Projects Don't Begin With Requirements*

Kevin Forsberg
ESEP, INCOSE Pioneer and Fellow
CEO, The Center for Systems Management
1951 Kidwell Drive, Suite 750
Vienna, VA 22182 USA
kforsberg@csm.com

Abstract. Many papers and books emphasize that projects are requirements driven, and until you have a good set of requirements the project should not begin. Twenty years ago an excellent CASE (Computer Aided Systems Engineering) tool was named RDD for "Requirements Driven Development." As one elaborated system requirements and entered flow down decisions to lower levels of the system architecture, traceability of all requirements, upward and downward, was maintained, and the evolving architecture could be viewed in many different ways (FFBD, N^2 diagram, IDEF0 diagram, etc.). What is missing is the traceability back to the user's needs, and more importantly, assurance that all stakeholders have had their needs expressed before the system concept trades have been completed. This paper discusses a number of instances in which the process of converting an idea to system requirements, and then to concepts and specifications, had omitted key user groups. This resulted in important user requirements that were not satisfied, and led to limited use or to project failure. Correct application of the Dual Vee model should minimize the risk of missed user requirements.

Introduction

In 1979 quality guru Philip Crosby boldly stated, "Quality is free!" (Crosby 1979). In his book by that title he advanced the idea that the solution to the quality crisis in North America was the principle of "doing it right the first time." He listed four major principles:

1. The definition of quality is conformance to requirements
2. The system of quality is prevention
3. The performance standard is zero defects
4. The measurement of quality is the price of nonconformance

History has proven that Crosby was and is still right, if the project requirements are complete and correct. Adhering to an incomplete requirement set will seldom lead to user satisfaction.

It is the responsibility of the systems engineer to ensure all relevant stakeholder needs have been identified, agreed to by the stakeholders, the customer, and the developer or contractor. Then the stakeholder needs must be appropriately translated into system requirements. As the project evolves it is typical for users to become aware of new capabilities, and user awareness's soon become new user (mandatory) needs; it is the job of the project manager and the systems engineer to manage these user expectations through an effective change management process.

* Received "Best Paper Award," INCOSE Track, IEEE International Systems Conference, San Diego, CA, 8 April 2010

Twenty years ago an excellent CASE (Computer Aided Systems Engineering) tool was named RDD for “Requirements Driven Development.” The software had a rigorous theoretical basis using behavior diagrams. As one elaborated system requirements and entered decisions to decompose requirements to lower levels of the system architecture, traceability of all requirements, upward and downward, was maintained, and the evolving architecture could be displayed according to the desired view (FFBD, N^2 diagram, IDEF0 diagram, behavior diagram, etc.). At the time concept dependent specifications were defined, the operator was asked also to define the verification plan to ensure that the specifications were properly stated and were verifiable. However, even though the theoretical basis was solid, and there was a growing community of enthusiastic users, the learning curve was steep for new clients, and the vague – but real – requirement for a “user friendly” system (which RDD was not) caused the company to fail.

What is often missing on projects is the traceability of system specifications back to the users’ needs, which is where the project really begins. More importantly, the systems engineers on the project need to provide assurance that all key stakeholders have had their needs expressed and responded to (accept/reject) in accordance with the stakeholder’s power before the system concept trades have been completed. This paper discusses a number of instances in which the process of converting an idea to system requirements and then to a concept had omitted key user groups. This resulted in user needs that were not satisfied, and led to limited use or to project failure.

Projects Begin With User Needs

Projects usually start with a project champion who perceives the need for a modification, enhancement, or totally new system, based on

- New opportunities
- New threat
- Emerging technology
- Paradigm shift
- Competitive pressure
- Political restructuring
- Additional features required for legacy systems
- Performance shortfall for legacy systems
- Business case change for legacy system

The project champion faces great difficulty, however, if he or she does not gain support from “real users” who have a vested interest in project success.

For instance, during the 1960s and 1970s many spacecraft designers praised beryllium as a structural material. It is lighter than aluminum, and stiffer and stronger than high-strength steel. However beryllium is almost as brittle as glass and this makes it difficult to attach to other parts of the structure, and when it is machined the dust is highly toxic. People did overcome the problems, and lived with its limitations. Then two researchers at the Lockheed Palo Alto Research Labs thought of melting aluminum and beryllium together to produce a new alloy, which they called “Lock-alloy.” It is not as stiff or as light as beryllium, but it is much stiffer and lighter than aluminum. Dust from machining Lock-alloy is not toxic and welding or riveting could follow standard practices. The researchers thought the project teams would be overjoyed with the new material. They published papers and attended symposia to tell the world what they had achieved. In fact 20 different companies expressed interest in licensing the material, but no project at Lockheed – where the material was developed – ever used Lock-alloy. Getting user support cannot rely on passive waiting to see if the users will come. And Program Managers, who are challenged daily by show-stopping issues, are not interested in details of new and innovative materials if the materials in use aren't delaying the project. So the Lock-alloy project died a quiet death because there was no user-champion.

