Position Paper for the 2007 Design Requirements Workshop

Martin S. Feather
Jet Propulsion Laboratory, California Institute of Technology
Martin.S.Feather@jpl.nasa.gov

Abstract

As often stated by arguably the world's most famous requirements engineer, Sir Mick Jagger, "You can't always get what you want". One of the tasks of requirements engineering is to arrive at an acceptable compromise among "wants". Often this necessitates determination of the cost(s) and benefit(s) of various selections from among those wants. This is an area that has received some attention, but is far from solved, due to a number of challenges that complicate the problem.

The position espoused herein is that this is an area in need of further attention. Indeed, the workshop itself faces this problem.

The problem area

The importance of negotiating and prioritizing requirements is a recurrent theme in requirements engineering. For example, in [1] Davis looks at the need to perform requirements "triage", which he defines as "the process of determining which requirements a product should satisfy given the time and resources available"; in [2] Ruhe and Saliu focus on "release planning", which they define as "decisions related to selecting and assigning features to create a sequence of consecutive product releases". Key to these is the need to estimate both value (how much worth should be ascribed to a set of requirements) and effort (cost, schedule and other resources that will be needed to develop the system that would satisfy a set of requirements). Karlsson & Ryan's pioneering work in this general area [3] demonstrated successful use of relative prioritization (using the Analytic Hierarchy Process' pairwise comparison method) for both purposes, and used simple yet cogent "cost-value" diagrams as a means to present the results. Robinson & Fickas [4] advocated use of planning techniques to identify requirements conflicts, and generate resolutions, the latter "based on two methods: (1) analytic multi-criteria goal maximization (compromise within a constraint space) and (2) heuristic reformulations … to reformulate conflicting system requirements into similar non-conflict[ing] requirements…"

Complications

Estimation in support of decision making is complicated by the following:

Multiple stakeholders with differing values and different perceptions. Boehm et al's "WinWin" work addressed the issues arising in seeking mutually acceptable (i.e., valued highly by all the stakeholders) sets of requirements [5]. Application of visualization techniques within this approach is described in [6]. Different stakeholders may have varying degrees of *confidence* in their valuations, an issue addressed by, for example, Ullman's "Accord" system http://www.robustdecisions.com/index.html

Dependencies among requirements, making it hard to consider requirements one-by-one. In release planning, it is often the case that some requirements serve as necessary precursors to others. In almost all non-trivial systems there is some form of interference between requirements (an oft-quoted example being that of accessibility and security). Robinson et al's survey paper [7] termed the analysis and management of these dependencies "Requirements Interaction Management".

General challenges of prediction. Considerable work in this area has been done to address effort (cost and schedule) prediction, for example, COCOMO is a well-known cost-estimation model by Boehm [8] – see http://sunset.usc.edu/research/COCOMOII/index.html for up-to-date information on COCOMO II. Work is underway to build a similar kind of model, COQUALMO http://sunset.usc.edu/research/coqualmo/ that predicts *quality*, another key attribute in estimating the value of a potential product. Other models that predict quality include that developed by Ann Marie Neufelder http://www.softrel.com/, and the work of

Norm Fenton et al [9] with a Bayesian Belief Nets underpinning.

The confounding factor of novelty. Advancement from the concept/idea/prototype stage through to a robust, usable product for new and/or breakthrough ideas is fraught with uncertainty. NASA's "technology readiness level" (TRL) has been in use for over a decade as a scale to compare technologies [10]. Work continues to improve upon this, including taking into account the systems context, e.g., [11], and adapting it to software, e.g., [12].

Decision making despite uncertainty. Because the selection of requirements is being done at requirements time, there will inevitably be uncertainty in the predications and estimates. The key question is whether we can develop approaches that lead to better decision making in spite of this uncertainty. Interesting work by Tim Menzies suggests that it is possible to find "stable conclusions" from purely qualitative descriptions [13]. Julian Richardson, Dan Port and I recently reported finding that superiority of our risk mitigation selections remains relatively robust even when there is considerable "noise" in the estimates on which our decision making procedures [14].

