

## Scope – Magic

By Ivy Hooks and Lou Wheatcraft

*This paper was written as a result of a presentation to the Alamo Chapter of the Project Management Institute (PMI) by Ivy Hooks of Compliance Automation, Inc. Portions of this paper are excerpts from “Guide for Managing and Writing Requirements”, Ivy Hooks, 1994 and from “Customer-Centered Products: Creating Successful Products Through Smart Requirements Management”, Ivy Hooks and Kristin Farry, 2000.*

**Abstract:** A key challenge for project managers is to deliver a quality product to the customer on time, on budget, and meeting or exceeding the customer’s expectations. Only too often, project managers are not able to achieve all of these things for their projects. Why is success so hard to achieve? A major contributor to project failure is the failure to spend the time at the beginning of the project to clearly define the project scope and define the product requirements before beginning product development. Meeting or exceeding stakeholder needs and expectations involves balancing competing demands: scope, time, cost, quality, and stakeholders with differing needs and expectations. This paper covers common scope issues, the benefits of clearly defining scope, and introduces five key tools and techniques you can use to define your project’s scope.

### Project Scope Management

The Project Management Institute (PMI<sup>®</sup>) Project Management Book of Knowledge<sup>®</sup>, defines product scope as “the features and functions that are to be included in a product or service” and defines project scope as “the work that must be done to deliver a product with the specified features and functions.” Project scope management is defined as the “processes required to ensure that the project includes all the work required, and only the work required, to complete the project successfully.” Scope management is primarily “concerned with defining and controlling what is or is not included in your project.” Scope planning is the process of “developing a written scope statement as the basis for future project decisions.” “The scope statement forms the basis for an agreement between the project team and the project customer by identifying both the project objectives and the major project deliverables.”

The scope of a project constitutes the vision: the need to develop or procure a product or service; the goals and objectives of the customer and the company; information about the customers and users of the product or service; and how the product will be developed or purchased, tested, deployed and used. The scope also includes the boundaries and constraints of the product. A project with no boundaries will diverge.

With any project, there may be many versions of the same story. Management or marketing has a view that possibly came from the customer. Architects and designers mull over the vision and modify it for what is feasible -- possibly with no customer contact. Developers take off with their own vision, which at this point may only bear a small resemblance to the original. Meanwhile, no one has been talking to the customer, who may have a whole new expectation. Without an agreed upon and documented vision, there is little hope of achieving success.

It is essential for each project to clearly define and document its scope so that the project can move forward in a coordinated manner and requirements can be written.

## Scope Issues

Some common problems are that a project's scope is often incomplete, ambiguous, unshared, and unstable. Failure to completely define your project scope will result in cost overruns and schedule slips due to rework. Ambiguous scope will result in misunderstandings causing unnecessary work or missed work. A scope that is not shared, i.e., not communicated, will cause misinterpretations in requirements, design, verification, and upgrades. Unstable or changing scope, referred to as "scope creep" is a prime cause of cost overruns and late deliveries and can result in a "never ending" project. The defense against all these problems is to clearly define your project's scope at the beginning and then validate that scope with all the key stakeholders, getting their agreement on the scope before charging ahead.

## Why We Fail

A major contributor to project failure is the failure to spend the time at the beginning of the project to clearly define the project scope and define the product requirements before beginning product development. There are several factors inherent in our culture that are responsible for failing to spend the time required to define project scope: management wants it now and designers want to design – right now.

Management wants to see some action NOW, which means jumping into design or procurement as soon as possible. In this world that demands getting products to market faster, management fears that anything other than building the product is a waste of time. This is simply not true. Those who have thrown out products quickly, only to have them thrown right back, will understand. Fast is not a substitute for meeting customer quality and feature expectations. Management needs to see the scope documented and agreed upon before investing in more expensive ventures such as writing requirements or beginning the development process.

