

## Viewpoints

‘Viewpoints’ is a regular section in *Requirements Engineering* for airing readers’ views on requirements engineering research and practice. Contributions that describe results, experiences, biases and research agendas in requirements engineering are particularly welcome. ‘Viewpoints’ is an opportunity for presenting technical correspondence or subjective arguments. So, whether you are a student, teacher, researcher or practitioner, get on your soapbox today and let us know what’s on your mind...

Please submit contributions electronically to Viewpoints Editor, Bashar Nuseibeh (ban@doc.ic.ac.uk). Contributions less than 2000 words in length are preferred.

## Is There Such a Thing as a User Requirement?

**Ian Alexander**

Chiswick, London, UK

We requirements engineers seem to be having quite a hard time tracking down the real, primal, User Requirement. In fact, it is as elusive as the Wild Man of the Appalachian forests.

When I actually meet ‘users’ – whatever they are, and that is an equally hard question, to which I will return – I find that they certainly have needs, but that these do not appear in canonical form at all. The needs come out in a rush, a mixture of complaints, design decisions, interface descriptions, current situations, and from time to time specific human–machine interface requirements (though not often). It is sometimes possible to isolate chunks of this as definite functions; it is rather easier to come up with required scenarios, which can be analysed as hierarchies of use cases.

Proper constraints are even harder to come by, even in safety-conscious industries such as rail and aerospace. Instead, safety and reliability and so on appear as oblique references to a mass of standards, precedents, policies, reviews and assessments, which never seem to add up to a neat and tidy safety case as understood by the academics.

In short, there seems to be a gap between theory and practice. Theory says that on the one hand we’ll find people who state what they want, and on the other, people who make things to please the first bunch of people. Instead, all the people and tasks and documents seem to be muddled up together.

This should be alarming news to those whose philosophy of system development requires a neat sequence from user requirement to system specification to design to test.

Now, I do not believe that I have been especially unlucky in my meetings with clients. I think they are, if not a random selection, at least reasonably representative of the industrial world at large. The people who I work with and for seem to have a good idea of their jobs, are generally hard-working, and are keen to provide a service. But they do not think of themselves as users or requirement providers.

This brings me to the term ‘users’. It is an odd word for the start of a process, as it clearly implies that something, some system, is being used. That system must already exist if the use is actual rather than potential, in which case we cannot expect to get pure, green-field requirements: we will get comments which refer, no doubt often obliquely, to the current system. That system is undoubtedly old, constricting and awkward, so the expression of wishes and needs for a new, custom system may well be distorted by the cramped present (or past) reality. Generals, for example, are classically accused of always preparing to fight the last war (or the last war but one) again, rather than the unpredictable next one.

Another possibility is that the ‘users’ do not yet have a system, so they are purely potential or planned users of a future system. In this case, they are being asked to describe how they would like something to be, when

they do not in fact know how they would interact with it. They are stretching their imagination from the limited present into the blue-sky future. The resulting requirements may be pure but are unlikely to be perfectly grounded: blue sky does not mean green field, perhaps.

For example, how will travel be when a translation-chip reliably and cheaply combines speech interpretation, grammatical analysis, translation and speech generation? Such a Star Trek system has already been prototyped in real life. Travel will, evidently, be easier in some ways. Perhaps the device will quickly become essential; or perhaps the rise of Business English will make it a mere gimmick. How much would you pay for such a device? As a business person? As a holiday-maker? Any answer seems risky, to say the least.

A separate challenge to the concept of a 'user' who specifies requirements is the mass market. How many users are there of Ford Mondeos, or Sony Walkmans? Evidently millions, for existing products; none, for future models, just as for custom-made systems. Here, it is clearly impossible to interview all the users, and equally impractical to offer options and variants to give each exactly what they want. Instead, the Marketing Department – if it does its job well – interviews a well-chosen statistical sample of the user community, derives user requirements, and supplies these to the Development Department. In other words, the marketing manager is a surrogate user. In many large mass-market firms, new products are under the control of a product manager, who is in a sense the surrogate customer. The product manager gets surrogate-user requirements from Marketing and places costed system requirements on development – and ultimately on production.