Projects or research efforts that are initiated with no user in mind face a very uncertain future indeed. It is often pure chance that will connect an innovative solution to a desperate potential user. For example, the rigidized fibrous silica insulation (designated LI-1500) was the result of a three-year research effort in the Lockheed laboratories (Forsberg 1995). When the principal investigator, Bob Beasley, was told by his manager that his research funding was being terminated, Beasley asked the company to apply for a patent. The corporate legal counsel asked what the material was for and who would use it. Bob said, "I do not know. There is no known user at present." The lawyer said, "Patents cost money. Treat it as a trade secret." So the material was put on the shelf – where it could have stayed for decades – and the research project ended. It was two years before a Lockheed Program Manager, Jack Milton, accidently found out about the material on the shelf, and began selling (ultimately successfully) its use to NASA for the Space Shuttle external insulation. This material, uncovered by chance by a desperate user, in fact became an enabling technology for the Shuttle Program.

Identifying User Needs

The challenge is to understand the customers' operational environment. You must involve all stakeholders in the development process, and you must understand what is really needed. When the customer speaks – Are you really listening? This is much harder than it sounds. You must work with the customer and key stakeholders in prioritizing requirements, including separating the "needs" from the "wants," to arrive at an affordable set of requirements. Melinda Ballou, research director of META group (Ballou 2003), said: "60% to 70% of project failures can be directly tied to poor requirements gathering, analysis, and management."

Stakeholder	Type	Expectation	Priority
Bus line Owner	Product User	• Low price • Low maintenance & operations costs • Resale value	
Environmental Protection Agency	Regulator	• Low emissions	
State	Regulator	• Safety	
Program Office Boss	Executive Management	• Low development cost	
Passengers	Service User	• Handicapped access • Reclining seats • Package storage	
Drivers	Operational User	• Comfortable driver's seat • User friendly controls • High MTBF	
Recreational Vehicle Owner	Future User	• Ease of conversion • Attractive appearance	

Figure 1. Bus Fleet Stakeholder Analysis - Example

In the example in Figure 1 seven stakeholders are identified in a process to capture driving issues from each stakeholder's perspective. Even stakeholders with weak involvement have been included for completeness. While most readers could add another four or five stakeholders, we would be hard-pressed to go beyond that. So I was amazed to learn that when a colleague of mine used this example as an exercise in a systems engineering seminar for practicing professionals, the students came up with over *twenty five* stakeholders, all realistic and defensible. If this level of detail were developed at the start of every project, there certainly would be a reduction in projects that fail because stakeholder needs were missed.

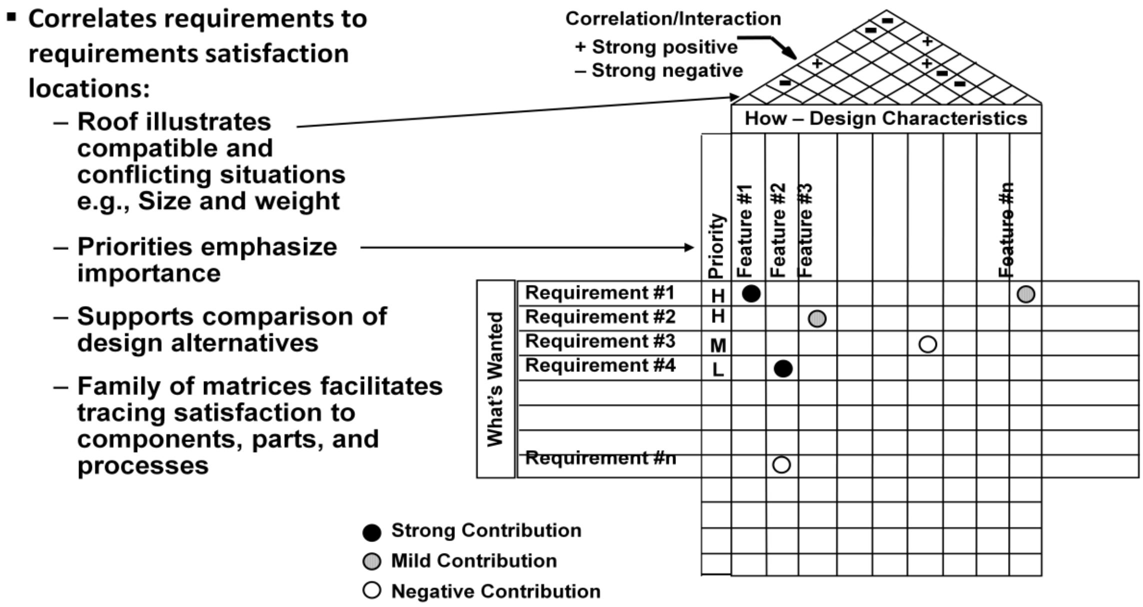

Figure 2 – House of Quality – A Requirements Management Technique (Hauser, Clausing 1988)

