



Requirements Quarterly

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

© 2007 RESG

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

RQ43 (March 2007)

Contents

<i>RE-soundings</i>	1	Habitual Processes	7
From the Editor	1	<i>RE-verberations</i>	9
Chairman's Message	1	Vint Cerf: IP Evangelist?	9
<i>RE-treats</i>	2	<i>RE-flections</i>	9
Evening Pub Talk: 10 Things You Need to Know About Requirements	2	A Simple Profession	9
AGM & Networking Evening with Soapbox Oratory	2	The Carpet Shall Fly at 300 Knots	10
Formal-Lite Requirements Event	2	<i>RE-partee</i>	10
RE Education & Training	2	Proverbs	10
<i>RE-calls</i>	2	RE-definitions	10
Mastering the Requirements Process Part 2	2	Safety Advice	11
Introduction to Requirements	3	Prohibitions	11
WER'07	3	<i>RE-publications</i>	11
RefsQ'07	3	Book Review: Performance-Based Earned Value by Solomon & Young	11
RE'07	3	<i>RE-sources</i>	12
<i>RE-readings</i>	3	Books, Papers	12
Service-Centric RE	3	Mailing lists	12
<i>RE-writings</i>	5	<i>RE-actors: the committee of the RESG</i>	12
Upwards and Onwards	5	<i>Contributing to RQ</i>	12
Creative Thinking	6	<i>Joining the RESG</i>	12

RE-soundings

From the Editor

In this issue, RQ is happy to welcome a consultant, Stephen Nolan, and a researcher, Kos Zachos, to its pages. You can find their articles in the *RE-writings* section. There's an unusual new book: unusual for RQ, as it's about project management; but also unusual in balancing and tying together requirements and management in a carefully-designed and beautifully-implemented package.

We have the usual mix of *RE-verberations* from the news, *RE-flections* on matters professional, and *RE-partee* on matters in the public domain that were not too well thought out.

The only RESG event reported on in this issue is this month's Service-Centric RE seminar, held at City University. If you missed it, read your *RE-treats* more closely next time: it was a state-of-the-art affair, not to be missed. It was typical of RESG events in its mix of

relaxed informality and highly knowledgeable people (both talking and on the receiving end).

Ian Alexander,
Scenario Plus

Chairman's Message

As I write this on the train from Lancaster to Euston, on my way to the RESG service-centric RE event, I gaze out at the swollen rivers and muddy fields so characteristic of the English midlands towards the end of winter. Meanwhile, in my newspaper, there's a striking picture of Worcestershire County Cricket club's ground, under many muddy centimetres of the swollen River Severn.

Flooding, we're told, is on the increase and investment is being made to find ways to mitigate its effects. One of these is better flood warning systems. Traditionally (i.e. about five years ago), a stretch of river or flood

plain would be instrumented with a set of depth and flow rate sensors which would periodically ship their data over a network to a meaty computer that would execute computationally expensive flood prediction models.

Increasingly, however, embedded devices are allowing developers to envision intelligent sensor networks. Instead of dumb sensor nodes, computational grids are composed of sensors with inbuilt processing power capable of executing predictive models on-site. One advantage of this is that the sensor networks can reconfigure themselves as the environment adapts. They can vary sensor reading frequency and CPU speeds according to the urgency implied by the river state. As the water rises and nodes become vulnerable to floating debris, they can vary the trade-off between power consumption and fault-tolerance by adapting the logical network topologies used.

Why should this interest requirements engineers? The point is that as new applications emerge that involve

embedding systems in dynamically changing environments, so we are going to need to be able to analyze, model and reason about requirements for systems to adapt, effectively, from one system to a different one. To date, those dynamically adaptive systems that have been developed and deployed have been very much technology-driven. That is going to have to change if they are to become useful. However, the state-of-the-art is such that the best organizations (but maybe only the best ones) can become pretty good at discovering the requirements for static environments, or at least environments that change slowly enough to permit off-line adaptation of their computer systems. Dynamically changing environments offer a new set of problems altogether.

I reckon the challenge offered by adaptive systems is going to keep all pretty busy for a while to come. It's going to be fun!

*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:

Evening Pub Talk: 10 Things You Need to Know About Requirements

Ian Alexander will give a very informal talk in a room above a central London pub. He'll try to answer some of the really intractable research questions in requirements today, in well under an hour. Why is it so hopeless just to try to write requirements down, when it seems so simple? How can something be simple but not easy? How can you retain some shreds of street cred while talking about goals, scenarios, and traceability? And what is the singular of 'acceptance criteria'?

Plenty of time will be left for questions, and of course for drinks.

6pm – 8pm, A room above a pub, London, Wednesday
16th May 2007

AGM & Networking Evening with Soapbox Oratory

This will be a relaxed opportunity to enjoy some lively

RE-calls

Recent Calls for Papers and Participation

Mastering the Requirements Process Part 2

19-20 April 2007, London, presented by Suzanne Robertson, Atlantic Systems Guild

Good requirements are crucial for good systems. This seminar and workshop is about better requirements. This is an advanced course: it improves the skills of

debating of current RE issues, and to meet other requirements people over a glass of wine and a finger buffet after your day's work. Free to RESG members.

