

Kick-Starting Your Projects – What Should You Know About Defining Scope

By Jamal Moustafaev, MBA, MBA, PMP
President & CEO
Thinktank Consulting

M247 “Sergeant York” Story

In late seventies the US Department of Defense (DOD) outsourced the development of the self-propelled anti-aircraft (AA) weapon which featured twin radar-directed 40 mm rapid-fire guns to Ford Aerospace. The project was assigned the name of “Sergeant York”, after the World War I US army hero, who undoubtedly would not have appreciated this dubious honor had he been alive in 1979. The weapon was intended to replace the M163 Vulcan Air Defense System and fight alongside the M1 Abrams and M2 Bradley fighting vehicles in the U.S. Army, and was similar in concept to successful Soviet and European systems such as the *ZSU-23-4* and *Gepard*

In essence it was an air defense weapon mounted on the surplus M48 Patton tank chassis, provided by the Army, which were held in large quantities in their depots. The main job of the weapon was to sit on the front lines and automatically target and shoot down enemy aircraft, especially helicopters. As a result it was designed to hone in on metal parts rotating in the air (i.e. propeller blades).



The final test of the AA gun involved a demonstration involving a prototype weapon shooting down a hovering helicopter on one of the US DOD proving grounds somewhere in the desert in the southern part of the United States. The cost of the project at that time was approximately US \$1 billion.

According to the legend cultivated in the aerospace and defense industries there was a portable toilet installed not far away from the testing grounds. Because of the hot climate, the toilet cabin had a small electric fan in it ...

You probably managed to guess the rest – the billion dollar piece of equipment “decided” to ignore the much larger target – the helicopter – and targeted the unique signature of the “port-a-potty’s” electric fan!

Further tests revealed that the AA gun had the following deficiencies:

- The radar could not track low flying targets due to excessive ground clutter
- The radar could not distinguish a hovering helicopter from a clump of trees
- Turret traverse was too slow to track a fast crossing target
- The radar can be jammed very easily

As a result, the project that began in 1975 was finally cancelled by then Secretary of Defense Caspar Weinberger in 1985.

This story is, in my opinion, a perfect example of a botched scope definition exercise as most of the Department of Defense requirements, both obvious (i.e. ability to track low flying targets, fast moving targets and resistance to radar jamming) and implied (not confusing the helicopter with a clump of trees or a portable toilet) have been left unfulfilled.

What Is The State Of The Industry?

Standish Group and Project Management Institute (PMI) paint a fairly bleak picture: we still only managed to “achieve” a 35% success rate on our projects. Apply this number to a total price tag of all projects in US (\$2.3 trillion) and you will arrive at a mind-boggling \$1 trillion dollars in underperforming ventures annually! Furthermore, according to the Standish Group five out of the top eight reasons why projects fail are related to scope definition. These include:

- Incomplete requirements
- Lack of user involvement
- Unrealistic customer expectations
- Changing requirements and specifications
- Customers no longer need the features provided

Further analysis of causes of “bad” requirements yielded the following results (see Exhibit 1):

Exhibit 1

Incorrect Fact	49%
Omission	29%
Inconsistency	13%
Misplaced Requirements	5%
Ambiguity	2%

In addition, the study conducted by Barry Boehm (one of the leading thinkers in the field of software development) involved an analysis of 63 software development projects in companies like IBM, GTE and TRW and determined the relative costs of fixing an error at various stages of the project. Exhibit 2 demonstrates a phenomenal increase in the cost of mistake from one dollar at the Initiation stage of the project to \$40-1,000 dollar range at the Closeout (see Exhibit 2).