The House of Quality (Figure 2) is a technique that has been frequently applied in many industries worldwide during the recent decades. One commercial corporation I consulted with decided to consolidate their administrative and scheduling operations for their worldwide sea and land delivery operations. They used the House of Quality as a means to plan this restructuring, and they thought that three or four months developing their House of Quality would be more than enough to initiate the 24-month modernization project since all of their users were internal to the corporation. They found the House of Quality process forced people to talk with one another, and as the discussions evolved, the true user needs emerged. So the initial plans were scrapped, and the teams spent most of the 24 months using the house of quality to derive a streamlined approach to scheduling, statusing, and controlling the combined land-sea-land transportation. This operational program was then implemented in less than six months. The real benefit, the project manager told me, was that they had all the user requirements up front, and they were able to break down old internal department barriers that the original approach would have left in place.

The most important phase in the project cycle is the User Requirements Analysis and Agreement phase (Figure 3), where off-core studies probe to identify and assess critical issues that will be encountered in satisfying the user needs.

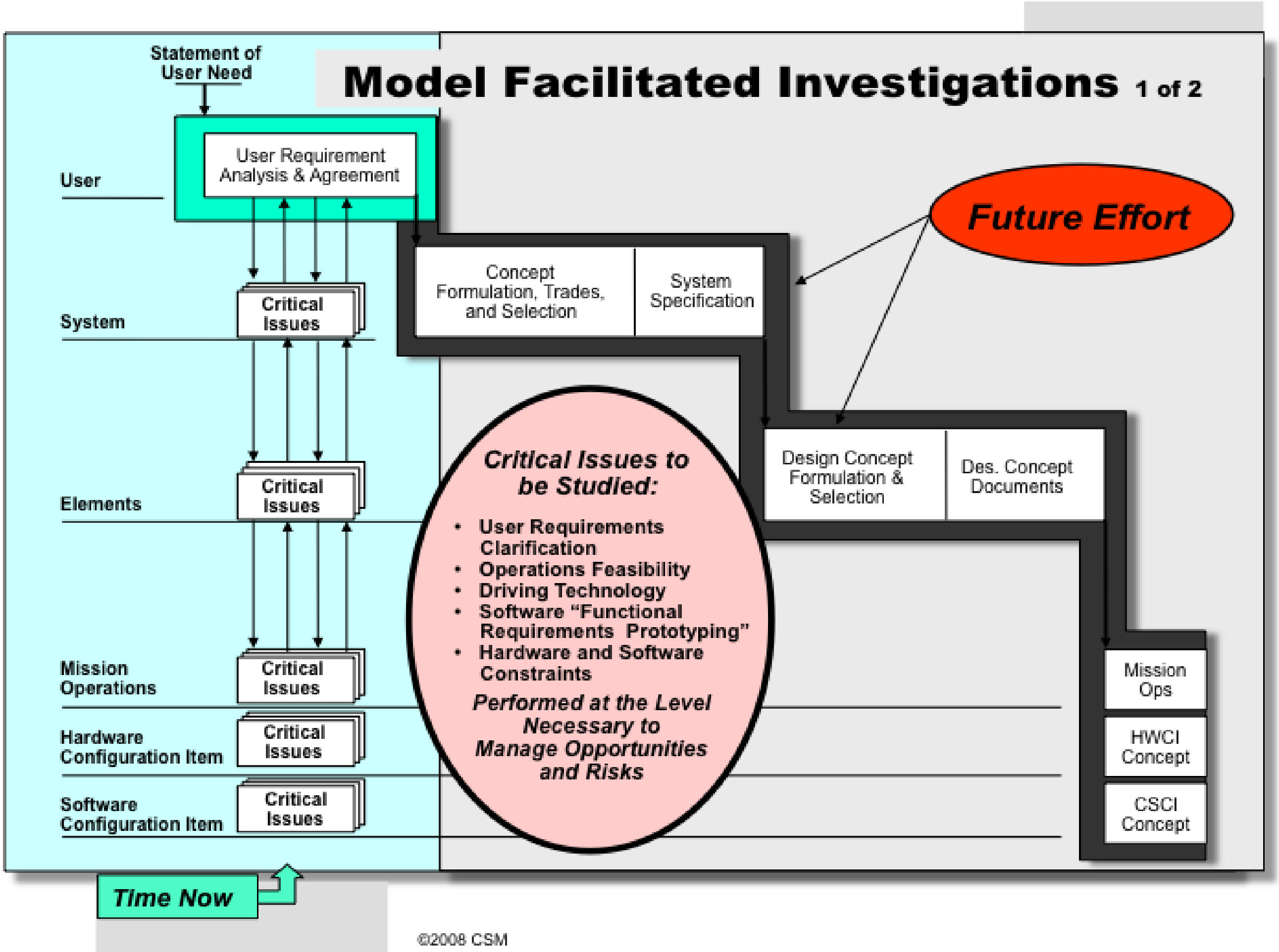

Figure 3 – The Vee model in early User Requirements Development (Forsberg, Mooz, Cotterman 2005)

The phase shown in Figure 3 encompasses what is often called early phase systems engineering (pre-Milestone A), and it is critical to broaden the stakeholder base at this point. A recent study by the National Research Council (2008) focused on reducing the development time for new US Air Force projects, and the report notes: "Simply stated, Systems Engineering is the translation of a user's needs into a definition of a system and its architecture through an iterative process that results in an effective system design." The iterative involvement with stakeholders is critical to the project success.

Stakeholder input is vital during evolution of any program, and this is emphasized in Figure 4, where the focus of the effort here is on concept formulation and description. The upward iterations with the system specification and with the user requirements are encouraged, but previous decisions are under change management. "Change management" does not mean "change prevention," but a disciplined process must be in place.

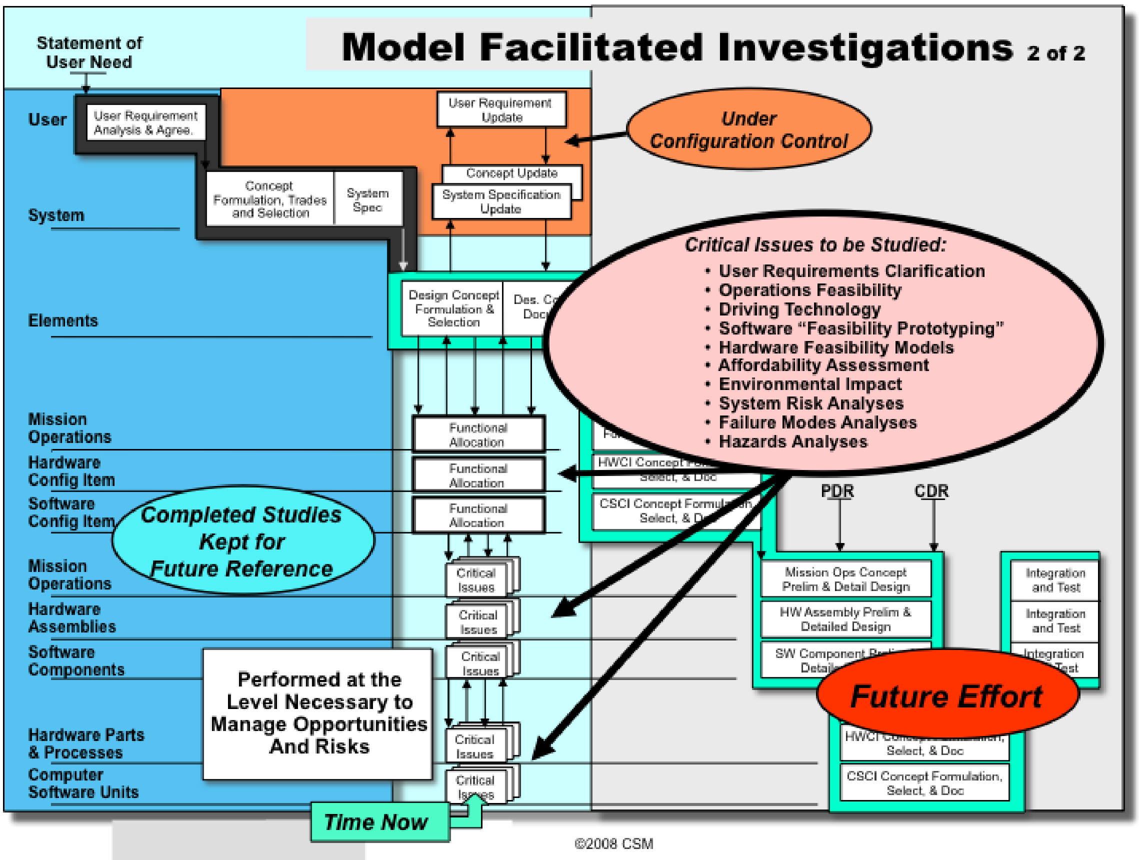

Figure 4 – The Vee model in Design Concept Development (Forsberg, et al. 2005)