4pm – 7pm, 5th July 2007, University of Westminster, London

Formal-Lite Requirements Event

Morning Tutorial and Afternoon Seminar
September 2007, University of York

Some of the world's leading experts in formal specification will introduce the topic and describe the state-of-the-art in this combined tutorial and seminar.

RE Education & Training

5th December 2007, City University, London

This event looks at the problem of how we should work to improve requirements skills. Education of university students and training of practitioners are the two most powerful interventions we have. Can we do better? Can websites, books, conferences and discussion groups help? University teachers and industry trainers meet to compare notes and discuss new approaches.

experienced business analysts, and teaches how to use the requirements deliverables for project management.

- Choose the best set of requirements
- Identify techniques for quantifying the business value of your requirements investment
- Learn how to anticipate market opportunities
- Understand how to deal with requirements for existing systems

- Learn how to discover the correct stakeholders for your project

www.irmuk.co.uk/58/

Introduction to Requirements

24-25 April 2007, The IET, Savoy Place, London,
presented by Ian Alexander, Scenario Plus

This 2 day course introduces the requirements process, in the context of engineering a system.

The course covers the whole process from launching the project, through discovering the requirements, prioritising, formalising, and validating them.

Throughout the course, participants learn and practise the key techniques such as identifying stakeholders, defining terms and detecting errors, omissions and conflicts. The use of tools to manage requirements is explored, along with requirements reuse.

The course is always a lively mix of explanation and practical exercises to get you familiar with applying effective requirements techniques.

<http://www.theiet.org/courses> for details and bookings.

WER'07

10th Workshop on Requirements Engineering
17th-18th May 2007, Toronto, Canada.

<http://www.cin.ufpe.br/~wer07/>

RefsQ'07

The 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, Trondheim, Norway. 11-12 June 2007

<http://www.refsq.org>

For its thirteenth birthday, REFSQ evolves into a working conference! However, REFSQ will perpetuate its tradition of being a highly structured and interactive forum for researchers and practitioners to address the problem of ensuring software quality through requirements engineering (RE).

RE is as integral to the assurance of software quality now as it was when the first REFSQ took place in 1994. Compared to 1994, our understanding of RE has improved, while newer and better methods and tools are available to practitioners. At the same time, however, new challenges have emerged.

REFSQ'07 seeks reports of innovative work in RE that

enhance the quality of software and systems, particularly where challenged by new development paradigms or technologies. We encourage researchers and practitioners from the RE, software engineering, information systems, and embedded systems fields to present original work. Contributions from cognate areas such as formal methods, systems engineering, economics and management and social sciences are very welcome for the insights they provide in RE.

RE'07

15th IEEE International Requirements Engineering Conference, 15-19 October, 2007, Delhi, India

Understanding Requirements in the Global Economy

As software development is now part of the global economy, requirements engineering is the key bridge between the customer and supplier. Understanding and translating users' needs into effective solutions has always been vital: however, as development is outsourced requirements have to reflect cultures and languages and local needs. Furthermore, understanding requirements becomes a collaborative activity across time and space.

The IEEE International Requirements Engineering conference provides the premier international forum for researchers, educators and industrial practitioners to present and discuss the most recent innovations, trends, experiences and concerns in the field of requirements engineering.

RE07 focuses on the international context for requirements engineering; as off-shoring and outsourcing become increasingly common, issues of culture and localisation become critical. Requirements engineering itself will change as it becomes a 24/7 collaborative activity across national boundaries. Globalisation highlights the problems and will stress-test the solutions that already exist in RE, so particular emphasis will be placed on:

- RE in the global economy
- Collaborative Requirements Engineering
- Requirements, culture and localisation

Submission Deadlines

Notification to authors 27 April 2007

Doctoral symposium, poster and other submissions 11 May 2007

<http://www.re07.org/>

RE-readings

Reviews of recent Requirements Engineering events.

Service-Centric RE

7th March 2007, City University

Neil Maiden welcomed us and introduced the question

of whether Services – a solution approach – are relevant to Requirements – a problem consideration. A service is much like a software component, except that its management is different (more like pay-as-you-go than purchasing). Services raise new challenges for requirements, because you need to match them to

service capabilities. These may execute remotely. A service may fail, so you may need to discover an alternative service automatically at run-time. This dream remains distant: there are today no semantics for expressing shareable specifications, so it is not yet possible to interrogate available services to discover what they offer. In other words, if you want to make use of some unknown services, dynamically, you need to be able to find them without human intervention, but nobody has created a protocol for that.

Olivier Nano, European Microsoft Innovation Centre, Aachen, spoke on The Role of Service Level Agreements (SLA) in Service Requirements Specification.

Nano writes: "Today, more and more applications are delivered through services. Services Oriented Architecture (SOA) is very valuable for loosely coupling different parts of a system. It adds agility to respond to environment evolution, it helps to quickly provides new functionalities. But [for] business viability, it is important to be able to describe the expected business level objectives. A Service Level Agreement (SLA) is a contract that links the service provider and the service consumer on the business level objectives of the service."