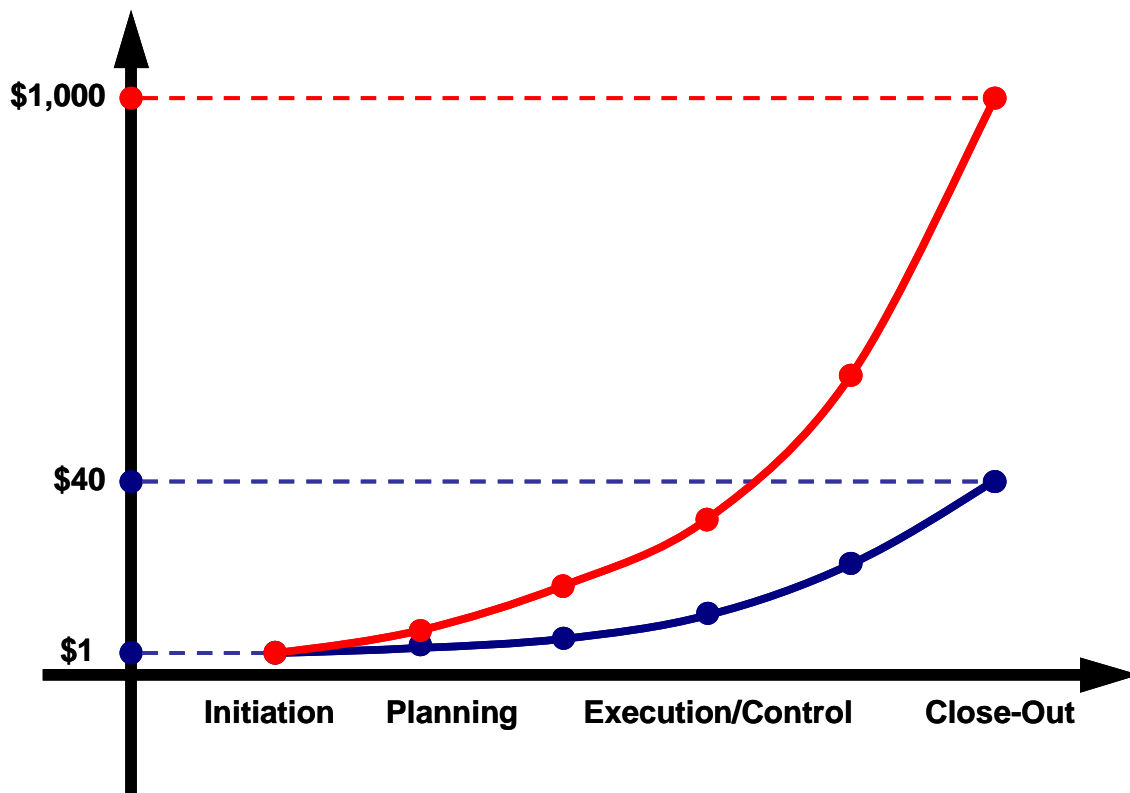
Now, I do realize that this information is based on the software development industry; unfortunately I wasn't able to find comparable data from other fields. However, I think it would be reasonable to assume that this trend will, to a certain extent, hold true for most of the other industries.

To prove my point I like to provide my students with the following example. Imagine that you have made a “one-dollar” mistake at the very beginning of the project ... and forgot to include a requirement for an underground parking garage for a condo building your company is about to erect. How much, do you think this mistake will end up costing at the end of the project when you realize that the only way to “insert” the garage is to completely destroy the tower and start the

entire project from scratch? The mistake ratio will no longer be 1,000:1 it would probably balloon to 10,000,000:1, wouldn't it?

The amusing aspect of this story is that until recently I used to apologize to my students for supplying them with this "silly" example because this "would never happen in real life", until one of course attendees informed me that a similar story indeed took place in one of the major North American cities. Apparently a real estate development company acquired a hundred-year old heritage building with the intent of converting it into a luxury condominium complex. One "little" thing that they somehow overlooked – heritage buildings do not have underground parkings! The problem was finally solved by acquiring (at a great cost) the rights to a portion of a several dozen parking spots in a new high-rise located nearby.