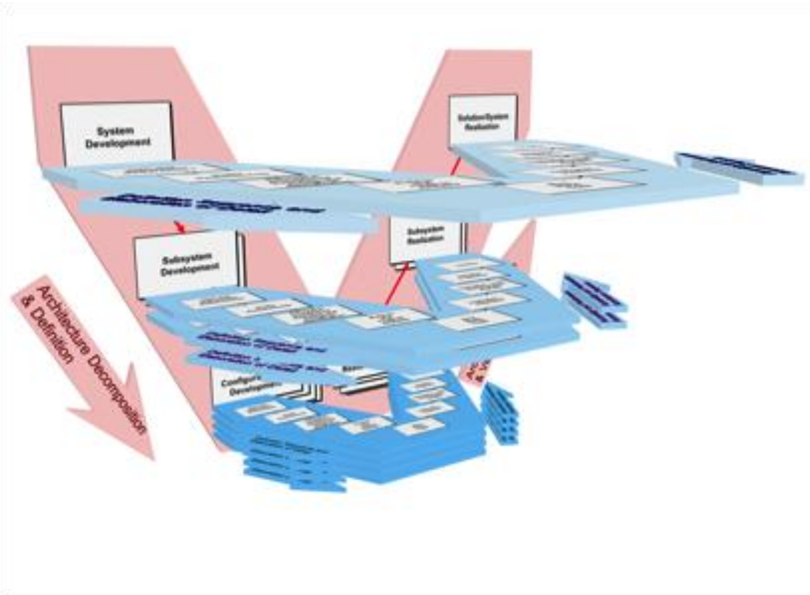

Figure 5 – The Dual Vee model (Forsberg, et al. 2005)

The Dual Vee depicts concurrent architecture and entity development. There is one architecture Vee for the system and it is shown as the vertical orange-color Vee in Figure 5. There is one entity Vee for each entity within the architecture (the blue "horizontal" Vees in

Figure 5). The architecture Vee depicts the architecture baseline evolution. The entity-Vees depict baseline evolution for each entity.

The Dual Vee emphasizes the multi-layer characteristic of entity development at every level in the development process. It is important to recognize that there are users and stakeholders at every level of decomposition and their management is key to total system success.

Figure 6 illustrates the situation for a battery developer or supplier who must satisfy internal and external users of electric power. At every level there are users above you who are external to your project, and who are the ultimate users of your product. There are also internal users one level above you, who define the requirements you must respond to, and there are associate users at your same level who must interface with your output. As an example consider the power system on a mobile phone. The battery developer, working in an entity Vee at the lowest configuration item level, as shown in Figures 5 and 6, must respond to the needs of the Power System manager, who defines the electrical requirements for the battery (voltage, current, heat output limits, recharging time, battery capacity, etc.). The Power System manager is working on an entity Vee at the subsystem level in Figures 5 and 6. Associate users at the Lowest Configuration Item level are identified in Figure 6. The style engineer will define marketing-dictated bounds of battery shape, weight, and dimensional restrictions.

The systems engineering function exists even at the Lowest Configuration Item level, to ensure that all user needs have been identified and conflicts are resolved or pushed upward if open issues persist.