An SLA is thus not only about Quality of Service (QoS). It also covers what the service is to deliver, price and penalties, obligations on both sides, and measures of success (MoS). Clearly there are strong requirement-like elements in this. An SLA is a promise, which carries an element of risk: you may not be able to meet the promise.

Unfortunately, there are obstacles. It is hard to predict code performance on different platforms. Policies have to be created by hand. The impact of technology on business goals is not obvious. Factors interact, creating an exponential increase in complexity. These define the need for managing SLAs. There are different views of an SLA – from governance and legal aspects to straight performance – and all need to have data gathered for them at run-time. Audit tags can be inserted to ensure that each gets the information it needs, at the right level.

There is thus an SLA life-cycle much like any development cycle: product/service development, (re)negotiation/sales, implementation, execution, assessment, and decommissioning (ie ending the contract), with continuous feedback. Tools like AmberPoint, and research projects like SeCSE (see below) are trying to help with parts of this process.

Konstantinos Zachos, Neil Maiden, Xiaohong Zhu, Rhydian Howells-Morris and Sara Jones, City University, London, spoke on Discovering Web Services to Specify More Complete System Requirements.

The EU-funded SeCSE Integrated Project (<http://secse.eng.it>) is exploring techniques that enable service discovery during early requirements processes

and exploit discovered services to enhance requirements specifications.

The SeCSE requirements process is an inquiry cycle from use cases to service requests to actual queries (to a registry of services). Service integrators and consumers can then consider the discovered services, modify their requirements, and go round the cycle again.

Use cases are modelled as containing sequences of Actions, each of which may correspond to a Requirement. These can be written within the SeCSE tool environment: they are stored as XML. A future stage may be to import requirements from Excel tables or Word documents.

A discovery engine, EDDiE, parses natural language service requests (extracted from use case text), tags up terms with their sense (eg driver means software driver not a human or a golf club), expands terms into sets of synonyms to raise the chances of finding matching services, and finally a matching engine uses these to search the service registry. Currently there are about 260 services scattered around Europe.

The result is to match actions to candidate services. Matches can be checked for compliance with Non-Functional Requirements (NFRs). Thus SeCSE handles both functional and non-functional needs.

Glen Dobson and Pete Sawyer, Computing Department, Lancaster University, spoke on Specifying Service Qualities: an Ontology-Based Approach. Theirs is yet another SeCSE project.

Services are often black-box, so assessing service quality is key to differentiating services technically and commercially. Quality of Service (QoS) is typically expressed as a bound on some metric, eg availability > 99.99%. QoS isn't restricted to low-level details of networks: it applies to services as wholes, ie as experienced by service users.

Unfortunately, services might express their QoS as an MTBF (Mean Time Between Failures, in hours or weeks of operation) and an MTTR (Mean Time To Repair), leading any naïve approach to fail to match the QoS requirement with the service's QoS specification. The answer to this problem could be to use some kind of formal semantics, ie a model or ontology of some kind. It isn't enough to try straightforward syntax matching.

There is quite a large choice of ontology languages, including OWL (inferences can be True, False, or Unknown) and SWRL, which uses OWL and something like Prolog's Horn Clauses to express rules. A semantic web (ie like the World Wide Web but with meaning) relates concepts as logical facts and rules.

There are about 20 commonly used attributes (reliability, security, accuracy, maintainability, etc) and about 40 common metrics. Clearly, measurement is essential if QoS is going to mean much.

Thus “mean time to complete < 2000 ms” should match a service which offers “< 1.5 seconds”. A check on an ontology that knows about time units will succeed. Similarly, the equation $AVAIL=MTBF/(MTTR+MTBF)$ enables an ontology-based reasoning system to match QoS requirements expressed in availability terms to services expressed in MTTR and MTBF terms, through automated inference.

Some metrics apply both to end-to-end QoS and to single services; others don't.

Interested readers can find out more at <http://www.comp.lancs.ac.uk/owl/qos/qosont2.owl>

David Budgen, Department of Computer Science, University of Durham, spoke on Requirements for an Integration Broker for Heterogeneous Information Sources.

A service broker gathers information from services on the fly, and provides them to the client to support queries and hence access actual data. IBHIS actually itself uses services but these are bound statically: user interface, query service, security service, discovery service, ontology service. Queries consult a semantic registry (see the SeCSE talks above for how this kind of thing works) which gives access to numerous data access services (maintained by different owners).

To try this out, the project created 6 healthcare use cases from the work of a single health authority (Solihull). These ranged from traditional history follow-up by the hospital doctor and the GP exploring a cancer case, to the difficulty of investigating scattered facts (say, a remote A&E department's records, a health visitor's observations) possibly relevant to a suspected child abuse case. In the former, the problem is that many records from a long history may together form a story that only a specialist can use to make a diagnosis. In the latter, the challenge is that 2 or more 'amber' facts, intentionally separated in time and space for nefarious reasons, may together make a 'red' alert. The IBHIS broker puts the facts together in something like 'semantic data mining' to add value to the information.