Example approaches

These multiple complications suggest that estimating value and effort can be far from trivial. As in much of requirements engineering, there is an art to striking the right balance between the accuracy desired of the requirements recommendations that emerge, and the analysis effort it takes to yield them. Several approaches are listed next (this is not intended to be anything close to a complete list):

In mainstream practice the Quality Function Deployment (QFD) group decision making technique has been used to match value to cost. E.g., the Wikipedia entry currently states in part: "*QFD … techniques extend the original HOQ [House Of Quality] approach by deploying resulting "HOWs" from the top level HOQ into lower tier matrices addressing aspects of product development, such as cost, technology, reliability*" . The QFD methodology is based on establishing *qualitative* relationships, generally in the form of matrices.

Qualitative relationships also feature in the graph structures seen in the i*/NFR/GRL/Tropos threads of work active in academia and beyond (e.g., see Mylopoulos' keynote slides from RE'06 [15]). Proponents typically construct goal models to capture the decomposition of top level requirements goals into lower level constituents, leading towards design artifacts. Annotations on the relationships (the arcs of the graphs) indicate the qualitative strength of contributing or inhibiting factors.

By way of contrast, a *quantitative* approach is seen in the DDP work I have been involved in [16]. This is somewhat akin to QFD, insofar as it employs two matrices at its heart: one of them relates requirements to the risks (more generally, impediments) that threaten to detract from them, the other to relate instances the risks to the options available for mitigating them. However, rather than using qualitative values, DDP is built upon a numerical basis, with a straightforward probabilistic interpretation. DDP goes partway towards more general graph structures by permitting risks to be organized into fault trees [17].

In addition to the above there are many decision-making approaches applicable when the number of options from which to choose is relatively small, For example, the Analytic Hierarchy Process (AHP): "*The Analytic Hierarchy Process is a method for formalizing decision making where there are a limited number of choices but each has a number of attributes and it is difficult to formalize some of those attributes.*" from http://mat.gsia.cmu.edu/mstc/multiple/node4.html For a more extensive discussion, see for example http://mdm.gwu.edu/Forman/DBO.pdf

Needs/Opportunities

There is almost certainly never going to be a single method that unifies all of the above. Different methods will likely be appropriate in different circumstances. **At present we lack an understanding of which methods to use when**.

Generally speaking we also l**ack knowledge of the accuracy of methods' results** (admittedly this is not that case for all the methods – models such as COCOMO have been well calibrated). Like much of requirements engineering (or for that matter software engineering in general), there is a paucity of thorough experimentation and evaluation.

There is scope for cross-fertilization of ideas between the developers of the various methods. In one form or another all the methods face the recurring challenges of modeling and elicitation – picking the right level of detail to model, efficiently populating it with user inputs, validating those inputs for correctness and completeness.

There may be scope for "hybrids" of these approaches, or at least transfer of information from one approach to another. This is easier said than done. In attempting to do something similar in the world of risk assessment [18] my colleagues and I found that there are a lot of impediments to sharing information among risk tools, stemming from the well-known phenomenon of "semantic dissonance". I speculate that requirements tools and methods to do cost/benefit tradeoff decisions will exhibit similar challenges.

We should be willing to look for and **adopt from other fields analysis techniques that will operate over our models**. An example of such is seen in [19], where Avesani et al use of machine learning techniques to overcome scalability problems when using AHP to prioritize requirements. We see that Tropos researchers have adopted AI planning techniques [20]. I we have had the benefit of stimulating collaborations with Tim Menzies and Jim Kiper; together we who have explored use of clustering and machine learning to help identify key decision points [21].

