# Requirements Management Obscures Systems Engineering

Jack Ring, February 25, 2006

I have discovered a major reason why many people do not see the need for systems engineering (SE). This became clear while attending the tutorial, "Writing Good Requirements"[1] and was reinforced by subsequent webinars.[2,3]

The reason people do not see the need for systems engineering is because they are being told, and apparently believe, that Requirements Elicitation/Engineering/Management, hereinafter called RM, is a viable way to traverse rapidly from 'Project Day One' to 'Start of Development.' Such siren song is powerful, indeed. It isn't as if SE and RM were compared and SE lost. It isn't as if RM is portrayed as the adequate, accurate and timely documentation of SE decisions. The problem is that SE is not mentioned or is misrepresented as a UML activity, while RM is portrayed as the key to initializing and ensuring development.

Although the "Writing Good Requirements" session triggered this discovery the balance of this paper is not a critique of the "Writing Good Requirements" session[4] per se, because when it comes to actually writing valid requirements statements the advice given is valuable. Rather, this paper seeks to alert the reader to how a preoccupation with requirements can suppress the appreciation of any perceived need for the due process of systems engineering. Further, this paper recommends ways for embedding the RM into SE practices. Although important today, these ideas become critical to success in the forthcoming era of model-based systems engineering.

## *Delight*

The value of complete, disambiguated, easily comprehended statements of requirements is not an issue. The tutorial, webinar and many other advisories regarding requirements writing encouraged clarity of articulation and gave many examples. The tutorial included work sessions to drive the points home, thus amply demonstrating that well written requirements are not easy to achieve.[5,6]

Other delightful messages included:

- "Almost all requirements work has to deal with interfaces" was a great admonition. Many examples illustrated the technique of referencing specific entries in an Interface Control Document.
- The tutorial emphasized separating requirements for product from requirements for the SOW. The deliverable system stems from, but is distinct from, the work system that produces it, thus the requirements must be segregated.
- Levels of requirements in the context of a system hierarchy were acknowledged. Also, 'flow down' and traceability (including labeling for clarity of 'derived requirements').
- A glossary was said to be helpful.
- Supplying a rationale for each requirement was stressed. Examples did not go so far as justifying the requirement, only describing the author's perspective.
- It was noted that managing more than about 150 requirements is not practical without some sort of tool. The tutorial noted that none of the current Requirements tools do a complete job.

---

1 By Ivy Hooks, in Albuquerque, May 6, 2005, sponsored by the INCOSE Enchantment Chapter. Approximately 70 attendees paid approximately $160 per person for a 74 page handout and seven hours of presentation filled with facts, ideas and examples. Ivy clearly demonstrated her expertise regarding requirements, well earned over several years of practice and teaching. Attitudes of attendees at the end of the session indicated that the majority were quite positive about the experience though somewhat embarrassed about their newly-recognized level of (in)competence.

2 http://www.telelogic.com/corp/download/index.cfm?id=3649&FileType=HTML, and (ibid) cfm?id=3698&FileType=HTML.

3 http://www.mks.com/go/thequestevent and http://www.mks.com/go/reqseventoct.

4 Ivy cautioned that this one day session would not cover the elicitation, management, or validation aspects of Requirements Management. However, most of the exercises involved the elicitation, management and verification of requirements.

5 Ivy quoted Tom Gilb as saying that after the seventh version of a requirements document he still found about one defect per page.

6 My high school English teacher would have been amazed that attendees had not learned this aspect of writing before graduating but so be it. Interestingly, at the end of the session when Ivy asked attendees to verbally share their thoughts regarding next steps they would take toward Writing Good Requirements no one mentioned taking a course in English grammar.

  

## *Disappointment*

RM advice was not related to systems engineering products or process. The tutorial made only a couple of passing mentions of SE and system architecting. The webinar did not note any.

The divisive practice of categorizing requirements as Functional and Non-functional was featured (although not labeled as divisive).

The tutorial was all about text. No advice was given about graphics and images as requirements communication techniques.[7] This is important because, as John Warfield has pointed out, most people presume that others readily understand graphic notation when, in fact, most do not, at least without adequate annotation.