Defining scope will not take long if you know what you are doing. You will have to move more slowly if you uncover many unknowns or cannot get an agreement. Still, charging ahead without required information is doomed to failure. Better to know now than to wait until you release a product.

No one went to school to bother with all this "business and political stuff". Your engineers and programmers went to school to learn how to design and that is what they want to do. Those who have had to throw it all away and start over multiple times are much more willing to listen to the idea of doing scope definition before charging ahead. You must be willing to pay for the resources and must make sure that your project team is doing the right thing at the right time.

## Scope - Its Benefit to Requirement Quality and to Your Project

Project scope definition benefits the quality of your product requirements by preventing incorrect requirements and preventing many omissions. An early scope definition keeps requirement writers from diverging, reduces requirement inconsistencies, and keeps the big picture in view. It also shortens the time required for requirement writing and rewriting and reduces debates.

If you spend time up-front to define a clear scope, you will have more knowledge before you begin to capture requirements. You will document and confirm assumptions and resolve action items before your start writing requirements.

Consider this quote:

***The beginning is the most important part of the work. Plato, 4th Century B.C.***

Obviously we are not dealing with a new problem. This quote indicates that we may be fighting a basic problem of human nature. But there are significant benefits to altering your process if you have not been clearly defining where you want to go before you start your project.

Spending time on defining your scope will directly benefit your project, making your job as a project manager easier. Involving key stakeholders in scope definition will avoid battles that result from differing visions and different interpretations of what your project includes or excludes. Issues can be identified and resolved before investing scarce resources into the requirement writing effort. Spending time to resolve issues, get questions answered, and confirming assumptions will result in less time being needed to write requirements and will speed up the requirement review and baseline process. The impact on development and testing will be even more significant because you will avoid rework and scope creep.

A more recent version of that quote was made by the Vice-President of Xerox:

***In architecting a new program all the serious mistakes are made in the first day. Spinrad, 1988.***

Studies conducted by NASA in 1991 and 1992 revealed average cost and schedule overruns of approximately 65% on 29 programs. This can take a big bite out of your budget. The cost of fixing requirement errors goes up astronomically as you progress through the design toward operations. If you increase your investment in the beginning phase of your project, what sort of return can you expect? Not surprisingly, the greater the investment in scope and requirement capture, the lower the cost overrun. The NASA studies showed that projects that invested 10% or more of their total resources in the requirement capture process before freezing requirements had 0-50% overruns. Cost overruns of 100-200% were common in projects that invested 5% or less.

The earlier you define the project scope, the more efficient the requirement capture process will be. All individuals who write requirements, review requirements, or create designs will create their own scope definition if not given one. Each individual's scope definition may differ significantly from that of project management and will surely differ between the individuals. The result is requirements written for very different goals, objectives, constraints, assumptions, operational concepts, and systems. Battles will be fought, not about requirements, but about these very basic precepts. The requirements will be incomplete or conflicting. This will result in increased costs and schedule overruns, and not deliver what is needed.

## Tools and Techniques for Defining Scope

Here are some tools and techniques that we find useful in capturing the project scope: define the project need, identify key stakeholders, identify project drivers, develop operational concepts, and identify external interfaces.

### 1. Define the Project Need

In product development and procurement, the need must be identified and agreed upon by all involved. You don't want anyone saying: "Why are we doing this?" It is the *need* that typically

initiates a project. The need may come as a result of a market demand, a business need, a customer request, a technological advance, or a legal requirement.

We often start by stating an implementation rather than the underlying need. 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 the question “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 web page, ask “why?” The real reason may be that they need to maintain market share by not losing customers to competitors that have a strong presence on the web.

Once derived, the statement of need should not change over time. If the need is changing, you don’t know what is really needed and you cannot build a product to meet a moving target. Don’t let the real need be forgotten. It is the focus of your investment.