We in the audience came to realise that there may well be something big under research and development here with this service thing. It hasn't hit the big time yet, but perhaps we're on the verge of a revolution. Perhaps, one day soon, requirements will be translated – dynamically – into the immediate provision of software services.

© Ian Alexander 2007

***RE*-writings**

Upwards and Onwards

Maintaining the Client's Vision through Requirements Management

By Stephen Nolan

Many businesses remain blissfully unaware of 'requirements' as a discipline in its own right. Like Dr Johnson's opinion of the performing dog: I am often not so much surprised that requirements are done badly, as that they are done at all.

In the early phases of software development projects, enthusiastic project managers often announce that they have 'done the requirements' and a weighty requirements document is then produced. This usually amounts to little more than a senior management-sponsored brain-dump of everything IT-related that they have been meaning to do for several years but have had neither the budget nor the political will to deliver.

Of course, as a consultant, I wouldn't presume to know a client's business better than they know it themselves. So why is it that they should know how to 'do requirements'? Well, it is presumably to demonstrate to an IT consultant that they know how software development works, or else to shave a few weeks off the project duration to save some budget. In my experience neither really helps the project and neither generally come to pass. Poorly-written requirements

take time and effort to rescue, and DIY requirements often have the same consequences as domestic DIY efforts – shave a few millimetres off a table leg here, a few there, and you destabilise the whole undertaking. It is best left to the professionals.

So what is the underlying problem? I believe it is all to do with timing. Clients often leap-in at what they understand to be the start of a project with little critical questioning of either rationale or scope. They are solutionising - looking at perceived problems and designing solutions they think will fit. But tinkering around with your table legs is often a sign that there are bigger problems - maybe it's the floor that's uneven, maybe its woodworm. Thus it is the same with requirements. Business problems require business solutions that may or may not require a specific technical solution. The question we really should be asking is not “how can we solve this problem” but “what is the real problem”? What exactly is it that the business trying to achieve?

Defining the problem space may not be thought of as requirements engineering in the conventional sense. Engineering a solution, after all, demands that the problem has already been addressed. However, we have a responsibility to add value. One area where those engaged in requirements activities should certainly be adding value is in helping the business to define its own problem spaces. This allows the business to understand the shape and fit of solutions which occupy those spaces.

Traditionally, strategy is the domain of the management consultants, and sometimes they can be very effective. However, at their worst, their output can be vague, poorly-qualified rhetoric that would give a skilled analyst sleepless nights. And of course strategy consultants are rarely responsible for delivery. This separation of strategy from the projects that are spawned by that strategy is divisive because it leads to projects (and particularly IT projects) being undertaken in isolation from 'the business'.

Requirements managers are, however, naturally strategic and understand how projects contribute to the larger strategic aims, or goals, of the business. The problem is that we are usually engaged at a point where, at best, the strategy has already been set; and at worst, there is an absence of any strategy at all. We are brought in to 'do the requirements'. However, our starting point should be to either align the project with a stated strategy or indeed to help define that strategy in the first place, and thus ensure that the business understands why the project exists at all.

We help clients by positioning this work as an exercise in goal definition – the acknowledged realm of the requirements manager. Helping clients understand their goals has several advantages: firstly, it encourages clients to assess the project's fit within their wider business strategy, which is good for them; and secondly, it enables requirements managers to position themselves at a level within the organisation above the middle managers who have been charged with delivering the project, which is good for us.

After all, we live and breathe this stuff. We have the unique ability to align these early deliverables (project goals) within the context of the wider business. Crucially, we are also responsible for defining what follows, be that solution, software or product development, to actually achieve those goals. By linking requirements to goals, we provide traceability back to the vision. We bridge the gap that often exists between vision and delivery, and thereby provide the continuity that is often lacking in projects.

This simplest of classifications – goals first, requirements later – allows us to position our work as strategic within the business. And strategy is generally 'done', or at least defined, at the senior management level. Gaining senior management buy-in allows the requirements manager to break out of middle management and gain that elusive 'access all areas' pass to roam freely across the business in later phases.

I find this tremendously uplifting in career terms. Requirements analysts and managers frequently originate from within the business analyst community and often struggle to move themselves up the food chain within the business. The perceived wisdom is that requirements managers may one day grow up to be project managers, but this is often far from the preferred career path for requirements analysts.

We must consider that a project manager can successfully deliver a project without ever having to think about 'the vision'. I would argue that requirements analysts should constantly seek to align their work with the strategic direction of the business and strive to be the keeper of this vision. This is what sets us apart and it is an area where we can add real value to our clients' or our own businesses.

© Stephen Nolan 2007

Creative Thinking

Can you help me widen my horizon?

by Konstantinos Zachos

"Creative thinking requires an attitude that allows you to search for ideas and manipulate your knowledge and experience. With this outlook, you try various approaches ...often not getting anywhere. You use crazy, foolish, and impractical ideas as stepping stones to practical new ideas. You break the rules occasionally, and explore for ideas in unusual outside places." - Roger von Oech, *A Whack on the Side of the Head*, Warner Books 1990



What does a coffee have in common with a car wash? Well, I've never thought of an association until the other day when I received a flyer advertising a new car wash called 'Car Wash Coffee'. They offer two types of car wash programmes: 'Espresso' and 'Cappuccino'. The former corresponds to the traditional 'small' (exterior) wash programme, whereas the latter corresponds to the traditional 'bigger' (exterior and interior) wash programme.

