Top 5 Causes of Scope Creep… and What to do About Them

By Richard Larson, PMP, CBAP and Elizabeth Larson, PMP, CBAP, CSM
Principals, Watermark Learning

Watermark Learning
7300 Metro Blvd, Suite 207
Minneapolis, MN 55439
Phone: 1(800)646-9362
Website: www.watermarklearning.com

Watermark Learning
ENHANCED PERFORMANCE. ENDURING RESULTS.

Abstract

Scope creep is a dreaded thing that can happen on any project, and it can waste money, decrease satisfaction, and not provide the expected project value. Most projects seem to suffer from it, and both project teams and stakeholders are consistently frustrated by it. Why does an effective means of controlling scope seem to continually elude us?

This paper covers the 5 most common causes of scope creep and what project professionals (project managers and business analysts) can do about it. Problems and their symptoms are presented from the standpoint of a project sponsor, and solutions from the project team's perspective. This unique combination provides attendees new insights and ways of controlling scope that can be applied to any type or size of project.

What is Scope Creep, Anyway?

In one of the authors' early days in the Information Technology field in the 1980s, there was no agreed upon use of either the terms scope or scope creep. Today at least, we have some common definitions of both. Here is our working definition of both terms that we will use in this paper.

- **Scope** = the extent of what a project will produce (product scope) and the work needed to produce it (project scope). The PMBOK mentions it is the sum of the products, services, and results produced in a project (PMI, 2008, p. 440). It is often documented using a scope statement and a Work Breakdown Structure (WBS), which are approved by the project sponsor.
- **Scope creep** = adding additional features or functions of a new product, requirements, or work that is not authorized (i.e., beyond the agreed-upon scope).

The PMBOK describes scope creep as "adding features and functionality (project scope) without addressing the effects on time, costs, and resources, or without customer approval." (PMI, 2008, p 440). Change on projects is inevitable, so the possibility for scope creep is also inevitable. Perhaps this is the reason why taming scope creep is so challenging.

We don't mean to imply that additional functionality or work is not desirable on projects. It also does not mean that scope creep occurs just because requirements change. The key part is whether changes are authorized or not. If an expansion of scope is approved, then it is not scope creep.

What is Wrong with Scope Creep?

By working on unapproved features of a product, a project team devotes time to the unauthorized changes. The work to incorporate these changes must usually be done within the original time and budget estimates, leaving less time for approved parts of the scope. That could mean approved features don't get completed, and the end-product is not what was chartered. Or, it can mean that time and cost overruns to finish the authorized parts of the scope will occur.

How Does Scope Creep Occur?

There are many ways scope creep can occur on projects. Executives at the sponsor level frequently don't want to be involved in every decision. So, project teams make them. Some change requests are or appear to be small, so again, project teams act on them instead of following a formal change management process. An inflexible or cumbersome change control process may also contribute to unauthorized scope additions.

For various reasons, the project team may want to exceed expectations and deliver "more value" by adding unrequested functionality. IT managers often fail to negotiate more time and budget when requests for additional functionality are made, and the scope creeps.

Case Study of Scope Creep

- **Overview**: New Energy, Inc. is a fast growing supplier of "green" energy solutions, including wind turbines, solar technology, etc. Their growing sales force and varying sales commission plans prompted the need for a centralized and automated commission system to replace the spreadsheet-intensive procedures they had.
- **Project**: Don is the head of Accounting, and approved the project plan developed by the assigned Project Manager, Mary. Their preliminary research showed that a software package was the most cost-effective solution. Mary's scope statement included these project objectives:
  - Research and produce a list of the 3 packages that best fit the company's need
  - Facilitate a decision for the best package to obtain
  - Procure the package; and
  - Install the package.
- **Progress**: The project took months to research and find the 3 packages. Selecting the package was contentious. The Sales Department wanted a web-based solution, which the VP of Sales, Chris, said would be easiest for the sales people to use. Accounting wanted a product that had tighter security and was easier to customize. Since the system was being paid for with Accounting's budget, Don prevailed.