In other firms, product managers get requirements mainly from external customers, and place system requirements on either subcontractors or on other departments of their own firm. For instance, the makers of a telephone switch may receive interface requirements from telephone operating companies (telcos), and place requirements for specific racks of equipment on hardware manufacturers, and for specific program modules on software houses. Here, the concept of the end-user is remote. The ultimate users of the switch are ordinary citizens, but they neither buy nor know about the switch or its properties. Actual users of the switch include maintenance and diagnostic engineers, but their interface represents only a small part of the overall functionality. The switch's product manager is both a customer and a supplier within a larger supply chain (to be precise, a system/subsystem hierarchy). User requirements, such as those of the diagnostic engineers, are a relative rarity in such a scheme.

So far, I have broadly assumed, as it seems most developers assume, that users are customers. But is a Mondeo owner the user? It might be his wife – or her husband. It might be the children ... At least for something like a Sony Walkman there is a fairly close association between user and customer: apart from gift purchases, you buy one if you want to use one. But for a car, there is evidently a wider group of users than customers: for a start, it has seats for passengers as well as drivers. Firms may be well advised to start by interviewing, and advertising to impress, customers rather than users: after all, they have the money. Or perhaps the advertisers are ahead of us, as usual: they know all about 'pester power', about how to teach our children to ask us to buy.

What is the equivalent of pester power in a large organisation? Can soldiers, for instance, effectively clamour for better armour for their tanks, or bureaucrats for friendlier software for their workstations? Perhaps the gleaming green hardware on display in the world's trouble-spots and the antique green-screen character terminals used for airline reservations say something about the relative amounts of pester power in different professions.

It seems to me, then, that we can in fact start to make some operational definitions of some of these familiar but slippery concepts.

The idea of an end-user seems rather poorly founded, if we ask 'the end of what?' If all systems are interconnected then ends are hard to find.

A customer is someone who buys something. Customers have certain rights, in particular to set the scope of their expenditure: I'll pay for this but not for that, I can't afford more than so much. This is not so much a user requirement as a constraint which is applied globally and also to specific items, ideally in a properly documented requirements review.

A user is someone who actually or potentially makes a specific use of something. Users need not be customers; most users are probably not customers, given the centralisation of purchasing authority in many organisations and households.

Potential users cannot express definite, current requirements. Developments of technologically risky projects for such users need to manage risk by involving the users cooperatively, demonstrating prototypes and simulations, and obtaining feedback, so that systems become more real and requirements become more realistic in tandem. There is no point in trying to construct complete and accurate 'user requirements' at the start because nobody knows what they will find as development – of concepts as well as of systems – proceeds.

Current system users are experts in current systems, which thus serve in place of the 'prototypes and simulations' of blue-sky projects. Developments to improve or replace such systems are inevitably iterations, and begin from firmly based 'current working models' in actual service. Requirements from such users are always incremental, and rightly consist of suggested changes to design, interfaces, performance and functionality. Complete and consistent 'user requirements' are generally unnecessary in such a well-defined context. In fact, the time that would be needed to prepare such documents can be so long that they are out of date before they are complete.

In between pure mass-market products and pure custom products are the commercial off-the-shelf (COTS) components such as Word and Oracle (or even SAP and BAAN) that provide ready-made solutions to parts of people's problems. A lot of work is currently going into how to glue these things together effectively into systems. Users of such COTS systems are a diverse bunch, but they have one thing in common: they have no

need to specify 'user requirements' for their components, as they get what they get, like it or not.

## About the Author

Ian Alexander is an independent consultant specialising in Requirements Engineering and Business Process Modelling. He mostly works with Quality Systems and Software, Ltd, providing consultancy and training on requirements using the DOORS platform. His principal research interest is in improving the requirements engineering process through cooperative goal modelling. This approach is supported by his Scenario Plus! toolkit, which is the subject of numerous papers. He is currently consultant to a NERC project which seeks to speed the product introduction process for new jet engines. He helps to run the BCS Requirements Engineering Specialist Group and the IEE Applied Systems Engineering professional group.