It’s easy to start with an idea for a product. For example, “Hey, our company needs a web page!” It’s even easier to move forward without establishing the actual *need* for a product. The product is not the need and assumptions about need may be conflicting or just plain wrong. Beware of assumptions, digression and gold-plating.

Here is an example of defining a product followed by the real need for the product.

Product:

- Military – new fighter jet
- FAA – new computers and displays
- Commercial – voice-activated thermostat
- Administrative – integrated payroll and accounting system
- Company X – web page

Need:

- Military - to avoid a specific threat
- FAA - to replace out-of-date equipment
- Commercial – increase product line for the sight impaired
- Administrative – to reduce the overhead associated with non-integrated programs
- Company X – to be competitive

## 2. Identify Key Stakeholders

The PMBOK<sup>®</sup> defines stakeholders as: “individuals and organizations who are actively involved in your project, or whose interests may be positively or negatively affected as a result of project execution or successful project completion.” Stakeholders include key representatives from various organizations and groups that have a “stake” in your project. This can include those who buy it, sell it, use it, train others to use it, design it, develop it, test it, market it, maintain it and expect to profit from it. Stakeholders may include designers, system engineers, programmers, verifiers, maintenance personal, installers, trainers, shipping, etc. Each may have a very different point of view and list of priorities. All should be accounted for and considered prior to writing requirements. How do you know the stakeholders’ expectations, assumptions, and rationale? You must ask them directly.

Identifying stakeholders and their expectations is part of understanding the project and realizing how differently it can be viewed. All stakeholders have their own agenda. Their views may conflict or diverge, they need to be addressed and converged. By being aware and prepared early on, conflicts can be handled earlier and possibly more easily. You cannot always meet all stakeholders' goals and objectives simultaneously or even at all. It may be necessary to reset the expectations of one or more groups.

Which stakeholders should be involved in your project? That depends on your organization and project. Here are some example stakeholders:

- Military – Navy, Air Force, supply depots, maintenance, pilot, logistics
- FAA – air traffic controllers, ground controllers, maintenance, airlines, pilots, public, Congress
- Commercial – marketing, engineering, manufacturing, test, maintenance, end user, technical support
- Administrative – finance, payroll, individual employees, training, maintenance
- Company X – marketing, sales, customers, potential customers, competitors

While it is never easy to bring diverse agendas to consensus, you will no doubt benefit from taking into account perspectives other than your own. What's the fun in designing something that is too expensive or impractical to build, test or maintain? Identifying, documenting, and considering stakeholders and their needs will prevent such unpleasant surprises.

Here is an example of different perspectives for a commercial company wanting to build a new "voice-activated" thermostat:

- Marketing - identified a need for a thermostat for the sight impaired. Marketing goals – can replace the competition in 30% of households; possibly all businesses. Available in 6 months. Cost is 1.2 times the regular thermostat.
- Engineering – will need a new resource familiar with voice activation equipment, problems, and resources to make schedule.
- Manufacturing – no redo of current process, except for addition on another chip at position 12 on the line.
- Quality – will get early test equipment and prototypes. Concerned about how to test for different voices. How do we keep some other noise from changing the thermostat?
- Technical support – worried about how to support customer who cannot see the box, how will sight impaired customers tell technical support what is wrong? Can the box talk back and provide data?
- End user - Can find out the temperature, can set the temperature, and have all the bells and whistles of the existing thermostat just by talking to it.

Customers and users are some of the most important of the product's stakeholders. Knowing the needs of your customers and users is critical to the success of the project. While most people would agree, they still often fail to ask the customers and users what they really need. They make assumptions. Stakeholders can make your project a success or failure depending on how they view it. Key stakeholders are often ignored until far too late in the process. It is vitally

important to your project's success that key stakeholders are identified during the development of the project scope, they are involved in the project scope definition, and they buy into the project scope. This will ensure everyone associated with your project shares a common vision.

