10 Golden Questions for Concept Exploration and Development

Dr. Dan C. Surber, CSEP
Raytheon Technical Services Company, LLC
6125 E. 21st Street
M/S 31
Indianapolis, IN 46219-2058

Phone Number: (317) 306-4114
Fax Number: (317) 306-4949
E-mail address: **D_C_Surber@raytheon.com**

Abstract. Project engineers and development teams must be able to quickly understand the customer's need. There are many tools, methods, and processes suggested for conducting "Concept Exploration" and "Concept Development". The author believes that there are "10 golden questions" which get the requirements elicitation done right. They apply to any Product (knowledge, good or service), system, or organizational structure. The "10 Questions" go a bit further than grammar school's: "who, what, where, when, why, and how." Interaction with the customer/user illumines a key aspect of the system solution, "How does failure affect customer satisfaction?" Asking, "What if the product, (seen at its various levels of decomposition such as, "system/product/component"), FAILS to satisfy these 'requirements'?", leads the designer to a better system solution. These answers take one to the next important discovery, answering, "How do we achieve mitigation and control of any critical failure modes and their effects on mission success, (through design, manufacturing, materials, and training)." This is the true purpose of the systems engineering lifecycle.

Overview

Faster, Cheaper, Better. Today's environment for systems developers is ultra-competitive, whether it is for the Department of Defence (DoD), commercial products, medical systems, automotive or consumer electronics. Managers and business analysts want pinpoint precision on cost and schedule, and the customer wants ultimate satisfaction with "no failures". Yet the crush of business demands on the developer's time, money and performance have not altered the basic challenge for any system development: what must the system do? Expanding this line of questioning usually leads to a number of "aha" events as the customer/user is led by the developer to explain the answers to the same questions most 5th grade English teachers spouted: Who, What, Where, When, Why, How?

There are standards and guides, used by NASA, AIAA/ANSI, and DoD to help with processes for answering these questions. What does a system developer in the commercial world do for some guidance and help? That is what this paper and its accompanying presentation will endeavour to explain. The fancy word is "CONOPS", short for Concept of Operations and Support.

The Basic Premise

First Things. The developer's fundamental challenge is get from concept to producible design. There are tools like Enterprise Architectures, Popkin Tool for architectures, IDEF diagrams, Use Cases, the 9 views in the UML, and entire processes like the Quality Function Deployment (all four tiers) that are touted for their power to help define a system from a concept. (Cohen, 1995). Most of them can be learned and mastered given enough time and effort, and with a lot of OJT (on-the-job training). Many of them require an "enabling tool", such as a database, and all tool vendors require an on-going licensing agreement. So where does that leave system developers in smaller companies with restricted resources? The author believes the answer is neither "not helpless", nor "hopeless", nor "out in the cold."

Next Things. The Software Engineering Institute at Carnegie-Mellon University has developed an entire family of capability maturity models for enterprises that desire to develop systems in a repetitive, effective manner that promotes continuous, incremental improvement and delivers high quality products at competitive cost. At the heart of the CMM-I for Software, Systems, Supply Chain and Integrated Product and Process Development is the premise that systems can not be developed without understanding the needs of all the stakeholders, and the constraints imposed on the system solution (both internal and external). (SEI, 2000)

Middle Things. The list of "10 Golden Questions" seek to explore the system concept using the same abstracted, three-tier approach used in the CMM-I: System level, Product level, and Component Level. (SEI, 2000) The list also includes a key question underlying the customer's responses to the questions: What effect on satisfaction does a failure have, seen through the customer's eyes. The importance of getting the customer to explain the CONOPS for the system can not be emphasized enough, since it forms the context of "what does failure to do 'x' mean?" Most customers come in the door with "wants and needs". It is the task of the systems engineer to "elicit the customer's real requirements…" through dialogue and exploration of the underlying concept of operation, maintenance, support and disposal. (Hooks, 2000)

Last Things. Using the questions does not guarantee that the developer will create the right system for the customer's need, nor that the system will be done right. However, using them does ensure that the developer is armed with much more of the knowledge about what will most satisfy the customer's need, at the beginning of the product development life cycle. Use of good systems engineering principles, processes, tools and project management discipline will help ensure the right system is built right. The author recommends that the answers obtained through the use of these questions be used to "seed" the requirements analysis, development and preliminary design processes within the enterprise product development process. Retention of this data in DOORS or other suitable requirements database will help trace the concept to the solution space and its methods of verification.

The "10 Golden Questions"

The Big Picture. This paper will provide a brief look at each of the "10 Questions" in the remainder of this discussion. Usage of the questions is meant to first explore the SYSTEM level; then into the PRODUCT level; and, finally into each product's COMPONENTS. Implicit in the answers to each question is to also understand the "effect of failure to meet/satisfy that question". Keep this in mind as the questions are reviewed. There is also an implicit "iteration" loop between these levels, as knowledge is gained at each hierarchical level. This "sharing" is meant to go both vertically and horizontally within the system hierarchy.