Can you see now the association made between types of coffee and types of car wash programmes? Here, the company made use of the idea of transferring knowledge from one situation (i.e. consuming coffee) to another one (i.e. washing motor vehicles) on the basis of the judgment that the two situations are essentially identical with respect to the task at hand (i.e. defining different types car wash programmes). In effect, these two situations are analogous to some degree.

Analogy-making is just one of the many creativity techniques found in literature. More importantly

though, it is a brilliant technique mostly neglected in RE. Let me explain...

Most software professionals would agree that creativity is an important part of being able to build software systems. Of course, there might be some discussion about precisely where in the software process this creativity is needed, but there would be little disagreement about whether it is needed. In the old industrial society, copying existing information was good enough or even the norm - originality was not appreciated.

Creativity, when talked of it all, was something for artists and authors aspiring to genius. In this new information society things have changed. One of the reasons is that copying existing information or products today is cheap and easy making original ideas the only results that matter. Another reason is the potential creativity has. M. A. Cusmano [1] has put it nicely by saying that "since human creativity is so vast in potential and computer hardware is still evolving by leaps and bounds, it would be foolish to think of software technology as being mature".

Requirements analysts in recent years have acknowledged the importance of creative thinking in RE and it makes sense, don't you think? Stakeholders and designers work together to create ideas for new systems, then express these ideas as requirements that envision these new ideas.

However, it is one thing to believe that creativity is an essential part of the work of software, and it is another to make creativity happen. For instance, how it is possible to find associations and similarities between two situations like the one described earlier in this article? Assuming you have a situation or problem which is vague and ill-defined - a typical scenario encountered in early requirements processes - how can system analysts make use of creativity techniques like analogy-making to enhance their requirements specification for the system-to-be? Neil Maiden [2] describes results from a creativity workshop ran early in the requirements process exploring the role of analogies in RE. Such studies have shown that people can exploit such analogies to reuse requirements if they are given support to recognise, understand and transfer the analogies.

The research I'm involved in tries to answer the above questions, by making use of web and software services. Web and software services are operations that users access via the internet through a well-defined interface independent of where the service is executed. Imagine having a large database of existing software or web services in various domains. Now imagine you would have a tool which uses analogical reasoning techniques to encourage the creation and invention of new requirements. Rather than rely on expert or your own presentations of domain knowledge, this tool will discover and retrieve services in analogical domains to your current application, to support creative thinking

about requirements for that application. Suddenly you are confronted with domains you have never thought of before when looking at your specification of a car wash system - and guess what, one of the discovered services is the coffee delivery service.

Yes, creativity exists in every individual and awaits only the proper conditions to be released and expressed. Robert L. Glass in his book '*Software Creativity 2.0*' concludes one of his chapters by saying that "system analysts can be creative; designers can be creative... It is not a matter of 'if', but rather 'how', that creativity can be freed and employed." Analogy-making tools are just one way to free and employ creativity. People who focus their attention only in one direction begin to realise, after a while, that their horizons have become narrow and their judgment impaired... Would you like some help in widening YOUR horizon?

© Konstantinos Zachos 2007

[1] Michael A. Cusmano, *The Business of Software*, Free Press 2004

[2] Maiden N., Robertson S. & Gizikis A., 2004, '*Provoking Creativity: Imagine What Your Requirements Could be Like*', IEEE Software, September/October 2004 21(5), 68-75.

Habitual Processes

Towards the western end of the Himalayas, the River Indus runs swiftly through an extraordinarily deep gorge. It is ten thousand feet from top to bottom. The river has cut right through the highest mountain range on earth. How? What can it mean?



The young Indus high in the Himalayas

The answer, according to geomorphologists, is that the Indus must have been flowing across the gentle plains of southern Asia before India joined, or rather collided, with the Asian plate. In that titanic impact, which is still continuing, India is diving under Asia, crumpling that plate and building the mountains in the process.

Meanwhile, the Indus continued to flow, transforming itself from a meandering lowland river into a raging mountain torrent. And while the land around it rose rapidly, it was able to cut down through its bed still faster.

When, finally, the plates have come to rest (or India has entirely vanished), and the mountains have been worn into dust and deposited as silt in the Indian Ocean, the Indus will still be flowing quietly across the Himalayan plains.

The power of habit

The moral of this tale is that we are overly impressed by large structures – such as mountains, and overly neglectful of quiet habitual processes – such as rivers.

Why? Because it is not structure that matters in an organisation, helpful though a good structure may be. It is not dramatic intervention that makes a good company either, however skilled the consultants may be. It is what the people and equipment making up the organisation's systems actually do every day that determines its effectiveness.

For example, aerospace engineers habitually work out very carefully, with traditional shall-statements traced from system to subsystem specifications, exactly how an aircraft is to be constructed and what each part of it is to do. Each requirement and each trace is exhaustively checked for correctness. Some mistakes still get through, but because everyone involved knows how costly such errors can be to fix late in the life-cycle, people use every tool at their disposal – simulation and modelling, traceability, requirements attributes – to minimise the risk of specification errors.

