# Scope Creep Horror

## It's Scarier Than Movie Monsters

*Wayne Turk*

For a program manager, there is something scarier than Halloween, the Blair Witch Project, Friday the 13th, or any other horror movie that you can think of. It's the monster on the other side of the wall waiting to devour resources and destroy the project schedule. It's ...

***Scope creep!***

**Turk** *is an independent management consultant. A retired Air Force lieutenant colonel and defense contractor, and the author of* Common Sense Project Management *(ASQ Press, 2008), he is a frequent contributor to* Defense AT&L.

One definition of scope creep is "the gradual expansion of project work without formal acceptance or acknowledgement of their associated costs, schedule impacts or other effects." Another is "the process of adding work and requirements, little by little, until the final project no longer resembles the original one and the original cost estimates and schedule have become meaningless and unworkable." It's very scary, and it happens with projects every day.

## Why Does Scope Creep Happen?

There are a number of reasons for scope creep, and the following are a few of the most common:

Poor initial requirements. Someone didn't do a good job on writing the original requirements or objectives. Too often, requirements are poorly written. They may lack clarity or detail. They may be ambiguous, vague, or not understandable. They may be contradictory. The end users or potential customers may not have been involved. The requirements may not be organized and prioritized. Whatever the reason, a poor set of requirements or objectives can lead to disaster when changes or additions come along.

Unwillingness to say no to a client. The client is ultimately in charge in that he or she is footing the bill and is the person to whom the project is delivered. It may be your boss, it could be someone else in the company/organization, or it might be an outside customer. Too often, PMs are intimidated by the client and afraid to say "no," or else they want to be seen as the good, can-do guy. Understandably, the program manager doesn't want to antagonize the client, but that reticence can be a steppingstone to failure.

No formal review and approval process for changes. Changes are accepted willy-nilly because no board, panel, committee, or person has the responsibility of looking at the changes and measuring them against some kind of acceptance criteria. There must be process and acceptance criteria, and funding for the changes must be included in those criteria.

Allowing people who don't do the work to accept the changes. Too often it is someone other than the PM or project team who accepts the change and then passes it to the team. That is not the same as having a person or group to review and approve changes within a formal process, and it is very dangerous.

Ego. The project manager has inflated pride, ego, or confidence in himself and/or his team. He thinks that they can accomplish anything. The team might be able to make the change, but at what cost (financially or otherwise)?

Thinking that one little change won't matter. That one change can lead to or force another and another until the one little change has become a large change or even a series of large changes. Once scope creep has its foot in the door, it is difficult to halt.

## Controlling the Scope Creep Monster

Scope creep can be the bane of a project's success, if not its very existence—and unlike a movie vampire, you can't keep it away with garlic or a wooden cross. It takes planning, determination, and good processes to defeat it.

### *Requirements*

Let's start with the project's requirements or objectives (the term requirements will be used from this point to describe both). The first characteristic of a good requirement is that it is necessary. With today's fiscal constraints, there is rarely any room for nice-to-have or frivolous requirements. The requirements must be accurate as to what the product needs to deliver. Requirements must be unambiguous. Multiple readers should come to the same understanding of what each means. If a requirement can be interpreted more than one way, you are in trouble because chances are that the developer or builder will interpret it the wrong way. Terms like "user-friendly," "fast," "easy," "flexible," "state-of-the-art," "maximize," "minimize," or "efficient" all mean different things to different people, so avoid them like the plague. All requirements must be feasible, attainable, achievable, and expressed in quantified terms that mean the same thing to everyone.

Requirements must be prioritized. The priority is normally set by the end user or customer, but the PM may have a say—especially when the user sets the same priority on a number of requirements. Along with operational needs, other factors can influence priority. For example, cost can play a huge role. If meeting one requirement will cause the expenditure of 75 percent of the budget, it probably shouldn't be the highest priority unless, of course, it is the primary requirement of the project. Technical risk and schedule impact are other influencing factors. They must be weighed and the users have to understand their effect on priorities.

All requirements must be quantifiable, measurable, and verifiable in some way. There are a number of ways to verify that a requirement has been met, among them inspection, analysis, demonstration, simulation, and testing. Just remember that every requirement must be verifiable in some way. It should be verified it in the most expeditious and least expensive manner possible.

Verifiability is related to traceability. While especially critical in software development, in any project someone should be able to trace a requirement from identification through development to final verification. Requirements need to be written with the same terminology and the same standards throughout. It also helps for them to be organized and grouped into defined categories. That allows the team to find duplications, inconsistencies, and contradictions.

Finally, requirements must be results-oriented. The objective of the complete requirements package is to provide a product that meets the users' needs and/or solves a problem. It doesn't necessarily have to look good, involve the latest technology, or do all kinds of extra things. It must provide the results and the product that is wanted.

**Arm yourself with good requirements, strength, determination, good processes, and planning, and you can defend against and even slay the scope creep monster that's threatening your project.**

### *Accepting or Declining*

Project managers have to learn when to say no and when to say yes. When the client wants to change or add a requirement, the change or addition should be analyzed for resource, cost, and schedule impacts. There should be a standardized review and approval process. If there is an impact to the cost or schedule, the client must have the facts presented and then must formally (and preferably in writing) accept any change to the cost and schedule. That usually means adding more funding to the project, extending the schedule, and/or dropping other requirements to compensate for the change.

At times, a change or addition will need to be declined. It isn't always easy to say no, especially if the change is coming from a boss or a good customer. It requires strength and determination. If the answer needs to be no, it will also require an explanation. The project manager needs to get the facts together as to why the change can't (or shouldn't) be accepted and present them logically and unemotionally. That is where the review process comes in. The analysis can determine what the negative impacts are and provide details and numbers as the basis for denial.

A project manager cannot let ego or fear get in the way of saying no. Even if the PM has a great team he thinks can do anything, they need the time, tools, and money to succeed. If a PM doesn't have the strength, willingnes, and communication skills to stand up and say no and explain her decision, she should not be in the management position. That is a cruel thing to have to say, but it is the truth.

### *When There is No Choice*

Yes, there will be times when the PM will be overuled by someone higher up the chain of command, logical arguments and facts notwithstanding. And someone else's decision to accept a change may not come with additional funding or schedule adjustment either. If that happens, there are a few things that can be done to minimize the schedule or cost impacts. (They are actually good guidelines for a project at any time.) This is certainly not an all-inclusive list, and the items are not in any order of priority, but it's a start:

- Leverage on previously developed work. If you can use something that someone else has already done or paid for, do it.
- Set a timeline or due date for all tasks. Have a tracking system for tasks, due dates, and action items. Review the tracking system frequently.
- Assign responsibility for each task to someone.
- Consolidate tasks for cost- and timesavings.
- Make tasks sequential only if they have to be.
- Use some form of earned value management.
- Track costs closely and compare them to planned costs.
- Project upcoming costs and revise them as changes occur.
- Don't use gold-plated requirements (those that are higher or more complex than actually needed).
- Use cost-benefit analyses to help make decisions.
- Don't waste resources on unnecessary work.
- Do things right the first time; rework is expensive.
- Prioritize requirements and tasks to identify what can be cut if something has to go.

## You Can Slay the Dragon

Scope creep is that monster hiding under the bed, ready to sneak out and kill your project. Yes, it's scary, but arm yourself with good requirements, strength, determination, good processes, and planning, and you can defend against and even slay the scope creep monster that's threatening your project.

The author welcomes comments and questions and can be contacted at rwturk@aol.com.