

Delivering Quality Products That Meet Customer Expectations

Louis S. Wheatcraft
Compliance Automation, Inc.

Why is it so difficult for project personnel to deliver a quality product on time and on budget that meets or exceeds their customer's expectations? A major contributor to project failure is neglecting to spend time at the beginning of the project on the basics. There are critical activities that must be accomplished and agreed to before writing requirements and beginning product development. These activities include clearly defining the project and product scope, including need, goals, objectives, drivers and constraints, assumptions, operational concepts, external interfaces, and feasibility and risk assessments. Unfortunately, many of these activities are often skipped. Developers jump into design without really understanding the reason for developing the product, and what it is they are supposed to do. This article focuses on one of the biggest problems in clearly defining your product scope: a failure to define and agree to the product's need, goals, and objectives.

Past articles in *CrossTalk* have cited dismal results for studies of software project failures and successes. In the April 2002 issue, Theron Leishman and Dr. David A. Cook reported the following facts regarding the Department of Defense software development:

At the 5th Annual Joint Aerospace Weapons Systems Support, Sensors, and Simulation Symposium in 1999, the results of a study of 1995 Department of Defense (DoD) software spending were presented. Of \$35.7 billion spent by the DoD for software, only 2 percent of the software could be used as delivered. The vast majority, 75 percent, of the software was either never used or was cancelled prior to delivery. The remaining 23 percent of the software was used following modification. [1]

Similarly in a July 1998 *CrossTalk* article, Lorin J. May reported on a Standish Group study of unsuccessful government software:

According to the Standish Group, in 1995, U.S. government and businesses spent approximately \$81 billion on cancelled software projects, and another \$59 billion for budget overruns. Their survey claimed that in the United States, only about one-sixth of all projects were completed on time and within budget, nearly one third of all projects were cancelled outright, and well over half were considered "challenged." Of the challenged or cancelled projects, the average project was 189 percent over budget, 222 percent behind schedule, and contained only 61 percent of the originally specified features. [2]

One problem that contributes to cancelled projects is a failure to establish a shared vision of the final product at the beginning of the project. Having a shared vision requires defining the product's need, goals, and objectives before writing

"An error made by some project teams is the failure to differentiate the product scope from the project scope. Each defines overlapping but different need, goals, objectives, stakeholders, drivers, and interfaces."

requirements and developing code. The purpose of new software or of an upgrade must be clearly understood before writing requirements, or divergent requirements will be written and important requirements will be missed. This vision must be provided to programmers before development to ensure that they maintain a single viewpoint.

Creating a shared vision is a basic concept in product development that is often neglected. There is an old saying that expresses this: "Failure to plan, is a plan to fail."

Project vs. Product

For the purposes of this article, I define a *project* or *program* as a system that consists of the people, processes, and tools that make up the environment in which a product or products will be developed. The product could be a hardware system,

a hardware/software system, a process, or a service. The product is a separate system within the project or program system. Each may have its own scope.

An error made by some project teams is the failure to differentiate the product scope from the project scope. Each defines overlapping but different needs, goals, objectives, stakeholders, drivers, and interfaces. While the product will be driven by the project scope, there will be other drivers as well. The perspective is different when focusing on developing the product vs. managing the project or program. For large programs or projects, there typically will be a project manager as well as a product manager. The product manager may be referred to as the system engineer, the lead engineer, the lead software engineer, etc. His/her focus is on product management vs. project management.

This article deals primarily with the product. The following sections discuss the problem in clearly defining the product scope identified earlier – a failure to define and agree to the product need, goals, and objectives.

Clearly Define Product Scope

The scope of a product constitutes the vision: the need to develop or procure a product or service; the goals and objectives of the customer and your company; information about the customers and users; and how the product will be developed or purchased, tested, deployed and used [3, 4, 5]. The scope must unequivocally define the product boundaries. A product with no boundaries will diverge, or as Yogi Berra said: "If you don't know where you're going, you'll probably end up someplace else."