This early activity, while called Concept Exploration and Development, is a powerful ‘driver’ on the end result: the system architecture, its design, and success of the integration, verification and system validation effort. Time and money spent in the Concept Exploration and Development phase is well invested, based upon the author’s own experience. Remember also that systems are composed of hardware, software, tools, training, technical data, people, facilities, and system data. (Rechtin, 2002) Architecture decisions made in the first 20% of system’s development can affect almost 70% of its ultimate cost. (Blanchard, 1998; Buede, 2000)

1. **Who are the System Stakeholders.** Most authors writing on the subject of requirements development emphasize the importance of understanding the customer’s need. Jeffrey Grady argues that there is truly just one requirement, the “need”, as everything else is derived from it (Grady, 1993). The CMMI model stresses the importance of understanding the INTERNAL, as well as the EXTERNAL, stakeholders’ expectations. Clarity of stakeholders is just as important as understanding the system context (item 6) and the system concept of operations and support (item 5). Consider this question as the understanding of the concept development within the enterprise and its environment.

2. **What are the System Goals and Objectives.** The perspective of the person, or the organization, making inputs will affect the stated Goals and Objectives for the system under investigation. Perceptions are an enormous influence in what is said, and how it is weighted. The systems engineer must gather “all” the points of view, and then filter through them to see the FULL picture of the system concept, as it is envisioned by the group of stakeholders. Fundamental questions regarding the maturity, or risk factor, of the technology and the market(s) targeted by the system can yield a large number of implied requirements, constraints, and other expectations.

3. **What is the market for this System.** The commercial product development “world” starts with an analysis of the customers, their needs and expectations, and the markets for a product. Then the firm risks its own capital to do the product development. A DoD acquisition is quite the opposite. However, both “markets” affect one significant source of requirements: product safety, reliability, and homologation. DoD systems often have these sources of requirements called out in their Statement of Work (SOW) or their System Specification. Such is not often the case for commercial products. The market of intended sale often defines the regulations to be satisfied for safety, reliability and homologation. The firm must have a solid and repeatable process for developing products if it is to ensure a reasonable profit on the finished system.

4. **What are the external constraints on the System.** The systems engineer is interested in the external interfaces, and the external and internal constraints, which will be imposed on the system. A constraint is seen as a type of requirement, and most often it means that the system being developed will have to ADAPT to the constraint…this means early definition and then rigorous control of that “interface” in order to ensure the system meets its requirement during integration and testing. Mr. Thomas Stephens, Chief Engineer for the Engineering and Production Support business unit of Raytheon Technical Services Company, LLC has noted that “Constraints can be any external influence on the system – including org structure of implementation team, teaming relationships, cost.... as well as technical.” This is an important distinction for the successful development of a full understanding of external constraints on the system.

5. **What is the CONOPS for the System.** Most commercial, and many military, product

development programs fail to adequately staff, develop, and design for the SUPPORT needs of the system after it becomes OPERATIONAL. The author believes that the only way to remedy this behavior is to DEMAND that the Concept Exploration Phase includes an explicit discussion with the customer, acquisition, and end users. This discussion must specifically detail the SUPPORT concept for the system, even at this conceptual stage. There most definitely going to be REQUIREMENTS defined in these expectations, goals and objectives. An enormous part of a system's Life Cycle Cost is its support needs during its operational phase. (Blanchard, 1998)

The DoD acquisition process mandates that a Concept of Operations, or Operational Concept, document be developed as part of the system concept exploration and serve as an input to the System Functional Review (SFR). The Use Case view in the UML is also an excellent way to extract the answers to "Who, What, Where, When, Why, and How?" The stakeholders and the CONOPS will help to understand the goals and objectives of the system.

6. **What is the System Architecture context.** This question is meant to focus on the people, facilities, support equipment, tech pubs, hardware, software, data and processes that will comprise the system once it is defined, designed, built and tested. (Rechtin, 2000) This question is asking the team to visualize how the concept will be produced, packaged, shipped, stored, readied for operational use, supported, and finally its disposal. Expectations for Pre-planned Product Improvement (P^3I), technology refreshing, dealing with obsolescence, and future systems integration (growth), will all affect decisions about the final architecture selected for the system.

7. **What are the man-machine interfaces to be satisfied by the System.** Operators and maintainers need to be considered in understanding the expectations for the man-machine interface, and the machine-to-machine interfaces. Networking technologies are making it much more likely that a great deal of the system communications will be on a network, and may not require a man-in-the-loop or even desire to have that interface. Many systems are adopting a report-by-exception method of reporting health and status as well. Use of any existing interface protocols is an important expectation to discover during this early phase of the system definition.

8. **What are the Key System Attributes.** There are many methods and tools for discovering and documenting the key requirements of system attributes. This author likes the approach offered by Jeffrey Grady in *System Requirements Analysis, 1993.* He proposes that there are four types, or categories, of requirements for a system. They are (1) performance; (2) environmental; (3) interface; and, (4) design constraint. The key factor in determining if they are true requirements is "can they be stated as a value, relation, units and method of verification." If the answer is "NO", then the systems engineer is still working with "needs" and must decompose further. In DoD systems they often identify Measures of Effectiveness (MOE) that relate to mission success. Commercial system developers can define similar "requirements" for their systems.

