the requirement that the authors criticise. A modern formulation à la Use Case might simply be 'The operator turns the machine on' – which Hooks and Farry would certainly agree was an operational description – and the requirement would be that this capability was mandatory in Version 1.0 of the system. And the magnificent Space Station requirement

'The monkeys shall not bite the astronauts'

(I told you the examples were brilliant) is definitely not a system requirement as framed.

However, the authors are not speaking to the trendy end of the market. They are addressing hierarchical big-business management, encouraging them to take requirements seriously in language they can understand and laugh along with, and they are not far from being the first to do so. Hooks and Farry have managed to communicate exotic-seeming concepts like user

involvement, viewpoints engineering and scenarios to an audience that is strongly oriented towards command-and-control. They have achieved this through a combination of experience, skill at their jobs and at teaching, and not least by having worked in institutions at the very heart of the military-industrial complex – and yet having managed to retain a clear vision.

You wouldn't expect an academic bibliography in this book and there isn't one. Instead there is a short list of books that a business manager could reasonably get hold of for each chapter, and a helpful index. There's no glossary, but the key terms should be pretty clear from the lively discussion and examples.

This book can wholeheartedly be recommended to its intended audience – 'a product developer or customer, a current leader or aspiring manager' as Kathryn Sullivan (astronaut, ret'd) puts it in the preface. It may be easy to assimilate, but it packs quite a punch. It tells the story as it is, and it speaks in plain language. Every manager new to requirements should read it on their next flight. Mind you, developers could do well to have a look at it too.

© Ian Alexander 2001

Book Review: Just Enough Requirements Management

Where Software Development Meets Marketing

by Alan M. Davis, Dorset House, 2005

Since the review of Alan Davis' (excellent) *201 Principles* in 1995, a great many requirements books have appeared (and countless papers, admirably listed on Davis' website). Is there still something new to say?

Happily, there seems to be. Everybody's take on requirements is different: there are requirements books all about software techniques, writing styles, UML, use cases, teamwork, planning, prototyping, and programming (to name just a few). This book does something new: it stresses the role of prioritisation, or Triage as Davis calls it.

Triage means dividing casualties into three groups:

1. those that'll certainly die;
2. those that'll certainly get better; and
3. those that need treatment more or less urgently to survive.

This cheerful word from military/disaster medicine is adopted by Davis because, he says, it perfectly suits the requirements situation. Some things are no-brainers; there's no need for discussion. Some are hopelessly dreamy: they should instantly be dismissed. And the rest need to be ranked in a practical way so you can plan your projects.

Triage / prioritisation seems to overlap quite strongly with the whole business of optioneering: you can't

really tell how important a requirement is until you know a possible solution; and if different options enable different requirements, the trade-offs are complex and probably cannot be optimised perfectly. In any case, there isn't time: if you go on until the requirements are perfect, you'll be bankrupt, and in any case the world will have moved on and the customers will have changed their minds. Davis doesn't use language like 'optioneering' and 'trade-off' but he's certainly at home in a world where imperfect, non-mathematical kludges are necessary to get a quick fix: in fact, it's the only kind of fix in town.

One thing that is specially nice about the book is that it does not just trot out Standish's 1994 Chaos report as evidence that requirements matter, but quotes studies going back to Endres in 1975, and Boehm, Daly, and Fagan in 1976, to show that nearly half of all errors stem from requirements issues. Davis divides these into three categories:

- Knowledge errors
- Triage errors
- Specification errors.

These correspond to Davis' view of the requirements process: Elicitation, Triage, and Specification (or if you prefer, Requirements Gathering, Prioritisation, and Writing/Formalising).

The book concentrates on the awkward middle ground in other dimensions also (apart from triage):

- between doing too much and too little (seeking the perfect spec, or rushing into code);
- between technology and the market (doing analysis, or thinking about what will sell).

This is brave, novel, and very welcome. No-one else, perhaps, could take a long view of the passionate arguments between traditionalists, formalists, and agile methods people, or of the differing viewpoints of developers, managers, and marketing on a software product's requirements, and the difficult interdisciplinary work (involving messy things like schedule risk and marketing strategies, rather than the merits of individual requirements) that has to be done to arrive at good decisions in that political minefield.