Exhibit 2



I think based on the information provided so far in this section it is safe to assume that a considerable share of our project failures originate in our inability to capture the scope properly. However, based on the feedback I was getting from the technical people in various industries (engineers, architects, construction experts, etc.) and my students, the scope problem is not rooted in our inability to come up with detailed blueprints, bills of materials or design and architecture documents. The predicament lies in the initial stages of the projects, when we need to elicit high-level initial set of customer problems, issues, needs and propose potential solutions.

Defining The Initial Scope

What Is A Business Requirement?

In order to answer what is a business requirement, let us first determine what requirements are NOT. The business requirement is not a combination of e-mails, voicemails, sticky notes,

annotations in your notebook, verbal instructions from the customers and your superiors and/or “drawings on a napkin”. Therefore a business requirement (aka high-level project scope item) is:

“A requirement is something the product or service must do or a quality it must have. A requirement exists either because the type of product or service demands certain qualities or functions or because the client wants that requirement to be a part of the delivered product or service”

Business requirements should be documented either in a standalone “Business Requirements” document or, at least, captured in the Project Charter.

Asking The Right Questions

The first step in every project should be trying to determine what is it that you are going to build for you customers, either internal or external. Setting off in a right direction depends to a large degree on the type of questions you or your team members ask. For example, let’s assume that the first question posed by your management was:

“Can you create a means for protecting a small group of human beings from the hostile elements of their environment?”

As a result of the discussion that may follow, some of the possible solutions could be:

- Shack
- Typical family home
- Buckingham Palace

But “what are we building?” is by far not the only question you should be asking at the beginning of the project. Exhibit 3 lists the mandatory high-level questions project manager should be prepared to ask once the project is handed down to him.

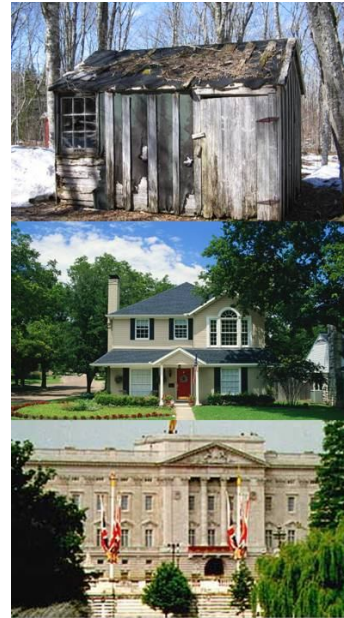


Exhibit 3

Question You Should Ask	Why Should You Ask This Question?	What Happens If You Don't Ask It?
Why are we doing this?	To make sure there is a clear understanding of the needs and benefits for the project	You may end up working on a project that will deliver little or no value to your company or the customer
What happens if we don't do this project?	Just another way of ensuring that the management thought through the potential “penalties” of not doing the project	Your company may end with low-priority, “soon-to-be-cancelled” projects while the high-value projects remain on the backburner
Who is the client?	You have to know who the original source of the	You may end up receiving requirements that have been

	requirement is	“distorted” by going through the “chain of command”
What problem are we trying to solve?	As a representative of the “technical” team you may have a much better (cheaper, faster) solution the customer is not aware of	You may end up delivering a “Ferrari” when a “bicycle” solution will do (or vice versa). As a result you may: a) waste company resources b) disappoint your customer
If you don’t know this, then who does?	You need to make sure you are getting the correct information	You may end up talking to the “messengers” instead of real project champions. As a result you may end up with bad-quality information
Who else could be impacted by this feature?	There could be hidden interdependencies between this feature and other ones you are not aware of	You can miss important interdependencies or even contradictions between various scope items.
How much money are you ready to spend?	You need to gain a high-level understanding project budget	There could be a significant discrepancy between what the customer wants and what you can deliver at the budget he has in mind
How much time do we have?	You need to gain a high-level understanding project duration	There could be a significant discrepancy between what the customer wants and what you can deliver with the time she is willing to provide you with

Once the ballpark scope of the project has been determined we can proceed to more detailed questions (these could also fall into the category of Project Charter-level questions):

- How big do you want the house to be?
- Is it going to be a one, two or three-level home?
- What square footage would you prefer?
- What style would you prefer (e.g. Victorian, Tuscan, Contemporary, etc.)?
- How many bedrooms do you need?
- How many bathrooms would you prefer?