### **3. Identify Project Drivers**

We are rarely able to just build what we want to. We are driven by many outside influences, e.g., regulations, standards, laws, and other considerations. A major driver for many organizations is the set of existing equipment, software, or processes. Other drivers include security and safety concerns. Depending on your type of business you may be affected by an overabundance of regulations from organizations such as the FDA, FCC, or FAA. Early attention to drivers is important to your project. Each driver needs to be identified, assigned for tracking, and included in the analysis of what the project is and is not. The constraints imposed by external drivers can be project killers if unknown or not addressed early in the project.

### **4. From Needs to Requirements – Develop Operational Concepts**

How do you take

*“I want it to be fool-proof”*

*“I want it to be user-friendly”*

*“I want it to operate like the last one did but to be cheaper”*

and turn them into verifiable requirements? --- We do operational concepts.

Operational concepts bridge the gap between product scope and formal requirements. The operational concepts are plain-language descriptions of user-product/system interactions in the life of your product for both nominal and off-nominal conditions. How will it be used, manufactured, tested, installed, maintained, stored, and decommissioned? Operational concepts may be “use cases,” “operation plans,” “scenarios,” or other methods of uncovering gaps in knowledge and scope. Many project teams will focus on just the operations and ignore the other phases of the product life-cycle. This is a mistake. By covering all of the life-cycle of a product you will help prevent missing requirements.

Many great products start with an intuitive operational concept. High-level operational concepts will help in the creation of the project scope. In the beginning, there may be several alternatives the project could select to meet the project need. In fact, the operational concept may exist before the need statement. Scenarios allow different operational ideas to be explored. Then the feasibility of each scenario is examined and more ideas are explored. A needs statement can be written after the scenarios capture the true need. The operational concept, like all other parts described above, is iterative. A single alternative must be selected and reflected in the project scope to ensure that key stakeholders share the vision before requirement capture begins.

The practice of defining and documenting operational concepts is a simple, cost-effective way to build consensus among all stakeholders and discover which questions still need to be asked and answered prior to writing product requirements. Operational concepts are intuitive and easy to generate, are in a language that everyone understands, facilitate complete and consistent requirements, reduce the debate, identify user interface issues, offer inexpensive opportunities for early validation, form a foundation for testing scenarios in product verification, and are an essential part of preparing for requirements.

Correcting what you have is easy compared with determining what is missing. This is the time, prior to writing requirements, to consider issues in manufacturing, testing, user training, installation, maintenance, and disposal. Missed requirements in these areas are a major source of rework and schedule slips. Use operational concepts to go beyond “normal operation think” and run through more interesting scenarios. What is the effect on performance of, say, an ice storm or migratory bird encounter? Will it be damaged during shipping? Can you get it in a standard elevator? Operational concepts are an ideal medium for finding problems outside the basic usage area.

By using operational concepts as a communication vehicle, you can avoid many of the fiery debates that often ensue during the average requirement definition process. Why? Most people have a non-documented “operational concept” in mind as they formulate, write or review product requirements that are usually the source of the disagreement. Unless these operational concepts are spelled out and shared, people involved can’t recognize these differences. It’s easier to hammer out differences and reach consensus while developing operational concepts than while you are defining project scope or writing product requirements.

You can take operational concepts to a wide variety of stakeholders to get feedback on what is feasible. This feedback will help fill in holes, find inconsistencies, and correct your course. Operational concepts can also be used as quick validation simulations.

Developing operational concepts is like storytelling. You simply imagine the operation of the product-to-be and write the steps or scenario. It’s easy to involve customers and users in operational concepts creation and critique.

### **Operational Concept: Thermostat from User Viewpoint**

- Walk over to within a couple of feet of the thermostat.
- Say something that gets the thermostat’s attention.
- It gives me verbal reply that it is listening.
- I ask for “setting”.
- It tells me: set for cool, fan is on automatic, and the setting is 78 degrees F. It then tells me that it is currently 79 degrees F. It then asks me if I want to change the setting.
- If I say no, then I’m done.
- If I say yes, it asks me if I want to change the temperature setting.
- If I say no, it asks if I want to change the fan to manual.