The book is addressed to software developers, who are assumed to be doing the requirements work. Obviously that isn't the only possibility. It's also not clear whether the book is intended for, or suitable for people who are more or less beginners. The advice given is generally spot on, but may be hard for the inexperienced to take.

For instance, it's all very well to advise people not to expect complete requirements (page 157), but try telling that to a naive client with a critical project. 'Just enough' is in its way as much a counsel of perfection as the old and obviously impossible 'complete, correct, consistent' dogma (p129ff, for which Davis was at least partly responsible). But is

'just enough' any easier? How do you know that you have the bare minimum? It might be safer to do just a little more to provide a margin for error! Of course, it depends who (developer, marketer, customer) is talking.

So, perhaps this is a book for experienced requirements people (ie, those who've been burnt once) to reflect on. It doesn't attempt to cover every analysis and elicitation method ever taught: the textbooks can do that. Instead, it takes a light, informed, politically-skilful and industrially-informed look at the problem of doing just enough. This is very timely, given the 'heavy RE' versus 'agile methods' debate: and Davis succeeds in

pointing out where the balance lies. Davis writes in a fresh and engaging way, telling stories from his long and varied experience as a consultant (and researcher).

Davis has come up with yet another good, practical book for industry. It's also packed with knowledge of how the requirements literature fits together, forming a useful synopsis of the last thirty years of research, which might be useful to academics starting out in the field. It is very definitely about software, and within that area it is remarkably comprehensive for such a short book; but much of the approach is applicable to requirements work in other areas too.

© Ian Alexander 2005

RE-Spones

RQ welcomes comments and reactions to articles and reports published in its pages.

RE-Sources

Books, Papers

The RQ archive at the RESG website:

<http://www.resg.org.uk>

Al Davis' bibliography of requirements papers:

<http://www.uccs.edu/~adavis/reqbib.htm>

Ian Alexander's archive of requirements book reviews:

<http://easyweb.easynet.co.uk/~iany/reviews/reviews.htm>

Scenario Plus – free tools and templates:

<http://www.scenarioplus.org.uk>

CREWS web site:

<http://sunsite.informatik.rwth-aachen.de/CREWS/>

Requirements Engineering, Student Newsletter:

www.cc.gatech.edu/computing/SW_Eng/resnews.html

IFIP Working Group 2.9 (Software RE):

http://www.cis.gsu.edu/~wrobinso/ifip2_9/

Requirements Engineering Journal (REJ):

<http://rej.co.umist.ac.uk/>

RE resource centre at UTS (Australia):

<http://research.it.uts.edu.au/re/>

Volere template:

<http://www.volere.co.uk>

DACS Gold Practices "Manage Requirements":

<http://www.goldpractices.com/practices/mr/index.php>

Mailing lists

RE-online:

<http://www-staff.it.uts.edu.au/~didar/RE-online.html>

The RE-online mailing list acts as a forum for requirements engineering researchers and practitioners. To subscribe to RE-online mailing list, send e-mail to majordomo@it.uts.edu.au with the following as the first and only line in the body of the message:

subscribe RE-online <your email address>

LINKAlert:

<http://link.springer.de/alert>

A free mailing service for the table of contents of the *International Journal on Software Tools for Technology Transfer*.

RE-Actors: the committee of the RESG

Patron:

Prof. Michael Jackson, Independent Consultant,
[jacksonma @ acm.org](mailto:jacksonma@acm.org)

Chair:

Dr Pete Sawyer, Computing Department,
Lancaster University,
[sawyer @ comp.lancs.ac.uk](mailto:sawyer@comp.lancs.ac.uk)

Vice-Chair:

Dr Kathy Maitland, University of Central England,
[Kathleen.Maitland @ uce.ac.uk](mailto:Kathleen.Maitland@uce.ac.uk)

Treasurer:

Prof. Neil Maiden, Centre for HCI Design, City
University,
[N.A.M.Maiden @ city.ac.uk](mailto:N.A.M.Maiden@city.ac.uk)

Secretary:

David Bush, National Air Traffic Services,
[David.Bush @ nats.co.uk](mailto:David.Bush@nats.co.uk)

Membership secretary:

Dr Lucia Rapanotti, Computing Department, The Open
University,
[lrapanotti @ open.ac.uk](mailto:lrapanotti@open.ac.uk)

Publicity officer:

William Heaven, Department of Computing, Imperial
College,
wjh00@doc.ic.ac.uk

Newsletter editor:

Ian Alexander, Scenario Plus Ltd.,
[ian @ scenarioplus.org .uk](mailto:ian@scenarioplus.org.uk)

Newsletter reporter:

Ljerka Beus-Dukic, University of Westminster,
L.Beus-Dukic@westminster.ac.uk

Regional officer:

Steve Armstrong, Computing Department, The Open
University,
[S.Armstrong @ open.ac.uk](mailto:S.Armstrong@open.ac.uk)

Student Liaison Officers:

Zachos Konstantinos, City University,
kzachos@soi.city.ac.uk

Andrew Stone, Lancaster University,
a.stone1@lancaster.ac.uk

Immediate Past Chair:

Prof. Bashar Nuseibeh, The Open University,
[B.Nuseibeh @ open.ac.uk](mailto:B.Nuseibeh@open.ac.uk)

Industrial liaison:

Prof Wolfgang Emmerich, University College London,
[W.Emmerich @ cs.ucl.ac.uk](mailto:W.Emmerich@cs.ucl.ac.uk)

Suzanne Robertson, Atlantic Systems Guild Ltd.,
[suzanne @ systemsguild.com](mailto:suzanne@systemsguild.com)

Gordon Woods, Independent Consultant,
[Gordon @ cigitech.demon.co.uk](mailto:Gordon@cigitech.demon.co.uk)

Alistair Mavin, Rolls-Royce,
[alistair.mavin @ rolls-royce.co](mailto:alistair.mavin@rolls-royce.co)

Individual Membership Form

1. Membership Type

Please indicate type of membership:

BCS/IEE members, please

BCS / IEE member (£10) []
number _____

indicate membership

Non-BCS / IEE member (£20) []

Full-time student (free) [] *Studying at:* _____

If you need a receipt, please tick here []

Corporate Membership also available – details at www.resg.org.uk or ask the Membership Secretary

Payment by cheque only. Please make it payable to "The BCS Requirements Engineering Specialist Group"

2. Your Details

Title: Mr/Mrs/Ms/Dr/Professor/Other:_____ (delete as appropriate)

First Name: _____ Surname: _____

Address for correspondence: _____

Postcode: _____

Phone: _____ Fax: _____

E-mail address: *Please write your e-mail address clearly using BLOCK CAPITALS*

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	5
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---

As an RESG member your e-mail address will be added to the RESG mailing list

Optionally, indicate:

Your organisation's name: _____

Its type of business/domain:_____

3. Your Specific Interests

Areas of interest for the RESG given at http://www.resg.org.uk/about_us.html. Is there another area would you like us to add?

4. Your Participation Preferences

Please indicate what timings/duration for RESG events would suit you:

Whole day [] Half day [] Evening [] Other (please specify) [] _____

Please indicate whether you would be willing to help the RESG with:

Publicity [] Newsletter contributions [] Organisation of meetings []

5. Your Mail Preferences

The RQ quarterly newsletter is regularly sent to all RESG members as a PDF e-mail attachment. If you would also like a hard-copy of RQ delivered by post, please tick here []

I understand that the information supplied on this form will be held in a database and used for the purpose of distributing information relevant to the activities of the RESG.

6. Your signature: _____ Date: _____

Please send this form with payment (if applicable) to: Dr Lucia Rapanotti, RESG Membership Secretary
membership-RESG@open.ac.uk Fax +44 (0)1908-652140
 Computing Dept., The Open University, Walton Hall, Milton Keynes, MK7 6AA, U.K.