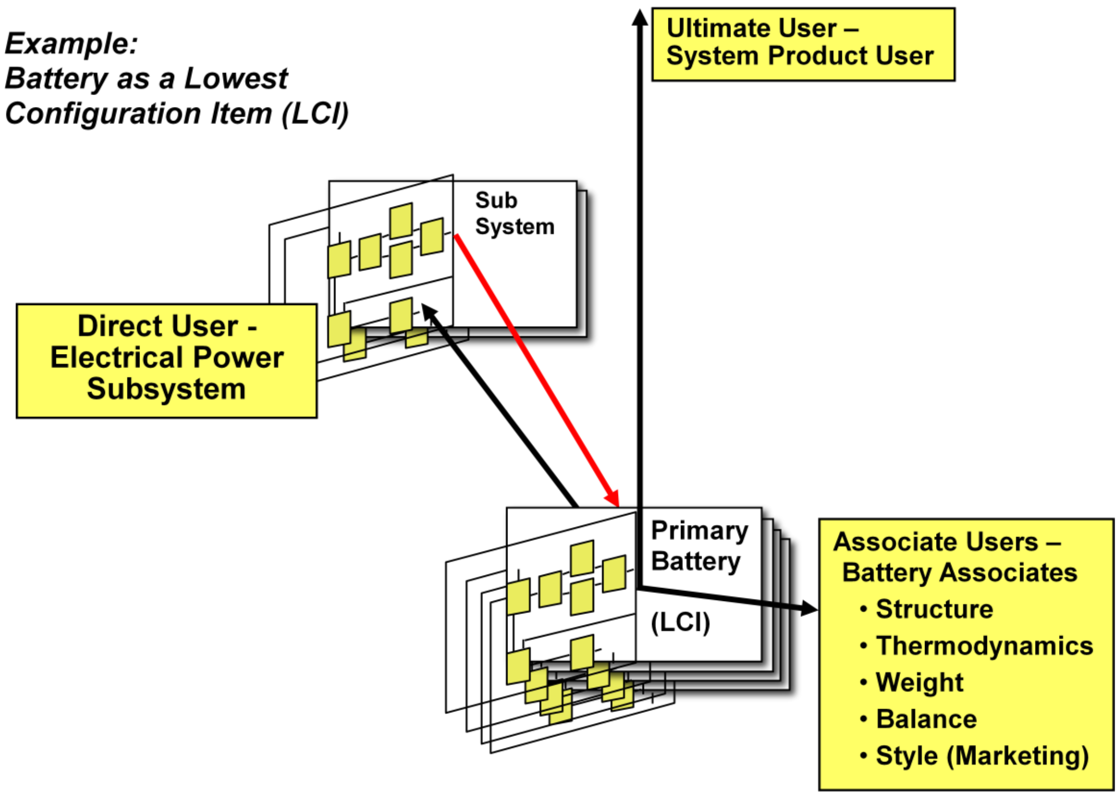

Figure 6. Three Categories Of Users Must Be Satisfied At Every Level.
Situation depicted is for a Lowest Configuration Item (LCI).

Case Studies On User Involvement

Most people are aware of the need for user involvement, but as the following six cases illustrate, this is not a trivial issue in practice, and it is often overlooked.

1. ***Lunch Bucket – Missed Internal Users*****.** In the mid-1980s an innovative engineer at a food processing and packaging company recognized that many people like hot food for lunch, but they do not want to take the time to go out for a meal. He envisioned putting soup or stew into a single serving container which could then be put into a microwave oven at work to provide a hot lunch. He recognized the need to get customer approval (i.e., test trials with people from offices who bring their lunch to work), and of course approval from the Food and Drug Administration (FDA). At the time the project was started there were no competitors visible, and with an aggressive approach his company could capture a major market share in this highly competitive commercial environment.

   The problem was that he and his team viewed the challenge as creating a bucket of food for a customer; stakeholders such as manufacturing and transportation were ignored initially. When the internal stakeholders such as manufacturing were brought into the process, each was treated in serial fashion. The product was ultimately successfully completed, but only after eight years of serial development. It took longer to bring this product to market than it took to design and build the Golden Gate bridge in San Francisco; it took longer to bring lunch bucket to market than it took to design, fabricate, assemble, test, and successfully launch America's first reconnaissance satellite in the late-1950s. It was a business failure because competition arrived first.

2. ***AutoCon – Missed key user*****.** A very bright and aggressive project manager (I will call her Mary, but that's not her real name) was delighted to take on a challenge of adapting a commercially available software package to meet the needs of her agency for automatic generation of terms and conditions (Ts & Cs) for all types of contracts her agency awarded to contractors large and small. After four previous failures over the preceding decade, Mary was careful to create a team with representatives from all contracts organizations, other stakeholders within her organization (such as finance), and even representatives from the contractor community who would be on the receiving end of the contract Ts & Cs. Her team developed a detailed Concept of Operations document and she had all the user and stakeholder bases covered.

   Her organization's software was based on Windows NT, so Mary's team made sure that the acceptable vendors had a program that ran in the Windows NT environment. To prove this each vendor was asked to provide a demonstration of his or her software in action. But because the development team had many individual war stories of positive demos yielding dead-on-arrival operational systems, the team placed highest weight on visiting other customers of each vendor, and then viewing the system in action at that site. The selected software passed these tests with flying colors

   The site visits to customers using the selected vendor's software were a great success, showing the program functioning flawlessly and with impressive speed. Unfortunately Mary and her team missed a small point: the software they witnessed was written in UNIX. What they needed (and what they bought) was a Windows NT product, which turned out to be an alpha version, so it had not even completed the vendor's verification testing. Mary's management was very unhappy with another sixmonth delay while the vendor completed his alpha and beta testing and finally delivered an operating system.

Mary survived the software delivery delay. What got her fired was a request from her manager for the status of contracts in work. Mary overlooked the fact that her manager was also a user, and he had several specific reports he needed – and Mary could not supply them, nor could anyone else because that user requirement had not been given to the vendor.