Information Visualization is another field with methods that can be of help to us. I mentioned earlier Karlsson & Ryan's "cost-value" diagrams [3], an example from which is shown below:

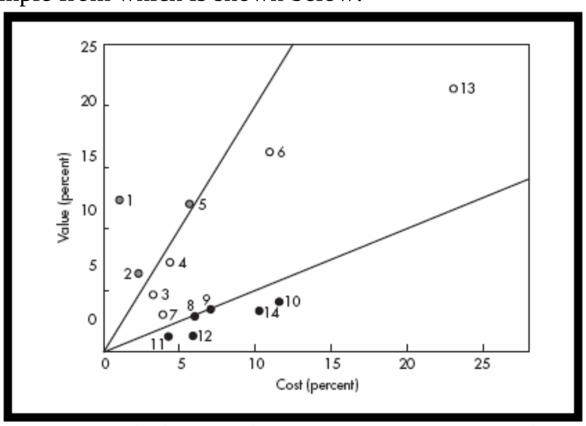


Figure 4. Cost–value diagram for the RAN project requirements. By not implementing the requirements that contribute little to stakeholder satisfaction, such as 10, 11, and 12, you can significantly reduce the cost and duration of development.

I also mentioned similar visualizations used within Win-Win [6]. Last year's International Workshop on Requirements Engineering Visualization was a venue at which we saw emerging applications of visualization within the RE field. These included an application of visualization to show quality attributes of a goal graph model [22] – the figure below is copied from that paper:

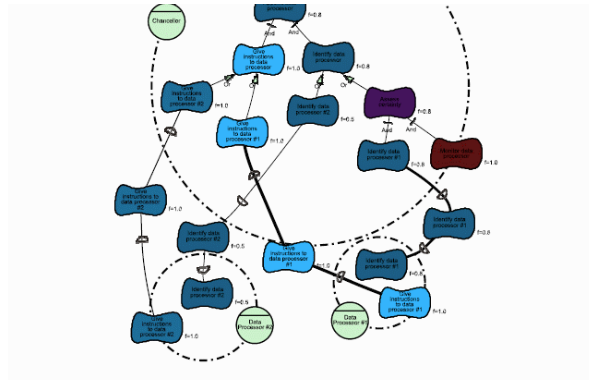
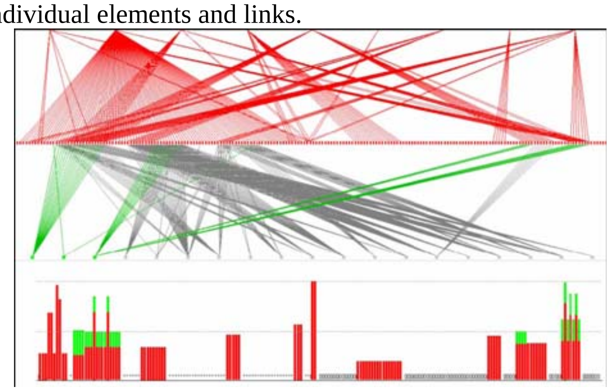
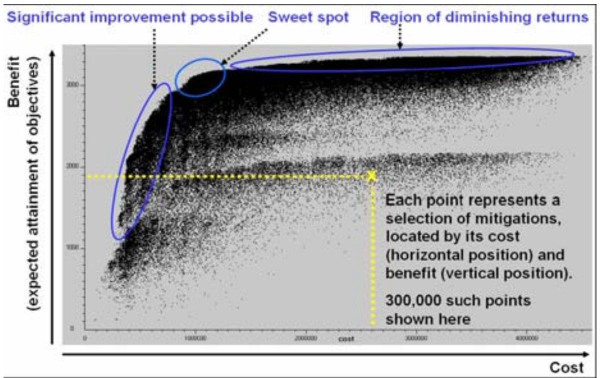


Figure 3. The quality attributes of feasibility, certainty, trustability and efficiency are shown on the goal model with visual clues

Diagrams such as this help by presenting models to the viewer in a form that is more conducive to overall scrutiny than, say, a purely textual representation. At that same workshop my colleagues and I presented a paper describing our experiences with visualization [23], for both the models themselves, and the results of calculations performed over those models. For example, the diagram below shows the "topology" of the data in one of our DDP models, and beneath it the bar chart displaying the quantitative results of calculating risks of a specific selection among the alternatives. Space precludes a complete explanation, but it is worth noting the evident contrast between DDP and the Tropos etc approaches: DDP has a simpler goal graph structure, but typically many more individual elements and links.



