Four Tips to Manage Scope Creep During Large Technology Implementations

by Jason Schrader

In any transformational, enterprise-level software implementation, scope creep is inevitable. No matter how well you've crafted a plan or developed a mutual understanding between project teams, at some point the originally defined scope will expand. Scope creep, also known as requirement or feature creep, refers to uncontrolled changes or continuous growth in a project's scope.

Here are four actions that implementation project leaders can take to discourage scope creep and minimize its impact.

Clearly Define Deliverables

If deliverables are not appropriately defined, there is a high risk that expectations will not be met. During the project definition phase, clearly define the deliverables and what 'complete' looks like. For each deliverable, define the following:

- What will be done?
- When will it be completed?
- Who will complete it?
- Who will approve it?
- What will the approval cycle be?

Do this early to protect all parties involved and establish alignment on exactly what end users will receive. Share examples of what the completed work will look like, to give both the team developing the software and the end user a tangible standard to work towards.

Consider the Impact of Change

Once the implementation is underway, recognize that changes in scope may be necessary. New findings and details often emerge that did not exist when the project was initially scoped, and they can have a big impact. The project team reserves the right to get smarter and cannot be resistant to necessary change.

When a change is identified, all involved parties may not agree on a course of action. In the event of a disagreement, project leadership must ask the following questions:

- What are the long-term ramifications of accepting or rejecting a change? Does it set a precedent?
- Will this change have an impact on other requirements or commitments?
- Is there a fundamental assumption that is being violated or proven to be incorrect?

Maintain Open Communication

When risks or issues are identified, they must be immediately communicated to the project team, and anything that could impact scope should always be documented and accessible. Communication is key with any stakeholders who will approve or accept deliverables. The more you understand their expectations, the better you will be at managing issues and working with them towards a resolution.

Risks should be reviewed with the project team on at least a bi-weekly basis (adjust accordingly based on the length of the project or time within project). Key risks should also be a part of any status report to the leadership team or even within the project team to keep everyone abreast of the current situation. The team cannot be afraid to document identified risks or issues. Documenting risks does not necessarily mean that they need to be resolved, but they should be acknowledged.

Study Downstream Effects

Myopic thinking cannot exist during software implementations, especially when multiple systems are impacted. When a change is proposed, the downstream impacts of that change must be considered. For example, deciding on a new number format for objects (new and existing) impacts not only the objects within the system being modified, but also other enterprise systems where the information is shared. Imagine changing the length of object numbers in one system from 30 characters to 60 characters. There may be several other downstream systems that still have 30 character maximums for any numbers they process.

Conclusion

While there is no guaranteed way to prevent scope creep, there are a variety of ways to minimize its impact. Clearly define deliverables, consider the impact of changes, maintain open communication, and study downstream effects of system changes to minimize scope creep. Performing these steps will help you achieve your timeline and keep your project sponsors happy. For end users and other stakeholders, controlling project scope delivers a better quality system more quickly.

Originally published on January 9th, 2017

What's your view? Add your question or comment

About the Author

Jason Schrader

jason.schrader@kalypso.com

Jason has over 17 years of experience providing clients with product lifecycle management (PLM) solutions with a focus on the PTC product line.