An open-plan office is no place to think

In contrast, people in other industries – perhaps less mature, or perhaps where the risks to profitability and safety are less evident – pay less daily attention to correct specification. There are many distractions. An open-plan office is no place to think. Meetings are called at random; mobile phones go off all day, disturbing 'normal work' and meetings alike. People arrive late or early, take flexitime holidays, get married, fall ill, bury relatives. Full-time workers leave, contractors flood in and out continually.

To cope with the chaos, people work long hours, and get into the habit of staying in the office till late: perhaps this gives their bosses the impression they are working hard. But it also forces them to run their lives from the office. People book dental appointments, buy sofas, browse the web, send personal emails, order theatre tickets, chat with their friends, meet possible spouses, eat, drink, relax, exercise, shower, and dress while in the office. Somehow they quite often manage to give the impression that they are also working.

In such a habitually just-in-time environment, is it likely that people will carefully trace the subsystem requirements to the system, user, and business

requirements, updating each set as necessary to maintain consistency with each change?

Agile Requirements for Chaotic People

Whatever the engineering merits of agility and iteration – earlier detection of mismatches and errors, closer involvement of stakeholders, earlier delivery of the first working version – the truth is that an agile requirements process is the only option for chaotic environments full of chaotic people.

If the attention span of a meeting is 20 minutes (on a quiet day), then that's the maximum length of an engineering presentation – including discussion.

If the habitual evaluation process for buying a house or a car or a washing machine is a quick scan of a couple of websites (ok, one website) followed by a swift electronic transaction, phone call, or text message, then the evaluation process for an engineering decision is going to have to be broadly similar. Otherwise, it will be perceived as a gross anomaly, the only large, slow, pedantic, plodding thing in a whirlwind life.

What are the implications of this for requirements?

One conclusion might be that careful industries need to recruit careful people, hardly a surprise. They will certainly find that increasingly difficult.



Art-Scene Scenario Walkthrough Tool on a PDA
<http://www-hcid.soi.city.ac.uk/research/Artsceneindex.html>

Another conclusion might be that since people like to do things quickly and with gadgets, it may be effective to provide people with requirements gadgets. Scenario tools on PDAs, creativity and idea-capture tools on tablets... the possibilities are endless.

A third conclusion, perhaps, is that requirements processes can only be changed by being quite pragmatic about the kind of effort and attention that people will actually devote to specifying what they want.

The likely outcome is a set of 'light' processes, supported by a wealth of templates, tools, gadgets, software advisors and reusable components to make the proposed tasks fit into the modern lifestyle.

Sounds horrible? Maybe, but the alternative is that your neat, tidy, heavyweight process will be shelfware.

© Ian Alexander 2007

***RE*-verberations**

Vint Cerf: IP Evangelist?



Vint Cerf

In the January 2007 issue of *IEEE Computer*, Vinton G. Cerf, “the father of the Internet” and now vice president and Internet evangelist with Google, wrote about the Information Avalanche that is developing on the World Wide Web.

“Imagine, for example, in the year 3000 doing a Google search for a 1997 Microsoft PowerPoint presentation or rendering such a file using the latest version of Windows. Will we need to preserve the original software used to generate such information? Source code?”

The tape drive shall be maintainable for 1000 years

This echoes a concern familiar to older readers, namely how to ensure that existing data in archives such as those of NASA and ESA’s earth observation satellites remains readable, in the face of numerous hazards. These include the stretching of magnetic tape, the deterioration and demagnetisation of storage media even in air-conditioned archives, and not least the fact that the computers and tape drives are already obsolete and no longer manufactured, while the software is written in languages that few can now debug. The problem isn’t going to get any better: feel like copying

10,000 Terabytes of data from antique tapes, anyone?



But Vint Cerf’s main point was about IP, which geeky readers might assume (given the authorship) meant the Internet Protocol, as in the TCP/IP stack running on your computer. Oh no.

“The production of content through blogging, video/audio uploading, and other Internet-based techniques has made the IP scene more complex. Most of these works are self-copyrighted and rarely formally registered, and authors can knowingly or unknowingly incorporate readily available but copyrighted online material.”

Yes, it’s the domain of that well-known American species, *Homo litiginosus*. IP now means Intellectual Property, as in Rights, ie Lawsuits.



“The result is a formula for confusion, tension and dispute in many domains.”

Remember to add “IP Lawyer” to your list of Hostile Stakeholders!

***RE*-flections**

A Simple Profession

“No”, said the famous consultant. “You can’t talk to these people about ‘requirements’ or ‘shall-statements’. You heard it yourself, they don’t have ‘requirements’.”

“They just plan, consult stakeholders, and specify systems”, I said, somewhat bitterly. “I’m losing the plot.”

“They’ve never heard of requirements. Shall-statements may have meant something to a bunch of defence engineers back in the 1980s. No, this lot need a simple message. What do they really, absolutely *have* to have done before they start optioneering, choosing the best option from the available designs?”