3. ***Aquila Remotely Piloted Vehicle (RPV) – No operational input in the initial development***. In the late 1970s (when UAVs were called RPVs) several contractors worked with a US Army development office to create cheap, throwaway remotely piloted vehicles that could provide tactical battlefield surveillance. The concept one vendor offered was a lightweight wood and aluminum airframe, with thin fiberglass covering. The engine would be a McCullough chain-saw motor, selected because it was inexpensive, widely available, rugged design, and proven to operate in a wide range of weather (from zero ºF to 140 ºF).

   Since the system concept was based on the need for lowest possible cost, the Army development office decided that the RPV would be a cheap throwaway airframe with low-resolution instrumentation ($20 K/ vehicle). Thus no landing gear or other capture mechanism would be needed. In fact, landing systems would add weight and cost, and so had negative value to the program. The buyer and seller teams were happy with the first flight test series, and the RPV performed as designed. No users outside the RPV program showed interest in those early days of the program.

   After successful flight-testing, the Army development office decided to bring in the operational commands that would be responsible for RPV deployment in combat. They were very impressed with the *Aquila*, but were very disappointed with the poor resolution from the cheap electronics. They wanted higher quality images. An upgrade of the electronics was successful, but new instrumentation cost $200 K per vehicle, and it was no longer acceptable to trash every RPV. Attempts to capture each vehicle in a net failed, adding landing gear forced redesign of the airframe, and the cheap engine was no longer powerful enough. The initial needs assessment by the development command missed key user requirements and this led to the failed program.

4. ***Hubble Space Telescope – Who are the users?*** The idea for the Hubble started with reorienting a reconnaissance satellite 180 degrees, to look at stars, instead of the earth. A small team ("gang of five," which included Maxwell Hunter, a colleague of the author) expended much effort to define objectives for the system (look into deep space, seeking the origin of the universe). Lockheed Missiles and Space Company, TRW, and NASA worked closely with the Congressional staff to create an affordable program, which ultimately has been very successful. This program was the next major project for NASA as the Shuttle development was nearing completion and transition to an operational stage.

   As the Hubble program was nearing completion, NASA helped establish the Space Telescope Institute (STI), which would represent the worldwide community of astronomers, to provide them the access to this observatory in the sky. The astronomers and system operators, however, viewed themselves as users, and they felt they had been left out of the concept definition phase. There was sufficient bitterness that a senior scientist, Eric Chaisson, at the STI wrote a book titled, *The Hubble Wars: Where Astrophysics Meets Astro-politics* (Chassion 1994).

5. ***The U-2, the TR-1A, and the helicopter-mounted SOTAS – Where dissimilar projects join.*** A few years ago the US Army developed a prototype system to provide tactical battlefield surveillance, with direct feed back to the tactical commanders on the ground. This system, designated SOTAS, consisted of a helicopter and a high-power side-looking radar antenna mounted under the helicopter. Several prototypes were built and tested, and they met all expectations. The Army included the SOTAS project in their budget request for the next year's funding.

   The Air Force asked the Army to cancel SOTAS, and move the mission to the TR-1A (a version of the high-flying U-2), because the high-flying TR-1A could see more of the battle-field and therefore could give more valuable tactical information to field commanders. Those SOTAS missions, combined with the Air Force strategic missions for battlefield surveillance, would give added justification to the total program, and ensure that congress would approve the funding. And so it came to pass.

   However, unlike other Air Force programs, the TR-1 program incurred overruns and schedule slippage. So to put the program back on track and within the limits of congressional funding, a reduction in requirements was authorized. To the dismay of the Army, by eliminating all the SOTAS-related requirements (i.e., dropping the tactical missions supporting ground commanders) the program funding was balanced – at the expense of a key user community.

   The user need was still there in the Army tactical commands, but now the Army had to start from scratch to rebuild its program team – including the former SOTAS users.

6. ***The Dual Vee view – Nokia mobile phone batteries and the missing user group.*** No one can deny the amazing success of Nokia over the past 18 years since they introduced the GSM mobile telephone system in 1992. Their market share is still over 35%. So it is interesting to consider the wealth of mobile phone handset models currently available.

   Nokia uses the Vee model in planning its work, including effort at the lowest configuration item level. By using common component development much design work and basic build can be accomplished before the designs diverge to satisfy unique project needs, as shown in the entity vees in Figure 7.