Requirements writing advice dealt with the content and structure of requirements but did not deal with the context of requirements; i.e., exploring 'where do requirements come from' (other than the tutorial's strong message that requirements come from interviewing customers and the webinar's presumption that stakeholders understand their needs). While highlighting the importance of questioning customers no clue was given about ensuring that such unstructured conversation with responsible and affected parties does not inject all sorts of unstated assumptions about system function and feature or, worse, stated assumptions that don't make sense. In contrast, consider these remarks from Prof. Derek Hitchins:[8]

> "It has been my sad experience that customers and clients do not always know what they require – certainly not in any detail. Talking to several different customer representatives in an attempt to elicit requirements is fraught with risks: each representative may have a less-than-holistic vision of the solution, and so – in effect – the elucidator is cooperating in a process of Cartesian reduction, i.e., picking up aspects of (differing) solution visions that are mutually inconsistent and incoherent. This is one reason why so many requirement statements contain inconsistent and even downright contradictory statements. Sorry, the customer may think he knows what he wants, but he speaks with many tongues, some of them forked. Rarely does he know what he *needs*. Worse, if talking to an expert customer, e.g. a military officer, experience shows that the officer takes for granted that the elicitor has a sound grasp of the operational situation. The officer generally does not appreciate how much he knows, and how little the elicitor knows. As a result, the military officer does not convey information which, to him, is tacit knowledge. The elicitor is blissfully unaware of this tacit knowledge, which is generally vital to incorporate in any useful requirements – if the resulting system is to be useful to the e.g., military officer. Such requirement shortfalls are legion in IDA systems, for instance, where much of the expertise is wrapped up in the operator, who is the focus of the present system and will be the focus of the new system."

These sources of RM advice could be much more beneficial to their constituencies if they:

- Acknowledged the context of requirements writing, e.g., mission analysis, system design and architecture, systems engineering for specification of components, etc., and indicate whether, with examples, requirements statements differ in each of the context phases.
- Dealt with the issue of 'when are requirements sufficient?' In other words the 'stopping rule' for writing requirements.
- Emphasized the value of stating a justification for each instead of just stating a rationale.[9] Showing, for example, the justifying or vetting of requirements statements with respect to MOE's, system principles, etc.
- Highlighted the opportunity to capture and organize variables and parameters, key units of information in system models, beyond just a glossary, e.g., in a taxonomy or, preferably, an ontology of the problem space and solution space.
- Showed cross references at finer level of detail than just documents.
- Mentioned vetting requirements to ensure suppression of unnecessary variety or duplication.

---

[7] Other RM guru's recommend UML diagrams but these generally add more complexity thus confusion to the communication.

[8] Private communication as a result of reviewing a draft of this paper.

[9] Justification examples given in the webinar were vague, not consistent with the espoused qualities of good requirements.

  

## *Opportunity*

The tutorial recommended ways of eliciting requirements that are not consistent with SE practices. If proper SE precedes requirements writing then the 'write right' advice is good. But if proper SE does not precede requirements writing then the tutorial advice may yield a pile of readable rumor, material that is likely to be not as consistent with Requisite Variety, Parsimony and Harmony as if SE were done.

## *Recommendations*

1. INCOSE should exert more influence on purveyors of RM advice.
   - Tutorials, papers and posters that focus solely on requirements and thus deflect the learning and adopting of good SE practices should not be considered for INCOSE-sponsorship.
   - INCOSE or any other vigilant organization of SE practitioners should 'certify' member education material, especially the material pertinent to the evolution of SE.
   - INCOSE should counsel independent purveyors of requirements advice to ensure that such advice is presented in the context of systems engineering.
2. For those looking forward to Model-based SE, RM thinking must be excised from intended practitioners.
   - Requirements at the Problem phase or Acquisition phase should be limited to statements of intended effect on the problematic situation, in the form of MOE's and Standards of Acceptance. All other observations about the problem space should be captured in a 'mission model' or equivalent. Ideally, such statements should be stratified between applicability to operational system vs. applicability to SOW or System realization system.
   - Requirements at the system design and architecting phase must acknowledge the *Wymore Six*[10] or equivalent adequacy criteria.
   - Statements of the 'system shall' form must be carried as a semantic net such that interrelationships among them are clear (preferably visualizable). Such statements must be treated as hypotheses to be verified during design and architecting. Therefore the web if interrelations must be clear.
3. For systems undergoing model-based development and evolution (including System of Systems) Practitioners need guidance that describes how to establish and write Real Requirements. Where:
   - *Real* signifies meaningful, adequate, accurate, timely, and coherent.
   - *Establish* signifies discovering, proposing, justifying, explicating,
   - *Write* signifies expressions in text, graphics, images, video, sound, etc. and signifies organizing the expressions in taxonomy or ontological form sufficient for efficient and verifiable communication with All Responsible and Affected Parties.
   - *Model-based* signifies focus on the meta-model of the system before or in coordination with a focus on the conceptual, logical or physical model of the system.
   - The material should clarify the kinds of requirements (semantic level) by SE phase as indicated in examples in Table 1 on the next page. Table 1 is still under development but the gist of the idea should be evident from this version. Readers are invited to collaborate in evolving this table by contacting jring@amug.org.