For example, you are assigned to lead a project team to upgrade a product. The main reasons for the upgrade are to fix known problems, to address lessons

| Product (Implementation) | Need |
|---|---|
| New remote-piloted aircraft. | To provide low risk, accurate, real-time situational awareness of battlefield operations. |
| New software standard message protocol. | To enable a fighting force linked together as an integrated collection of interoperable systems. |
| New integrated intelligence software system that will collect and integrate terrorist information from multiple sources. | To quickly and accurately assess potential terrorist threats. |
| Integrated acquisition system. | To reduce the integration and maintenance overhead associated with current non-integrated, distributed systems. |
| A proactive tool provided to officers that ensures the accuracy of their promotion data prior to its submission to the primary promotion board. | To decrease the cost and use of resources in support of military promotion boards. |

Table 1: Examples of Defining a Product vs. Defining the Real Product Need

learned, and to improve its operability, reliability, security, safety, and maintainability. The functionality of the current product is adequate to meet the basic need for the product; however, there are stakeholders that would like to add to current functionality and improve product performance.

While some stakeholders want to add a few *bells and whistles*, others have legitimate reasons to add to the product's functionality. These reasons include upgrading to new technology, making the product more competitive, making the stakeholder's jobs easier or more effective, and adding features based on changes to the product's operating environment. If you skip the up-front activities and fail to define and get an agreement on the scope of the product upgrade, there will be no clear boundaries of what is included in the upgrade and what is not. You are doomed to failure.

Getting agreement from all key stakeholders of the product scope before your team writes requirements and begins design ensures that everyone will clearly understand the requirements' boundaries for the upgrade. Involving key stakeholders in scope definition will avoid battles that result from differing visions and different interpretations of what should be included or excluded in the product. Issues can be identified and resolved before investing scarce resources into the requirements writing effort. Spending time to resolve issues, get questions answered, reduce debates, and confirm assumptions will result in reducing the time to write requirements, as well as speeding up the requirements review and baseline process.

Scope definition keeps requirements writers from diverging, reduces requirements inconsistencies, and keeps the big picture in view. An agreed-to product scope contributes to better requirements whose impact on development and testing is to avoid incorrect design and to reduce requirements discrepancies found

in testing. Having a clear product scope will allow ground rules to be established.

In the preceding example, your first priority is clearly to maintain current functionality while specifically addressing the known problems and lessons learned as well as improving the current product's operability, reliability, security, safety, and maintainability. Requirements to add additional functionality or features will only be considered if specifically agreed to and documented; if they do not impact the development schedule and budget; and if they do not conflict with your efforts to improve the product's operability, reliability, security, safety, and maintainability.

Once product scope has been defined and agreed to by all key stakeholders, it must be formally baselined and controlled. Managing change to avoid *scope creep* is one of the biggest problems in government and industry. Too often, the scope of the software development project changes in midstream as stakeholders think of new features to add. One reason given for project cost or schedule overruns is a change in the agreed-to product scope without a corresponding change in cost or schedule. Change is inevitable. However, if new features are requested then the scope, which includes the cost and schedule, must be adjusted accordingly. Change is not free. Key stakeholders are often ignored until far too late in the process. By involving key stakeholders in product scope development and baseline, you will be able to minimize change. The ultimate payoff is reduced rework, reduced cost overruns, and reduced schedule slips.

Defining Need, Goals, and Objectives

The foundation of a product scope is having a clear understanding of the product's need, goals, and objectives. You do not want anyone on your team saying: "Why are we doing this?" You do not want everyone on your team having

a different vision of what they are to develop.

In defining product scope, one problem you will frequently face is understanding product need, goals, and objectives and their relationships.

Product Need

It is the *need* that typically initiates a project. The need forms the basis of the project: The product is being developed to fulfill the identified need. The product need may come because of a new threat or opportunity, a mission or business need, a customer request, a technological advance, a deficiency in an existing product, or a legal requirement. The need may require a new product or an upgrade to an existing product. The product need is the driving purpose for designing and building or upgrading the product.