Also, keep in mind that all of the questions directed at the entire project in Exhibit 3 should now apply to every scope item under discussion. For example, let’s review some of the questions a project manager can ask with respect to the number of bedrooms:

- Why do you need five bedrooms?
- Would you be OK with just four?
- The budget you mentioned allows only for a four bedroom home; would you like to downgrade the scope or allocate more money to the budget?

Assigning Priorities

Since typically customers want way more features than the project budget and timeline will allow, it is also very important to assign priorities to the scope items or “features” discovered at this stage. The suggested categories for the priorities are:

- Priority 1 (“Must Have”)
- Priority 2 (“Should Have”)
- Priority 3 (“Nice To have”)

Unfortunately the concept of assigning priorities is very frequently misunderstood by the project stakeholders. Typically a lion’s share of the feature gets stamped with “Must Have” priority label right at the beginning of the process, thus eliminating any potential flexibility in future scope-related decisions.

Therefore I like to provide my students with the following example of priority assignment. Imagine that we are designing the first-ever car. What scope items would fall into the “Must Have” category? In my opinion, these should include:

- Frame
- Four tires (or maybe even three, depending on design)
- Engine
- Driver-side seat
- Steering wheel
- Brakes



Things like passenger-side seat, seat belts, windows and roof will fall into the “Should Have” bucket. Car stereo with speakers and navigational system should fall into the “Nice To Have” category

Getting Rid Of The Ambiguity

Another important and very frequently overlooked aspect of the scope definition is the ambiguous language one can encounter in the project and portfolio documentation. These include words like: “fast”, “pretty”, “big”, “small”, “cutting-edge”, “usable”, etc. The danger of these words is that psychologically we are “trained” to accept them as normal, everyday concepts. However, when it comes to project management, words like these can act as deadly time-bombs. For example, the word “cutting edge” seems like a normal, even cool word to describe a product. And while it looks fine in marketing materials or TV ads, introducing this word to scope documentation can cause a lot of issues down the road.

The reason for that is, while everyone understands, more or less, what “cutting-edge” means, no two people have an identical understanding of the concept. These differences in understanding can lead to very expensive and time-consuming scope adjustments once the project moves into planning and execution stages.

One of the executives I was having a discussion with told me, “We had a construction project budgeted at \$500 million dollars once. Because we allowed the words like “cutting-edge” to sneak

into the Business Case our final bill for the project ballooned to \$700 million. There always was at least one key stakeholder in the room for whom a given feature was not revolutionary enough! And because we initially made a commitment to being “cutting-edge” there was no way for us to back away from that”.

The Problem vs. Solution Discussion

When gathering the requirements project managers very frequently come across stakeholders’ ideas for a solution rather than the description of the underlying problem. You should always strive is to interpret what is said thereby uncovering the **essence**. Let me provide you with a couple of examples.