The diagram below is also from [23], and shows the tool-calculated cost-value tradespace for selections from among over 50 independent choices.



Note that the Karlsson & Ryan cost-value diagram plots the *individual* requirements, while the diagram above plots *combinations* of requirements. The latter is needed when there is a non-trivial interaction among requirements.

Since many of the problems we deal with will have *multiple* kinds of costs and *multiple* kinds of values, two static dimensions will not suffice for presenting these. The information visualization field pursues a variety of approaches to dealing with this problem, e.g., "parallel coordinates" [24], "Star Coordinates" [25]. Our field should look to these other venues as a source of appropriate techniques to adopt.

Workshop Challenge

The workshop itself could well lead to the need to perform cost-benefit analysis on the ideas that the participants will suggest. From the invitation letter itself it is clear that there are multiple areas of concern: "elicitation, validation, and management of design requirements", spanning multiple application environments "such as e-business, pervasive computing, and large enterprise systems". If there are indeed representatives of the multiple research disciplines listed, "including such areas as design science, information systems, behavioral studies, software development, organizational design, and requirements modeling", there is sure to be no shortage of ideas. Which of these will deserve to be in the "10-year agenda"? What will be the costs of, and other obstacles to, pursuing these ideas? How will the workshop identify the options and arrive at decisions? I claim it would be a sad commentary on the value of our methods if the workshop did not make use of any of the practices promoted by its participants.

Very briefly, here's a possible way to approach this in the DDP style:

1) Brainstorm to emerge with a list of the potential objectives of the research agenda for the decade.
2) Brainstorm and list the (current) impediments to those objectives (possibly organized into small and/or/not trees if there is some logical structure to the way impediments combine), linking each impediment to the objective(s) it impedes.
3) Brainstorm and list the potential areas of research that, of successful, would address impediments, linking each research area to the impediment(s) it overcomes.

This will lead to a topology akin to that shown in the red/green DDP goal graph on the previous page, from which we will be able to discern recurring research needs, unaddressed problem areas, plausible opportunities for improvements, etc.

I suspect that some of the other approaches represented at the workshop would also be applicable – the ones I'm most aware of in this respect are Tropos (John Mylopoulos) and KAOS (Axel van Lamsweerde); possibly GIBIS (Colin Potts)?

Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and funded through NASA's Exploration Systems Mission Directorate. I particularly thank JPLers Steven Cornford and Kenneth Hicks, who have been instrumental in promoting the successful application of the DDP method that Cornford himself invented.

References

[1] A. M. Davis, "The Art of Requirements Triage", IEEE Computer, pp. 42-49, March 2003.

[2] G. Ruhe & M.O. Saliu, "The Art and Science of Software Release Planning", IEEE Software, pp. 47-53, November/December 2005.

[3] J. Karlsson & K. Ryan, "A Cost-Value Approach for Prioritizing Requirements", IEEE Software, pp 67–74, September/October 1997.

[4] W.N. Robinson & S. Fickas, "Automated Support for Requirements Negotiation", AAAI-94 Workshop on Models of Conflict management in Cooperative Problem Solving, 1994.

[5] B. Boehm, P. Bose, E. Horowitz & M. Lee "Software Requirements as Negotiated Win Conditions",

Proceedings 1st International Conference on Requirements Engineering, Colorado Springs, Colorado, pp 74-83, 1994.