   And Nokia has been incredibly successful in anticipating market trends and providing mobile phones to meet evolving customer desires. But it is clear that they have given low priority to one user category: ***logistics***. At present there are over 108 Nokia mobile phone models available on the web (Nokia 2010). While many of these phones look similar, there are 45 different batteries required to support these phones. (See Table 1.) Only 9 batteries support more than two handset models. While this apparently makes good marketing sense, it is overwhelming to retailers who must stock the variety of batteries for the handsets they sell. And in fact the sales outlets do a poor job of support in this area, leaving loyal customers frustrated. This is a case of a user ignored or forgotten. It appears that Nokia is not alone since other manufacturers (Motorola, Samsung, etc.) do no better.

   When one considers how many electronic items are in our lives today, and how many devices are able to use one of a few commercially available batteries, it is surprising that the mobile phone industry cannot do the same with five or ten standard batteries.

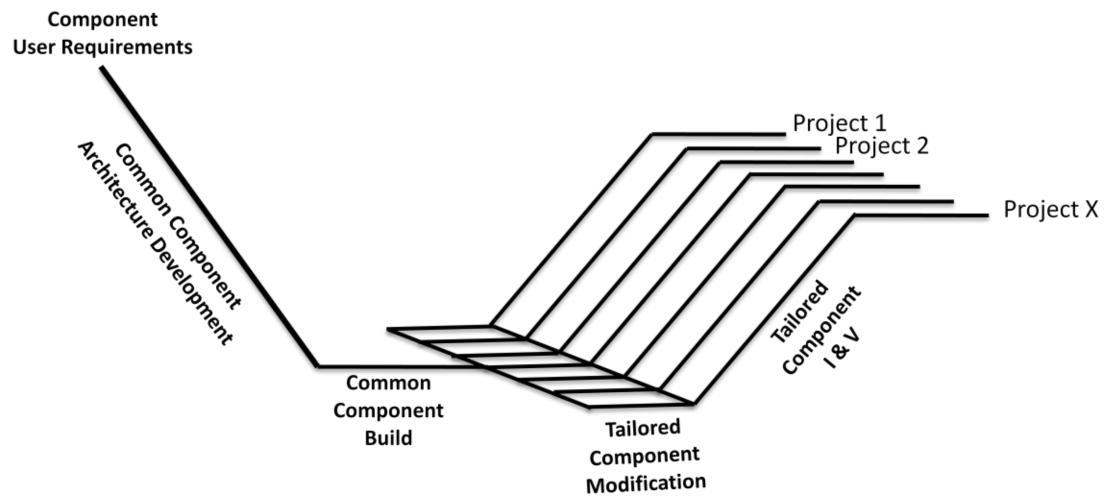

Figure 7. Common Component Development to support multiple end products.

Conclusions

It is obvious to almost everyone that input from the user community is essential at the start of a project. What is not so obvious is that the interaction with the user must be maintained throughout the project development, and beyond into operations. Equally difficult is ensuring that the "user net" is cast widely so that no influential stakeholders are ignored.

It is the systems engineer's responsibility to ensure that all key stakeholders are included in (or represented on) the project, and then the SE is responsible for managing the stakeholder expectations to stakeholder satisfaction.

References

Ballou, Melinda. 2003. Research Director of META Group

Chaisson, Eric. 1994, 1998. *The Hubble Wars: Astrophysics Meets Astro-politics…* HarperCollins Publishers, Inc.

Crosby, Philip B. 1979. *Quality Is Free*. New York: McGraw Hill.

Forsberg, K. 1995. "If I Could Do that, Then I Could…" NCOSE Fifth International Symposium, (St. Louis, Mo.) Seattle: INCOSE.

———, H. Mooz, H. Cotterman. 2005. *Visualizing Project Management*, 3rd edition. New York: J. Wiley & Sons. pp. 242, 244, 350.

Hauser, John R. and Don Clausing. May 1988. "The House of Quality." Harvard Business Review pp 63-73.

National Research Council (Part of the National Academy of Engineering). 2008. Report, "Pre-Milestone A and Early Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Acquisition."

Nokia Mobile Phones. 2010. http://www.1800mobiles.com/nokia.html

Biography

Dr. Forsberg has 27 years of industrial experience in systems engineering and project management. His experience ranges from research projects, to development efforts, to full-scale production implementation. Since 1983 he has provided training and consulting to both government and commercial international clients, specializing in systems, hardware and software project management, and the related processes, techniques, and skills essential to achieving predictable project performance.

M.S. and Ph.D., Engineering Mechanics, Stanford University (1958, 1961)
B.S., Civil Engineering, Massachusetts Institute of Technology (1956)
Stanford Executive Sloan Program (1979)
Fellow, ASME (1973)
Pioneer, INCOSE (2001)
Fellow, INCOSE (2006)
Certified Systems Engineering Professional (CSEP) (2004)
Expert Systems Engineering Professional (ESEP) (2010)