Example 1

Let’s pretend that you are security company expert and you have been contacted by a manager of a very large downtown office building that had experience a string of thefts and break-ins from the offices of the companies located there. There are two possible scenarios of how this conversation may go:

Scenario A:

Building Manager: *“Your company has to install card readers at all the entrances to the building and we need all the employees in this building to have an access card”*

You: *“OK, no problem. Let’s determine the number of entrance points into the building and the number of people requiring the card and I can probably provide you with an estimate for the project.”*

Or we can go with scenario B:

Building Manager: *“Your company has to install card readers at all the entrances to the building and we need all the employees in this building to have an access card”*

You: *“So what you are really saying is that we need to make sure that only authorized people have access to our building, right?”*

Building Manager: *“Yes, that is exactly the problem we have”*

You: *“Well our company has a variety of alternatives for you depending on your budget, time and how ‘state-of-the-art’ you want the solution to be. Here is a list of options we can offer:*

- *Retina scan*
- *Fingerprint scan*
- *Palm scan*
- *Voice recognition*
- *Card and card readers*
- *Remote controls*
- *Chips implanted under the skin*
- *Hiring a security guard*
- *‘Neighborhood Watch’ program”*

Building Manager: *“Oh, I didn’t realize we had so many options. Let me talk to my colleagues and get back to you”*

In which case do you think you have made a better impression on the customer and, more importantly, in which case would the solution chosen address customer’s needs better?

If this sounds like a bit of a cheesy scenario, especially considering the “chip implanted under the skin” part, let me give you an actual real-life and extremely unglamorous example of how this essence principle works.

Example 2

These events took place while I was working at an IT department of a very large international financial institution.

My boss: *“Risk Management is having problems with their desktop statistical analysis software. They are asking for an Enterprise Edition of the software and a dedicated server. Our initial ballpark estimates for this project are at \$500,000 and we have neither money in our budget, nor the resources to accomplish that. Your mission is to explain to them that they are not getting this stuff in the next couple of years!”*

Me (going over to the Risk Management Department): *“What is the problem?”*

Risk Management people: *“We have to store files on the shared server because of the privacy laws and access them through our desktops. Processing times are very slow. We have to upgrade to Enterprise Server Edition and get a new server”*

Me (calling Network and Infrastructure people): *“But why is the processing slow? Is it the network or the overloaded server?”*

Network people: *“We checked the network and it does not appear to be overloaded”*

Infrastructure people: *“Are you kidding me?! The entire building is using this server. Of course it is overloaded and slow”*

Me: *“So, what can we do?”*

Infrastructure people: *“We will give them a dedicated NT server; we think we had one lying around here somewhere ...”*

The end-result of this story? The entire \$500,000 project was diminished to a \$2,000 server and 3 man-days of work.

Constraints, Preferences and The Tilt Concept

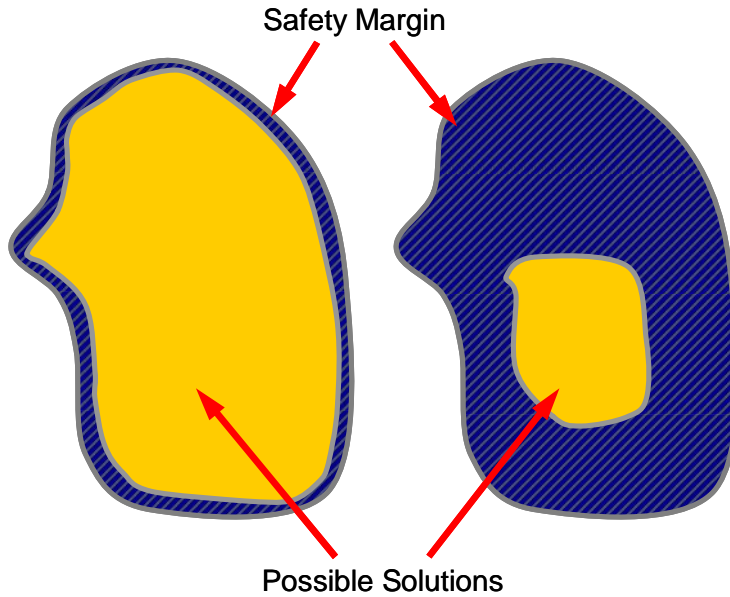
In one of the books I was reading I came across an interesting concept presented by the author - “The Tilt Concept”. The author offered his observations of various pinball players. According to him, some pinball players tilt their machines all the time. These are not very good players because they can’t restrain themselves. Some of the players never tilt the pinball machines; they are the worst players because they always follow the rules. Author contends that the best players do tilt the machines but they never overdo it. Hence, “The Tilt Concept” - If you never tilt, you are not using your resources!