The need is why you are doing the work. For example, say I am going to build a house. Why? I need to protect my family. I am protecting my family from a number of things: the natural environment, persons or critters that might want to harm us, or persons who may want to steal our possessions. The basic need is to protect my family.

A common error when identifying the project need is to begin by stating an implementation (naming a product) rather than stating the underlying need for the project. The product is not the need; assumptions about need may be conflicting or just plain wrong. There are often several ways a need can be met. As an example, consider this statement: "We need a drill bit." What we *really* need is a hole. It may be that a drill bit is the best solution to obtaining the hole, but it is the hole that is needed.

One prime method to uncover the real need is to ask "Why?" If someone says they need a drill bit, ask "Why?" In this case, a drill bit is *not* needed, the hole is. If someone says they need a remote-piloted aircraft, ask "Why?" The real reason may be that they need to provide low-risk, accurate, real-time situational awareness of battlefield operations. In both cases, the drill bit and the remote-piloted aircraft are one of several solutions that will meet the need. A product development effort should be *need-oriented* and should *not* seek to justify a specific solution or acquisition program (see Table 1).

Identifying the problem the product is to solve is one of the best approaches to determining the real need. This point is clearly made by Robert Frosch, senior research fellow, Kennedy School of Government, Harvard University and

former NASA administrator:

In my work here, my most frequent refrain is: "What problem is it you're trying to solve? What's the underlying question?" In my work on the Reports Review Committee of the National Academies, I have found that the most common problems faced by an advisory report at the end are generated by lack of clarity in setting the scope of the question at the beginning. [6]

For your project, you should be able to clearly define the specific problem the product is to address and demonstrate your understanding of the problem. Knowing and understanding the problem will enable you to explain why the project is worth doing, why the product is needed, and why the product is important to the customer.

In solving the problem, you can ask any or all of these questions: What is it you cannot do with the current system? What are the limitations that will prevent you from meeting the projected mission need? What is the key capability (benefits) that having this product will provide for your organization, the customer, or the product users? What will you be able to do differently if you have this product (in terms of improved productivity, operational effectiveness, or efficiency)? What will you not be able to do if you do not have this product? What are the consequences of not having this product (impact on customer/sponsor and/or users to perform mission responsibilities if the capability shortfall is not resolved, or impact of lost technological opportunity in terms of cost to your organization)? The answers to these questions can help you identify the real need for your product.

The process of answering these questions in the government arena for large projects is often accomplished during an activity called the *need assessment*. The outcome is a problem statement that summarizes the analyses and conclusions of the assessment. The resulting problem statement is referred to by various names: statement of need, need statement, or mission need statement [7 (a template for this process can be found here)]. From this problem statement, the need for the product can be identified, as shown in Table 2.

The need should be a short and concise statement. Once derived, the need should not change over time. If the need

is changing, you do not know what is really needed, and you cannot build a product to meet a moving target. Do not let the real need be forgotten. It is the focus of your investment.

Once the need for the product is captured, identify its goals and objectives. When in a group, ask what the need is and you will often find a variety of needs stated. In some cases, this is due to a lack of agreement on what the need is. In other cases, they are listing goals for the product. If you find yourself creating a long list of needs, you are probably mixing goals and objectives with the true need. The challenge is to differentiate between the need and the goals and objectives.

Notice that I am advocating determining a *single* need. There might be times when a product will meet more than one need, but too many needs, like too many cooks, spoils the broth. The attempt to define a product to support multiple needs often results in a product no one can build, no one will buy, or no one can use. It does not mean your product does not have multiple features. Expanding that need into goals and objectives, developing operational concepts, and reviewing with stakeholders will result in many features that will become requirements to meet the need. Do not confuse features with the need.