“Well, they better know what their goals are, I mean, what they’re trying to do for the world.” I reflected for

a moment. “And they have to find out who all their stakeholders are, or they’ll get into a tangle.”

The famous consultant looked at me quizzically.

“And they’d better write down the assumptions they are crucially depending on.”

“And if they don’t know what they have to interface to, it’ll be the same cockup as with the stakeholders.”

“And before they spend a lot of money, they’d better have a reasonably convincing Business Case or frankly they’re just messing about,” I concluded.

“Well, those are the Requirements” said the famous consultant. “All the rest is just a lot of people who like sitting in a room spouting complicated theories that only they are clever enough to understand, and they have conferences where they preach to each other

about how important it all is, and lament that the rest of the world isn't listening."

"Surely that's a little harsh", I began.

"Hmm", said the famous consultant. "Perhaps Human Factors people are worse. But either it's simple and practical: commonsense stuff that everyone can see must be right, or it makes no sense at all. You're not inventing Nuclear Physics: you're just making sure that people know what they want before they go shopping."

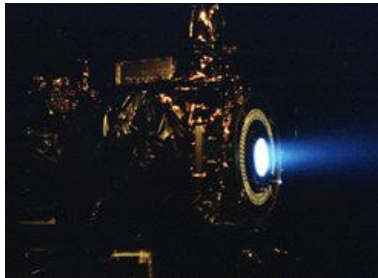
"Put like that, it does sound quite a simple profession", I said, feeling somewhat battered.

"Well, there's no point making things more complicated than they are," said the famous consultant. "Unless you're trying to blag people into giving you a research grant or something."

"It makes the world go round," I replied.

© Ian Alexander 2007

The Carpet Shall Fly at 300 Knots



NSTAR Ion Thruster operating on the DS1 spacecraft (Image courtesy of NASA)

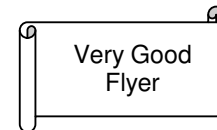
Satellites can be steered and even accelerated to interstellar speed by a tiny push, such as the pressure of the solar wind, or the miniscule but constant recoil from an ion thruster. This small device accelerates charged atomic particles to enormous speeds, generating a push no stronger than the weight of a coin. It's the very opposite of the titanic warp-drives imagined in Star Trek, but it's a practical device.

Similarly, the key to education is a small, steady push. It has to be just enough to keep up the momentum of daily progress. Too little and learning stops; but attempting to climb too fast makes learning stall.

Research is another area where the same rule applies (perhaps it's universal). Too safe and you get the usual plodding PhDs, Kuhn's "normal science" writ large, with no "paradigm shifts" from one decade to the next. Too bold and you get Cold Fusion and The Cure For Cancer, and sometimes invented evidence too.

Requirements likewise are best slow and steady. Too ambitious is no good; so is too tame. As with education, the right amount is what can be achieved only with barely-attainable effort.

Some think the way to guarantee success is to define measurable criteria. These help enormously, but they don't solve the problem. How do you want your carpet, sir? You can't just tell me you want it to fly well and be easy to control.



All right then, so I want it to fly at 300 knots at a height of 1000 feet, with tolerances of 20 knots and 50 feet. I want it to be controllable with a hand movement accurate to 1 degree in elevation and 1 degree in azimuth. There, that's nice and precise.

Ah, but it's not attainable, at least not with a Kilim or Soumakh weave, nor with a knotted construction. You can't just do what the "users" want: you have to do what works. That means obeying Newton's laws, not to mention everything else that's relevant in the environment: the Air Traffic Regulations, and getting an Airworthiness Certificate to name but two.

If you think about it, the dogma that all System Requirements should trace to (ie come from) User Requirements is just plain daft. Why on earth (OK, or up in the sky) should they do that? Stakeholders' goals are fine, but if their dreams are ever going to fly, they must be anchored in reality. That means complying with standards and regulations. And a just bold enough push towards the just barely attainable.

© Ian Alexander 2007

RE-partee

Proverbs

- Strike while the iron is hot.
- Look before you leap.

Well, they can't both be right.

RE-definitions

- Domain Modelling** – excuse for drawing diagrams.

- Analysis** – excuse for procrastination.
- Consultation** – way of seeming to follow a Systematic (q.v.) Process while actually taking no notice of the Stakeholders involved.
- Questionnaire Survey** – giving the users a yes/yes choice. Compare Consultation.
- Risk Register** – unread book used to demonstrate Best Practice (q.v.)

- **Best Practice** – constructing a Risk Register (q.v.)
- **Systematic** – thorough, good, praiseworthy, as in Analysis and Best Practice (q.v.). Compare Systemic.
- **Systemic** – careless, bad, blameworthy, as in ‘systemic failures’.

Safety Advice

“Fire Safety. In Emergenc[y] [sic], Do not panic and follow the advice of the fire department.”

Someone thinks the fire department’s training is not what it might be.

Prohibitions



Shouldn’t that be “No bathing, fishing, or dogging”? Wikipedia calls this Syllepsis, not Zeugma, by the way.