Exhibit 4

WHAT SOMEONE SAYS	WHAT WE DO
<i>"That information is confidential"</i>	We stop pursuing something that may be essential to good design
<i>"That violates standards"</i>	We drop a promising idea
<i>"The boss would never approve of that"</i>	We shrivel into a ball and change the subject

Some of the examples of us being afraid to "tilt" are shown in the Exhibit 4. Tilting also means unnecessarily constraining your design options. Here is an example of what "bold" vs. "timid" designers would do in similar situations (see Exhibit 5)

Exhibit 5



Related to the "Tilt Concept" is the ability to distinguish between mandatory constraints and desired preferences.

For example, an information system designed to tabulate and report US presidential elections must be ready to operate by the first Tuesday after the first Monday in November of the next leap year. If the product is not ready by that day – it has no value for the next four years. This is definitely a constraint! A requirement to upload a photo of the new executive team member on to the company website by end-of-day Friday is typically a preference.

Conclusion

A large share of our project troubles is rooted in our inability to capture the initial high-level scope of our ventures. There are several techniques that a project manager should utilize in order to elicit the right business requirements early in the project.

First, in order to overcome this issue project managers have to understand the definition of business requirements and record them properly either in a separate standalone document or include them in the project charter.

Second, project managers have to ask the right questions including such “uncomfortable” ones as “Why are we doing this?”, “What problem are you trying to solve?”, “What happens if we don’t do this project?”, etc.

Thirdly, a disciplined prioritization approach should be utilized on every scope definition exercise as well as “ambiguity time bombs” should be eliminated from the project management documentation.

Concentration on the essence (i.e. the original problem) rather than on the implied or suggested solutions should always reign supreme in project manager’s mind.

And finally, understanding the, oh so subtle, differences between the real, hard constraints and desired preferences is another important ingredient in defining project scope.

Bibliography

1. “Project Management” by Robert Santarossa
2. http://en.wikipedia.org/wiki/M247_Sergeant_York
3. Major Micheal Ditton, "The DIVAD Procurement: A Weapon System Case Study", The Army Lawyer, August 1988, pp. 3–9
4. "Gunning for Sergeant York", Time, August 1985
5. "No time for Sergeant", The Nation, September 1985
6. “Software Requirements, Second Edition (Pro-Best Practices)” by Karl E. Wiegers
7. “Effective Requirements Practices” (Addison-Wesley Information Technology Series) by Ralph R. Young
8. “Exploring Requirements: Quality Before Design” by Donald C. Gause and Gerald M. Weinberg

About Thinktank Consulting

Founded in 2003 by Jamal Moustafaev Thinktank Consulting is an industry leader in providing management consulting services to North American software development companies and IT shops. Our portfolio of services includes:

- Project Management and Requirements Engineering (Business Analysis) consulting and outsourcing
- Process improvement programmes & audits including:
 - Business Process Re-engineering
 - Requirements Engineering
 - Estimation
 - Project Management
 - Portfolio Management
 - IT/Business Alignment
- Training
 - *“Practical Portfolio Management - Selecting & Managing The Right Projects”*
 - *“Successful Hands-On Management of IT and Software Projects”*
 - *“Successful Hands-On Management of Modern-Day Projects”*
 - *“From Waterfall to Agile - Practical Requirements Engineering”*

For further information, please contact:

Jamal Moustafaev BBA, MBA, PMP
President and CEO
Thinktank Consulting Inc.
Phone – 778-995-4396
E-mail – info@thinktankconsulting.ca
Website – www.thinktankconsulting.ca