[6] H. In & S. Roy, “Issues of Visualized Conflict Resolution”, Proceedings of the 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada, pp. 314-315, 2001.

[7] W.R. Robinson, S.D. Pawlowski & V. Volkov, “Requirements Interaction Management”, ACM Computing Surveys 35(2), pp. 132-190, June 2003.

[8] B. Boehm, “Software Engineering Economics”, Prentice Hall, 1981.

[9] N.E. Fenton, M. Neil, W. March, P. Hearty, P. Kraus & L. Radlinski, “Project Data Incorporating Qualitative Facors for Improved Software Defect Prediction”, ICSE PROMISE workshop, 2007.

[10] J.C. Mankins, “Technology Readiness Levels: A White Paper”, NASA, April 1995 – available from http://www.hq.nasa.gov/office/codeq/trl/trl.pdf

[11] B. Sauer, J. Ramirez-Marquez, D. Verma & R. Grove, “From TRL to SRL: The Concept of Systems Readiness Levels”, Conference on Systems Engineering Research, Los Angeles, CA, 2006.

[12] R. Mackey, R. Some & A. Aljabri, “Readiness Levels for Spacecraft Information Technologies”, IEEE Aerospace Conference, Big Sky, MT, pp. Vol. 1-391-Vol. 1-398, 2003.

[13] E. Chiang & T. Menzies, “Simulations for very early lifecycle quality evaluations”, Software Process: Improvement and Practice, 2003.

[14] J. Richardson, D. Port & M.S. Feather, “Robust Derivation of Risk Reduction Strategies”, IEEE Aerospace Conference, Big Sky, MT, March 2007.

[15] J. Mylopoulos, “Goal-Oriented Requirements Engineering, Part II”, Keynote 3 at the 14th IEEE International Requirements Engineering Conference, Minneapolis / St Paul. MN, 2006 – available from http://www.ifi.unizh.ch/req/events/RE06/ConferenceProgram/RE06_slides_Mylopoulos.pdf

[16] M.S. Feather, S.L. Cornford, K.A. Hicks & K.R. Johnson, “Applications of tool support for risk-informed requirements reasoning” Computer Systems Science and Engineering, 20(1), pp. 5-17, January 2005.

[17] M.S. Feather, “Towards a Unified Approach to the Representation of, and Reasoning with, Probabilistic Risk Information about Software and its System Interface”, 15th IEEE International Symposium on Software Reliability Engineering; Saint-Malo, Bretagne, France, pp. 391-402, November 2004.

[18] M.S. Feather, S.L. Cornford, L. Meshkat & L. Voss, “Overcoming Obstacles to the Exchange of Information Between Risk Tools”, Space Systems Engineering and Risk Management Symposium, Los Angeles, CA, October 1005.

[19] P. Avesani, C. Bazzanella, A. Perini & A. Susi, “Facing Scalability Issues in Requirements Prioritization with Machine Learning Techniques”, 13th International Conference on Requirements Engineering, 2005.

[20] V. Bryl, F. Massacci, J. Mylopoulos & N. Zannone, “Designing Security Requirements Models through Planning”, CAiSE, June 2006.

[21] T. Menzies, J. Kiper & M.S. Feather, “Improved software engineering decision support through automatic argument reduction tools”, Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering; San Francisco Bay, California, July 2003

[22] N.A. Ernst, Y. Yu & J. Mylopoulos, “Visualizing non-functional requirements”, International Workshop on Requirements Engineering Visualization, St. Paul, MN, September 2006.

[23] M.S. Feather, S.L. Cornford, J. Kiper & T. Menzies, “Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation”, International Workshop on Requirements Engineering Visualization, St. Paul, MN, September 2006.

[24] A. Inselberg & B. Dimsdale, “Parallel coordinates: A tool for visualizing multidimensional geometry”, IEEE Visualization Conference, 1990.

[25] E. Kandogan, “Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions”, IEEE Information Visualization, Hot Topics, pp. 4-8, 2000.