RE-publications

Book Review: Performance-Based Earned Value by Solomon & Young

Paul J. Solomon and Ralph R. Young
Wiley, 2007

Maybe as a requirements person you have never heard of Earned Value. Maybe you think it sounds like something that managers use, and you’d be right. Maybe you think it has nothing to do with requirements, and you’d be wrong.

Solomon and Young have put together a compact, clear, and very well organized book on Earned Value (EV). The prefix ‘Performance-Based’ is strictly unnecessary, as EV can’t mean anything else, but it makes for a more approachable book title. The cover carries the equation $PBEV = EVM + \text{Quality}$, below the Cost/Schedule/Technical triangle, conveying the message that PBEV is a value-added application of EV.

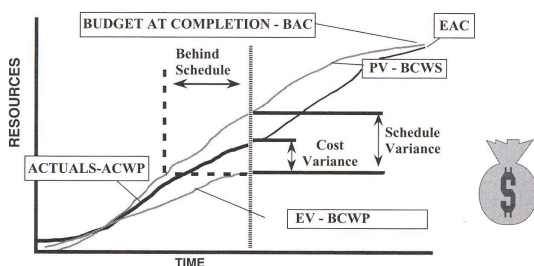


Figure A.3 S Curve.

The basic idea of EV is to model and graph the money spent against the work actually done. This is far better than congratulating yourself for being ahead of the spending at month 7 of your project, without any idea how much progress has actually been made. That’s not to say that EV has no subtleties, nor that it always gives the right answer. For instance, if a job gets stuck because you run out of paint after 19% of the surface is covered, and have to repaint everything when the new batch arrives, you have to restate the EV, and the 19%

of the effort already spent is treated as negative -- a cost not a benefit. This is clearly realistic. Older approaches which said you were 19% along the way were seriously misleading, both in time and in cost terms.

Like requirements engineering, then, EV is tightly focused on results. But this book is not of interest purely because of this analogy. In a welcome departure from project management tradition, the book interweaves chapters on EV with chapters on requirements.

The book begins as you’d expect with an overview of EV, followed by its principles and guidelines for its use. Chapter 3 then dives into the Product Requirements Baseline, ie what the EV is going to measure progress against. Common Sense! What a rare thing that is. The book then goes on to cover the need to maintain bidirectional traceability (ok, ok, you thought managers couldn’t even spell that), progress towards meeting product requirements, planned values and budget, variance analysis, scenarios, level of effort, risk management, changes to performance measurement, agile methods, requirements and EV, application to software development, and supplier acquisition management.

In other words, this is a remarkably open and comprehensive book. It is further strengthened by detailed appendices, covering the fundamentals of EV, guidance, Standards, Federal Acquisition Regulations, and worked examples.

This is a fascinating and valuable primer on EV for project managers who want to create practical and realistic measures of progress. That of course means being requirements-based; it also means understanding how to measure the right things, which is what EV is all about.

© Ian Alexander 2007

For more requirements book reviews, visit
<http://easyweb.easynet.co.uk/~iany/reviews/reviews.htm>

RE-sources

Books, Papers

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

Software Requirements Engineering Articles (India):
<http://www.requirements.in>

Mailing lists

RESG Mailing List

Subscribe: admin-mail-list@resg.org.uk with text "subscribe" in the subject line.

RE-online

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

Subscribe: majordomo@it.uts.edu.au with the body of the message set to "subscribe RE-online <your email address>"

Requirements Networking Group (RQNG)

www.requirementsnetwork.com

RE Yahoo Group

<http://groups.yahoo.com/group/Requirements-Engineering/>

RE-actors: the committee of the RESG

Patron:

Prof. Michael Jackson, Independent Consultant,
jacksonma@acm.org

Chair:

Dr Pete Sawyer, Computing Department, Lancaster University,
sawyer@comp.lancs.ac.uk

Vice-Chair:

Dr Kathy Maitland, University of Central England,
Kathleen.Maitland@uce.ac.uk

Treasurer:

Prof. Neil Maiden, Centre for HCI Design, City University,
N.A.M.Maiden@city.ac.uk

Secretary:

David Bush, NATS,
David.Bush@nats.co.uk

Membership secretary:

Dr Lucia Rapanotti, Computing Department, The Open University,
L.rapanotti@open.ac.uk

Publicity officer:

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

Newsletter editor:

Ian Alexander, Scenario Plus Ltd.,
iany@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

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

Industrial liaison:

Prof Wolfgang Emmerich, University College London,
W.Emmerich@cs.ucl.ac.uk
Suzanne Robertson, Atlantic Systems Guild Ltd,
suzanne@systemsguild.com
Gordon Woods, Independent Consultant,
Gordon@cigitech.demon.co.uk
Alistair Mavin, Rolls-Royce,
alistair.mavin@rolls-royce.com

Contributing to RQ

To contribute to RQ please send contributions to Ian Alexander (iany@scenarioplus.org.uk). Submissions must be in electronic form, preferably as plain ASCII text or rtf. Deadline for next issue: 7th June 2007

Joining the RESG

Visit <http://www.resg.org.uk/membership.html> for membership forms, or email membership-RESG@open.ac.uk