Goals

In developing a product, ask what you hope to accomplish in meeting the need. *Goals* are the end toward which your efforts are directed. Each goal is tied to a part of the process in meeting the need. Goals are general responses to the need statement. Goals translate the need into a given solution to the problem. What will the project accomplish to affect the problem and meet the need?

There may be more than one way to meet a need. The goals you document will differentiate what you are going to accomplish vs. some other implementation.

Objectives

Objectives expand on the goals and state, in measurable terms, what you are trying to achieve. Objectives state the customer's expectations for performance. Objectives can include quality, new capabilities, needed functionality, etc. Objectives address these questions: In order to accomplish each goal, what specifically are you going to do? How will you know if you succeeded? What results do you expect? They may include cost and schedule objectives inherited from the parent project or program.

Both goals and objectives are short declarative sentences; goals are rather broad, and objectives fall under each goal

Table 2: *Examples of Problem Statement, Need, Goals, and Objectives*

| Military Promotion Board Information Availability | Military Logistics Command [8] |
|--|--|
| <p>Problem: There is no easy way for officers up for promotion to review the information contained in their promotion files. Because of this, there have been too many requests for supplemental promotion boards due to incorrect data in officers' promotion files used by the primary promotion board. This puts a burden on our personnel and promotion board workers.</p> | <p>Problem: Equipment is maintained at logistic centers that are distributed around the world. Command instructions specify how the data concerning the tasks and training of skilled mechanics and technicians doing this work will be managed. Currently, each logistic center has unique processes and software tools to do this. The costs of maintaining these individual software tools have become excessive. Command has dictated that the process and tools be standardized.</p> |
| <p>Need: Decrease the cost and use of resources in support of military promotion boards.</p> | <p>Need: Reduce the life-cycle costs of the command logistics process.</p> |
| <p>Goals:</p> <ul style="list-style-type: none"> • Put the responsibility for checking officer promotion data in the hands of the officers being considered for promotion. • Provide officers a proactive tool to ensure the accuracy of their promotion data prior to the primary promotion board. • Eliminate excessive workload for the personnel office and promotion board personnel. | <p>Goals:</p> <ul style="list-style-type: none"> • Reduce the costs of maintaining multiple logistic systems with a standard, command-wide software system to be used at all logistic centers. • Provide an integrated training and tracking system for all logistics personnel. |
| <p>Objectives:</p> <ul style="list-style-type: none"> • Provide 24/7/365 access to members' official personnel records. • Provide correct display of data in officers' promotion files. • Remove personnel office workers as the middleman. • Provide up-to-date information to officers on corrective actions when errors are found. • Give officers correct and current promotion products. • Lower supplemental promotion board requests from 75 per board to 40. • Prevent fraud. | <p>Objectives:</p> <ul style="list-style-type: none"> • Use state-of-the-art computing, networking, and software technologies. • Deploy the system as a client-server system. • Allow supervisors to manage skills and training data in accordance with Command Instruction 21-108, Production Acceptance Certification. • Support the procedures to maintain Air Force Materiel Command Form 75, Job Knowledge-Training Certification Standard. • Interface to a command-wide training management system. • Develop a module to collect data for process improvement evaluation, also required by Command Instruction 21-108. |

COMING EVENTS

February 10-13

*Commercialization of Military and
Space Electronics Conference*

Los Angeles, CA

www.cti-us.com/ucmsemain.htm

February 24-27

*Software Engineering Process
Group Conference*



Boston, MA

www.sei.cmu.edu/sepg/

February 25-26

*Data Mining Technology for Military
and Government Applications Forum*

Washington, D.C.

www.worldrg.com

March 24-28

*International Symposium on
Integrated Network Management*

Colorado Springs, CO

www.im2003.org

April 8-10

FOSE 2003

(Federal Office Systems Exposition)

Washington, D.C.

www.fose.com

April 28-May 1

Software Technology Conference 2003



Salt Lake City, UT

www.stc-online.org

May 3-10

*International Conference on
Software Engineering*

Portland, OR

www.icse-conferences.org/2003

May 6-8

TechNet 2003

Washington, D.C.

www.technet2003.org

and are somewhat more specific.

