Keeping the Scope in Focus

By: Joy Beatty and Karl Wiegers

When discussing requirements, people naturally focus on a system’s functionality. However, exploring requirements needs to begin much earlier, with an understanding of the business requirements for the project. Business requirements include statements of business objectives, success metrics, a product vision statement, scope and limitations, and similar information. This article, adapted from our new book *Software Requirements, 3rd Edition* (Microsoft Press, 2013), describe some aspects of keeping the project’s business objectives and scope in focus to keep the project on track.

**Why Define Scope?**

The project’s scope defines the boundary between what’s in and what’s out. It indicates which subset of a grand ultimate product vision will be implemented during a specific project, release, or iteration. The scope therefore defines a baseline against which proposed changes can be evaluated for possible incorporation. Scope change isn’t a bad thing if it helps you steer the project toward satisfying evolving customer needs, although increasing scope always has a price. You can modify the scope for a future iteration or for an entire project if it’s done consciously, by the right people, for the right business reasons, and with understanding and acceptance of the tradeoffs.

Whenever someone requests a new requirement, the analyst needs to ask, "Is this in scope for the current project or release?" Of course, you cannot even answer that question unless the scope has been defined and agreed to! One response might be that the proposed requirement is clearly out of scope. Perhaps it's interesting, but it should be addressed in a future release or by another project. Another possibility is that the request obviously lies within the defined project scope. You can incorporate new in-scope requirements in the current project if they are of high priority relative to the other requirements that were already committed. Including new requirements often involves making a decision to defer or cancel other planned requirements, unless you're willing to extend the project’s duration.

The third possibility is that the proposed new requirement is out of scope, but it's such a good idea that the project scope should be modified to accommodate it, with corresponding changes
in budget, schedule, and/or staff. That is, there's a feedback loop between the user requirements and the business requirements. This will require that you update the vision and scope document. Keep a record of rejected requirements and why they were rejected; they have a way of reappearing.

**Success Metrics**

It's important to specify the indicators that stakeholders will use to define and measure success on this project. Otherwise, how can you tell if the project has accomplished its intended business outcomes? Identify the factors that have the greatest impact on achieving that success, including factors both within and outside the organization's control. Business objectives sometimes cannot be measured until long after a project is complete.

In other cases, achieving the business objectives might be dependent on projects beyond your current one. However, it's still important to evaluate the success of an individual project. Success metrics indicate whether a project is on track to meet its business objectives. The metrics can be tracked during testing or shortly after product release. Consider a sample project called the Chemical Tracking System. One business objective might be "Reduce time spent ordering chemicals to 10 minutes on 80 percent of chemical orders." This business objective also serves as a success metric because you can measure the average order time during testing or soon after release.

**Using Business Objectives to Make Scoping Decisions**

The business objectives are the most important factor to consider when making scope decisions. Determine which proposed features or user requirements add the most value with respect to the business objectives; schedule those for the early releases. When a stakeholder wants to add functionality, consider how the suggested changes will contribute to achieving the business objectives. For example, a business objective to generate maximum revenue from a kiosk implies the early implementation of features that sell more products or services to the customer. Glitzy features that appeal to only a few technology-hungry customers and don't contribute to the primary business objective shouldn't have high priority.

If possible, quantify the contribution the feature makes towards the business objectives, so that people can make scoping decisions on the basis of facts rather than emotions. Will a specific
feature contribute roughly $1,000, $100,000, or $1,000,000 toward a business objective? When an executive requests a new feature that he thought of over the weekend, you can use quantitative analysis to help determine if adding it is the right business decision.

Assessing the Impact of Scope Changes

When the project's scope increases, the project manager usually will have to renegotiate the planned budget, resources, schedule, and/or staff. Ideally, the original schedule and resources will accommodate a certain amount of change because of thoughtfully included contingency buffers (see Practical Project Initiation: A Handbook with Tools by Karl Wiegers, Microsoft Press, 2007). Otherwise, you'll need to re-plan after requirements changes are approved.

A common consequence of scope change is that completed activities must be reworked in response to the changes. Quality often suffers if the allocated resources or time are not increased when new functionality is added. Documented business requirements make it easier to manage legitimate scope growth as the marketplace or business needs change. They also help a harried project manager to justify saying "no"—or at least "not yet"—when influential people try to stuff more features into an overly constrained project.

Using Business Objectives to Determine Completion

How do you know when you can stop implementing functionality? A business analyst who is intimately familiar with the business objectives can help determine when the desired value has been delivered, implying that the work is done.

If you begin with a clear vision for the solution, and if each release or iteration is scoped to deliver just a portion of the total functionality, then you will be done when you complete the preplanned iterations. The completed iterations should have led to a fully realized product vision that meets the business objectives.

However, particularly in iterative development approaches, the end point might be vague. Within each iteration, scope is defined for that iteration. As the project continues, the backlog of uncompleted work dwindles. It's not always necessary to implement the entire set of remaining functionality. It’s critical to have clear business objectives so that you can move toward satisfying those objectives incrementally as information becomes available. The project
is complete when the success metrics indicate that you have a good chance of meeting the business objectives. Vague business objectives will guarantee an open-ended project with no way to know when you're done. This is expensive and time-consuming. Funding sponsors don't like it because they don't know how to budget, schedule, or plan for such projects. Customers don't like it because they might receive a solution that is delivered on time and on budget but that doesn't provide the value they need. But that might just be the risk of working on products that cannot be clearly defined at the outset, unless you refine the business objectives partway through the project.

Focus on defining clear business objectives, success metrics, and scope definitions for all of your projects. Otherwise, you are just wandering about aimlessly hoping to accomplish something useful without any way to know if you're reaching your destination.

==================================