9. **What are the System functions (behaviors) that satisfy the Key Attributes.** Now the systems engineer can start to have some fun. Identifying WHAT the system must do is an essential task to begin before the designers (hardware or software) start to "leap" to the solution. Remember that the key to innovation and customer delight is maintaining the "solution space" at its maximum during concept exploration and system definition. Early commitment to design solutions often causes a sub-optimized system with problems that are

not found until integration and test…very expensive. (Blanchard, 1998; Buede, 2000)

10. **What happens to "success" if the System FAILS to meet any of the above.** A chief intent of this method of using "10 questions" is to intentionally ask the question "What if the system fails to do this?" The concept is to do something akin to the Functional Hazard Assessment demanded by the Federal Aviation Administration (FAA) when a commercial transportation system is being developed. The developer must answer the question "How does your design mitigate and control the potential hazards this system may encounter?" By doing so, the developer understands which system functions are CRITICAL, and which parts of the design perform those functions. Systems engineers are interested in interfaces, functional, logical and physical…because most of the failures occur at the interfaces. (Leveson, 1995) This information can then be passed to the design engineer(s) for a more robust implementation of this concept through preliminary design and detailed design. Traceability of criticality and mitigation through design to verification also helps ensure key system behavior that is essential to customer-defined success is not lost during iterations and change.

SUMMARY

The fundamental challenge for the developer will not go away…they must still move as quickly as possible from concept to design and finally to manufacturing. Along the way the designer must be systems engineer and find the "needs" that the customer has not made known. Prioritized requirements that are the system's key attributes for the chain of understanding leading to effective designs. The developer (team) must see the system's life cycle, and "be" the maintainer as well as the operator. A successful developer (team) must also understand the inherent architecture in which the system's end design will operate, and how failure of a system's individual requirements (key attributes) will affect the architecture, mission success, and ultimately, the customer's need.

It is essential that the initial analysis team that developed the concept(s), requirements, and any trade studies capture this data in the "requirements database" so that further elicitation and decomposition of the requirements and their deployment through the design process can be traced to functions, interfaces and methods of verification. The insight the team gains from use of these 10 questions can accelerate the product development process and improve the hand off of a solid concept to the functional analysis team and the identification of key measures of effectiveness and initial technical performance measures.

CONCLUSIONS

More formalized structures and methodologies can be used for concept exploration and definition. However, most of them demand tools with databases, and licenses, and some amount of learning by the tool user. The author has suggested a more brief, but concise, list ten (10) questions which can be used to rapidly elicit the system requirements and expectations from the customer's "need". These requirements can be understood at three (3) abstract levels: first at the system; second, at the product; and, third at the components. The author believes that this is the best, and fastest, way for a team to achieve their understanding of the system concept. This approach employs systems engineering principles, requires use of cross-functional team members, and follows a top-down, hierarchical approach that seeks functions, then form (design), and applies this understanding to the system's architecture (functional, logical, and physical). Interfaces are key points of understanding, because failures most often occur at those interfaces, internal and external. Any team can use this approach, even with a simple tool like a spreadsheet. The author hopes developers and teams in the wide world of product development will use these "10 Golden Questions", and offer feedback and lessons learned on their utility.

REFERENCES

Blanchard & Fabrycky. *Systems Engineering Analysis, 3rd ed.*, 1998.

Buede, Dennis. *The Engineering Design of Systems: Models and Methods*, 2000.

Cohen, Lou. *Quality Function Deployment*, 1995.

Grady, Jeffrey O. *Systems Requirements Analysis*, 1993.

Hooks, Ivy. *Customer Centered Products: Creating Successful Products Through Smart Requirements Management,* 2000.

Leveson, Nancy. *Safeware,* 1995.

Maier & Rechtin. *The Art of Systems Architecting, 2nd ed.*, 2000.

SEI, Carnegie Mellon University. *CMMI: Staged Representation (SE/SW/IPPD), v.1.1*, 2001.

BIOGRAPHY

Dr. Surber is an INCOSE Certified Systems Engineering Professional, and has worked as a pilot, engineer and manager in avionics systems and heavy equipment engineering for over 30 years with three large defense contractors and a large, commercial corporation. He is an experienced senior pilot, flight examiner/ instructor, human factors engineer, systems safety engineer, and military accident investigator. He has accumulated over 5,000 hours of flight and simulator time in 17 types of military and commercial aircraft; holds a pilot rating in single and multi-engine aircraft; and, is a rated parachutist. In 1998 he retired after 29 years of military service with the United States Air Force, and with the Army National Guard and Reserve in various armor, mechanized infantry, aviation and military intelligence units. He is a Principal Systems Engineer for Raytheon, where he works on the V-22 Osprey program, and supports process improvements as a Raytheon Six Sigma Specialist. Dr. Surber has been a member of INCOSE since 1998, and is currently Past- President for the INCOSE Crossroads of America chapter in Region IV.