Operational concepts result in the development of a common project vocabulary. While different stakeholders go through their operational concepts, they will use certain terms. It is important to agree on the definition of these terms and document them in a project glossary. Another result is the identification of assumptions. Ask the stakeholder what must be true in order for their operational concept to be valid. This will bring out key assumptions the stakeholder is making. As different stakeholders “tell their story” in the form of an operational concept, there may be disagreements, incompatibilities, and other issues that need to be resolved before requirement capture can start. The resolution of these issues may identify the need for additional research, prototyping, or modeling to validate a concept or examine alternatives. Areas of uncertainty will be identified, as will areas of possible risk. These need to be documented and tracked as part of the project’s risk management process. Finally, a major result of developing operational

concepts is that functions the product needs to include and key areas that must be addressed in the requirement document will be identified. Taking the time to do operational concepts has large paybacks. It reduces the project risk early in the system life-cycle.

## 5. Identify External Interfaces

Serious problems arise at interfaces. Your project is particularly vulnerable to interfaces with other products over which you have no control. Consider IBM at the Atlanta summer Olympics – they bragged about their great system for capturing the data from all the contests, only to have a major failure at an interface. IBM could not send data in the correct format to the newspapers. Thus the newspaper could not automatically insert daily game results into their issues. Most of the failures NASA has sustained in the past few years in their Mars program are a result of interface problems.

Defining interfaces was much easier when all was hardware, now that we have software, with data and commands going everywhere, it is a much harder task and there are often a very large number of interface requirements. With hardware there were interface drawings, not so with software. With software, the interface definition can be extensive and clear documentation is crucial.

***Be sure to ask the question, “What is the worst thing that other elements could do to you across the interface?” Kuchta, 1989.***

Many projects neglect to identify and control the interfaces until testing. The first encounter with the results of this oversight often occurs when people find out that they cannot plug test equipment into the product to perform the tests. Worse yet, some projects neglect interfaces until operations. By identifying your external interfaces early, you are identifying key drivers for your product that must be addressed in your requirement document.

External interfaces fall into two broad categories: user (usually a human being) and everything else. User external interfaces include buttons, levers, handles, straps, warning bells, safety labels, and displayed information. Non-user external interfaces include avionics buses, structural support, power supply, command, data, operating system, computer system, storage container, and existing equipment used with your product.

Identifying external interfaces early will clarify your product scope, aid risk assessment, reduce product development costs and improve customer satisfaction. External interfaces can be dealt with prior to design start. An early study of the interfaces will eliminate development rework due to bad interface assumptions.

There are several external interface considerations that should be made prior to writing requirements, including the need to interface with a non-standard product, lack of interface documentation, and the risk of changes to products out of your control with which yours must interface. All of this information is vital to your project’s risk assessment and change management processes.

There are different classes of interfaces you must consider: First, there are fixed interfaces that are well known and unchangeable. You will design your product to interface with this type of interface per a documented set of interface requirements. Second, there may be existing interfaces that can be changed, either to save money, or because that interface is being updated.



Negotiation is needed with the interface owner. The third type of external interface is one that does not currently exist. In both the second and third cases, negotiation must be completed and agreed to as part of the process. However, the key point is to identify and classify this interface early, before you begin requirement capture.

To identify external interface issues, make a brief list of anticipated external interfaces, considering each item throughout the product's life-cycle. Then, ask these questions at each boundary: What does my product do to the world? What does the world do to my product? What is the worst thing that can happen across this interface? Is this interface likely to change during the development of my product? What units are this interface in? And finally, is this interface likely to change after my product is in use?