LinkedIn colleagues cited reasons for scope creep that include:

- Lack of clarity and depth to the original specification document.
- Allowing direct [unmanaged] contact between client and team participants.
- Any scope added to the baseline outside of the Requirements Management Process.
- Customers trying to get extra work "on the cheap."
- Beginning design and development of something before a thorough requirements analysis and cost-benefit analysis has been done.
- Scope "creep where you do it to yourself" because of lack of foresight and planning.
- Poorly defined initial requirements.
- "Management promises the sun and the moon and breaks the backs of the developers to give them that in impossibly tight time frames."
- It is impossible to control scope creep, so always work on the highest priority features.

  (Bleiweiss, A., Bupp, J., Higgins, M., Johnson, D., Meister, J., Murphy, B., Temchin, M., & Oldfield, P, 2009)

- **Pitfall #1**: Once the software contract was signed, Don became busy and largely left the installation to Mary and the technical team. Don wanted the project completed by a fixed date, and Mary felt pressured to get installation done without much requirements analysis, especially given she only had one systems analyst, Bob, on the project.
- **Pitfall #2**: Bob focused his time on converting the spreadsheet data to the new system, and on customizing the many new user interfaces available. He conducted extensive interviews of key staff to learn their needs for the interfaces. As Bob was interviewing, Chris requested that the most common sales interface be duplicated as a web page, and Bob agreed because it was easy. While they were at it, Bob showed Chris a nifty Flash component for the page, which Chris loved, and it was added as a requirement.
- **Pitfall #3**: During the package installation, Sales announced the addition of a new product line that has an entirely different commission structure. Don held a meeting and informed the team that the new commission plan had to be incorporated into the project and that the deadline could not slip.
- **Pitfall #4**: During installation, Bill, a developer, and the Accounts Payable lead, Allison, discovered the project scope did not include changes to other systems like General Ledger (G/L). Bill said he could code something quickly, and it would save Allison's team many hours every payroll. He was ahead of schedule, so Mary agreed. Unfortunately, it took longer to code and test than Bill thought, and consequently he didn't finish testing all the exceptions for the new commission system. The package was implemented on schedule, but with several problems due to some exceptions not working correctly. There were other problems with bad data and incorrect reports. Don was not happy.

Let's explore this example with the reasons why scope creep occurs and what to do about it. We will address most of the preceding factors and causes as we present our list of reasons behind scope creep. Our top 5 list of why scope creep occurs includes:

1. Ambiguous or unrefined scope definition.
2. Lack of any formal scope or requirements management.
3. Inconsistent process for collecting product requirements.
4. Lack of sponsorship and stakeholder involvement.
5. Project length.

For each of our factors, we'll present findings of why the factor causes scope creep, and cite portions of the case study to illustrate. Then we offer some practical ideas that project teams can employ to reduce or eliminate scope creep.

Lack of Scope Definition

Findings

Sponsor View: Projects start with a high-level idea that has been defined and proposed in business terms, not project terms. Sponsors are focused on their customers and competitors, and do not think in terms of "scope" much less "scope management." Likewise, sponsors are typically not familiar with the concept of project charters, which explains why project managers usually create them. Sponsors rarely think their scope is undefined, and they expect others to quickly match their level of understanding. They have been planning and thinking about their business needs, sometimes for months or years, and expect project teams to get up to speed fast.

Project View: The high-level scope is best documented in a project charter and expanded in the scope statement. These high-level views are not sufficient to understand scope until a decomposed deliverable-based Work Breakdown Structure (WBS) is created. Without the detail of a full WBS, a gap in understanding will exist. If the high-level scope is not clarified further with a detailed and validated requirements analysis, then the gap grows wider.

- **Unclear initial scope**. The Sales Commission project has unclear scope definition. Mary did not decompose the scope enough, and the requirements analysis was superficially done. Both of these elements contributed to an undefined scope.
- **Lack of detailed scope**. The team felt pressured into not articulating the scope in detail, often a pitfall of software package installation. Bob focused on the interfaces, which paradoxically opened the door to expanding the scope to include a web page, something clearly out of scope.
- **Poor communication**. Mary did not do an effective job of communicating scope, particularly that the G/L interfaces were out of scope.

If a project skips detailed decomposition and requirements analysis, the scope remains ambiguous. In our experience, the ambiguity is then addressed by design and development teams in one of two inappropriate ways: 1) The team interprets the requirements and builds what it thinks is right, or 2) The team clarifies the scope by eliciting requirements directly from stakeholders and subject matter experts (SMEs). The chances for misinterpretation and expansion of scope are high in these cases.