It is common for people to confuse goals and objectives. Sometimes one or the other is defined without making a distinction. Sometimes the words are used together: *goals and objectives*. The important thing is that these topics and their content are addressed up front. One approach that can be used to differentiate goals and objectives is to think of a top-down approach. You start with defining the need. Then, you list goals. For each goal, ask the question "Why?" The answer should be the need. Then for each goal, list your objectives. Each objective should be traceable to one or more goals.

Once you have a draft of the product need, goals, and objectives, you must communicate them with the major stakeholders and get their agreement. The need, goals, and objectives form the foundation of your project's scope. Once you have a better understanding of the product, goals and objectives may change – but changes need to be closely managed. A change to goals or objectives is a change to the scope, which could impact cost, schedule, and risk.

Conclusion

To deliver a quality product, on time and on budget that meets your customer's expectations, get back to the basics and define, communicate, and get agreement on a clear vision for the product. To establish this vision, spend the time at the beginning of the project accomplishing and getting agreement on critical activities before writing requirements and beginning product development. These activities include clearly defining the project and product scope, including need, goals, objectives, drivers and constraints, assumptions, operational concepts, external interfaces, and feasibility and risk assessments. Going back to the basics of product management and establishment, getting buy-in, and communicating a clear vision for the project will lead toward a successful project. Anything less will compromise schedule, budget, quality, and mission success. ♦

References

1. Leishman, Theron R., and David A. Cook. "Requirements Risks Can Drown Software Projects." *CrossTalk* Apr. 2002 <www.stsc.hill.af.mil/CrossTalk/2002/apr/leishman.asp>.
2. May, Lorin J. "Major Causes of Software Project Failures." *CrossTalk* July 1998 <www.stsc.hill.af.mil/crosstalk/1998/jul/causes.pdf>.
3. Hooks, Ivy F., and Kristin A. Farry.

Customer-Centered Products: Creating Successful Products Through Smart Requirements Management. New York: Amacom 11 Sept. 2001.

4. Wheatcraft, Louis S., and Ivy F. Hooks. "Scope Magic." Nov. 2001 <www.complianceautomation.com/papers/scope_magic.pdf>.
5. Wheatcraft, Louis S. "The Importance of Scope Definition Prior to Developing Space System Requirements." *INCOSE INSIGHT*, Jan. 2002: 4:4 <www.complianceautomation.com/papers/scopedefincoseinsight.pdf>.
6. Robert Frosch. "Various." E-mail to Ivy F. Hooks. 6 Mar. 2002.
7. Federal Aviation Administration. "Acquisition System: FAA Mission Need Statement – Aids and Tools." <<http://fast.faa.gov/archive/v0800/flowcharts/maflow/drifmnsaids.htm>>.
8. Lynch, K. Edward. "AFMC Production Acceptance Certification and Depot Maintenance Quality Software Project." *CrossTalk* May 1998 <www.stsc.hill.af.mil/crosstalk/1998/may/afmc.asp>.

About the Author



Louis S. Wheatcraft

has more than 34 years experience in the aerospace industry, including 22 years in the U.S. Air Force. During the last two years, Wheatcraft has worked for Compliance Automation where he conducts seminars on defining product scope, writing good requirements, and requirement reviews. He is a member of the Project Management Institute, INCOSE, and Toastmasters International. Wheatcraft has a bachelor's of science degree in electrical engineering from Oklahoma State University, a master's of arts degree in computer information systems, a master's of science degree in environmental management, and is completing a master's of science degree in studies of the future at the University of Houston, Clear Lake.

Compliance Automation, Inc.

1221 South Main St.

Suite 204

Boerne, TX 78006

Phone: (830) 249-0308

Fax: (830) 249-0309