Benefits of addressing external interfaces during scope definition:

- Reveal the bounds of your system
  - Define what is in
  - Define what is out
- Provide a list of areas that will need to be explored fully
- Expose potential problem areas

## Define, Document, & Disseminate Scope

As time goes by and your team conducts design studies, the initial scope definition may need revising. Change can come from suppliers, customers or outside factors, such as new technology. The revised scope document must be available to everyone on the team to help them stay on the same track.

A formal documentation of scope is essential to keeping a project on track. The purpose of documenting the scope is to ensure that everyone that signs up to do the job has signed up to do the same job. Not only must a documented scope exist, it must be made available. This is often overlooked and many people working on a project have never seen and do not even know that a unified scope exists. Scope is of little use if it is not read by everyone. Even if you are managing a very small project, you need to document its scope to ensure success.

The order and amount in which scope items are covered in this document will depend on the size of your project. In general, the scope document should be short, succinct, and available to everyone. Defining scope is also iterative for each level of a project. As you break the project into manageable pieces, such as "hardware" and "software," you must also scope these pieces. Later on, use the scope document to ensure written requirements fall within your scope definition. This will help ensure a complete requirement set as well as prevent gold-plating. (Gold-plating is when features or functions are added to a product without an approved system requirement for that feature or function. Gold-plating can result in increased development costs as well as increased verification costs.)

## What Is the Vision?

How would you answer the following? Have projects you've been on taken longer than they should? Have you or others involved in these projects had difficulty interpreting the requirements? One reason could be the lack of a shared vision while defining your project scope and product requirements. If each person writing requirements or reviewing requirements has a

different perception, it can lead to trouble. To avoid this, the project vision must be agreed to, documented, and shared with everyone involved in the project's life-cycle.

Change of direction is probably the single largest factor in cost and schedule overruns. We need to stay on course and keep from changing scope and requirements. We need a better way that will help us "do it right the first time."

## Parting Thoughts

You must insure that resources are allocated at the beginning of your project to define a complete, non-ambiguous, shared, and stable project scope. Doing this will allow you to deliver a quality product on budget and on schedule that meets or exceeds the customer's expectations. Anything less will compromise schedule, budget, and quality. To be a successful project manager you need the "magic" that will result in defining project scope.

## About the authors:



**Ivy F. Hooks** is President and CEO of Compliance Automation, Inc. She has nearly 40 years experience: 20 years with NASA, where she was the separation integration manager for the Space Shuttle and also manager of Shuttle flight software verification, and 17 years in private industry. For the past 12 years, she has specialized in requirement definition and management, conducting training seminars, and performing consulting work for government and industry. Ivy holds a BS and MS in Mathematics from the University of Houston. Ivy is co-author of the book, *Customer-Centered Products:*

*Creating Successful Products Through Smart Requirements Management* (AMACOM) ISBN 0-814-40568-1.



**Louis S. Wheatcraft** joined Compliance Automation, Inc. in the summer of 2000. Lou has over 34 years in the aerospace industry, including 22 years in the United States Air Force. Prior to joining Compliance Automation, Lou worked for Barrios Technology, providing support at Johnson Space Center to the Astronaut Office developing operational concepts, requirements, ground rules, and constraints for how the astronauts will live and work on the International Space Station. Lou has a BS degree in Electrical Engineering from Oklahoma State University, an MA degree in Computer Information Systems, an MS degree in Environmental Management, and is completing an MS degree in Studies of the Future from the University of Houston – Clear Lake.

If you want additional information about CAI products or services, please contact us via e-mail: [sales@complianceautomation.com](mailto:sales@complianceautomation.com) or call (830) 249-0308.

"PMI" and the PMI logo are service and trademarks registered in the United States and other nations; "PMP" and the PMP logo are certification marks registered in the United States and other nations; "PMBOK", "PM Network", and "PMI Today" are trademarks registered in the United States and other nations; and "Project Management Journal" and "Building professionalism in project management." are trademarks of the Project Management Institute, Inc.