## *Coda*

Derek Hitchins[11] observes:

> "Requirements should provide a coherent, consistent view of the solution system, operating in its future environment and interacting with future other systems. To develop necessary and sufficient requirements, then, it is essential that the first phases of the overall systems engineering process should have been undertaken. Perhaps they have, by the customer's own

[10] A. W. Wymore identified six categories of requirements, Input/Output, Performance, Technology, Cost, Test and Trade-off as the necessary and sufficient bounding of a system to be designed.

[11] Private communication as a result of reviewing a draft of this paper.

  

systems designers and architects. If so, then a full and consistent set of requirements would include descriptions of the problem space, would show how the proposed solution solved the problem, and would include functional, and probably physical and intra-communication architectures, too. It would also include – essentially – clear descriptions and definitions of how the constructed system was to be proved, i.e., in what environment and interoperating with which other systems, real or simulated.

If, on the other hand, no one has undertaken the first phases of the systems engineering process then, eliciting requirements simply by talking to the customer's many representatives is a brass-bound recipe for disaster. And you could, no doubt, cite hundreds of examples to prove the point."

| Phase | Purpose of Requirements | Semantic Levels of Requirements |
|---|---|---|
| Identification | Clarify intended effect(s) on problem situation. | ▪ ConOps (Capabilities)<br>▪ System Context Diagram<br>▪ Key Tensions<br>▪ Mediation Preferences (gradients)<br>▪ Measures of Effectiveness<br>▪ Standards of Acceptance |
| Design/Architecture | Ensure that a critical mass of system capabilities is identified. | ▪ Performance<br>▪ Technology<br>▪ Cost<br>▪ Test<br>▪ Tradeoff<br>▪ Orchestration (interoperation)<br>▪ Component budgets (thermodynamics, informatics, teleonomics such as size, weight, failure, speed, capacity, motivation, risk taking, styles, attitudes, etc.)<br>▪ Modularization (Subsystem, Subassembly and Component) |
| Engineering | Identify the functions and features the components must have. | ▪ Viable implementation technologies<br>▪ Verification of budget conformance<br>▪ etc. |

**Table 1. Examples of Semantic Levels of Requirements by SE Phase**
(A work in process. Contact jring@amug.org with collaborative thoughts.)

## *References*


- Brown, R., "Write Right First Time," Literati Newsline, retrieved 1/21/1998 from http://www.mcb.co.uk/literati/write/htm
- Gopen, G. & Swan, J., "The Science of Scientific Writing," American Scientist, vol 76, Nov-Dec, 1990.
- Meszaros, G. & Doble, J., "A Pattern Language for Pattern Writing," retrieved 9/29/97 from http://st-www.cs.uiuc.edu/users/patterns/Writing/patterns.html.
- Warfield, J., "Understanding Complexity: Thought and Behavior," Ajar Publishing, 2002.
- Wymore, A. W., "Model-Based Systems Engineering" CRC Press, 1993.


***Jack Ring*** is a Fellow of the International Council on Systems Engineering (INCOSE), and co-chairs its Intelligent Enterprise Working Group. He has more than forty six years of experience as executive, system thinker and management practitioner in aerospace, industrial, commercial and public sectors. His experience includes project, engineering and product manager as well as market research and competitive intelligence director, and corporate strategy and development VP. He has mentored more than a dozen high tech startups. He is sole proprietor of Innovation Management and a founding partner of Kennen Technologies LLC.

  