- Bill and Allison appeared to build what they thought was best, not what was in scope.
- Bob and Chris worked together in an unmanaged way and the Flash component was added for personal reasons and not because it addressed a business need.

Recommendations

1. Clear, well-managed scope is a key element to successful projects. Sponsors can contribute to clear scope by developing their own charters. Our idea of the charter is a short document that contains mainly the business need, the project vision and high-level features in and out of scope. These are all items that executives should be able to articulate on their own.
2. Scope statements should include both features in and out of scope. A robust WBS that decomposes deliverables into work packages is a must. The decomposed deliverables form the basis for beginning detailed requirements analysis.

3. Business analysts can contribute to clear scope with effective requirements elicitation and by analyzing and documenting clear, complete, and concise requirements. This takes time, of course, and the project team needs to plan for the time and be firm about it in the project schedule.
4. When unrealistic deadlines are imposed, the PM needs to work with the sponsor to either negotiate a later delivery date for the project or reduce its scope when the date is fixed.
5. To help clarify the scope early in the project, the systems analyst could have used scope models and diagrams, such as Business Process Models and Use Case Diagrams. They provide a visual aid to clarify scope, leading to a more effective sharing of “mental models” with stakeholders.

Summary: Whether scope is initially unclear or it stays vague as a project unfolds, if the product scope and its underlying requirements are not clear and precise, it is a “breeding ground” for scope creep.

Requirements and Scope Not Managed

Findings

Sponsor view: Once a project scope has been agreed to, sponsors view it like a contract for what will be delivered. Software projects in particular require more learning time and discovery, which often lead to new, previously unknown requirements. These “discovered” requirements can often become scope creep. Sponsors typically do not want to be involved in what they perceive are small decisions, and they may abdicate them without a change management process in place.

Project view: Most projects have a number of requirements that grow as the project progresses. In the day to day discovery and detailing of requirements, it is easy for stray and “rogue” requirements to be added, even without knowing it. Without clear decision-making authority, it is difficult to determine if new requests are in or out of scope. These common problems then occur:

- **Lack of scope management**. Without the control provided by scope management, project teams often make the decisions about whether to add or reject changes to product requirements. This is particularly true when sponsors are not actively involved with a project. Scope then tends to be haphazardly maintained, often in status meetings or when the PM or sponsor happen to hear about it. Bill let Mary know about the new interface requirement during a status meeting.
- **Requirements without context not tracked**. Without the guidance provided by tracing requirements back to business and project objectives, project teams often rely on their own judgment about whether to include individual requirements. This factor becomes even more challenging the closer the relationship between SMEs and business analysts or developers. Relationships such as this can sway decisions about scope changes when scope or requirements management processes are absent. For example, Chris convinced Bob to add a major feature to the project, which was out of scope.
- **Gold-Plating**. Developers may add product features they think are useful or interesting (called “gold plating”). Whether or not changes add value, developers or SMEs should not be making decisions about added functionality. The people authorized to make decisions should be the only ones making the decisions. Bob convinced Chris to try out the nifty new flash component, another bit of scope creep.

- **Scope "kill**." Conversely, if there is a rigid or cumbersome process for changing scope, project managers may inadvertently reject valid scope changes that would benefit the project. We call this "scope kill." Project teams should recommend changes, which the sponsor or other authorized person decides on. Because of the tight deadline, Mary may have rejected the G/L change instead of bringing it to Don if they had a rigid process.

Recommendations

1. Project managers should include a change management process in the Scope Management plan.
2. PMs help control scope by employing effective scope management processes. Every project should have clear roles and responsibilities defined for making scope decisions.
   Scope Management = Ensuring that all new features and functions requested are handled in a way deemed satisfactory by all stakeholders.
3. Business analysts contribute to scope management by reviewing all requirements with stakeholders and getting approval by stakeholders authorized to approve them.
4. BAs should also employ requirements traceability. Traceability is a technique that documents the business need for every requirement by "tracing back" to the business or project objective. It also "traces forward" each requirement to design, test, and construction to ensure approved requirements were implemented. A traceability matrix is a standard tool for supporting this concept (PMI, 2008, p. 111).

Summary: Scope creep occurs when scope or requirements management doesn't occur. Changes to scope need to follow a clear process to prevent haphazard changes. The opposite can also happen, in which project teams prevent changes by strictly enforcing scope and doing what we call "scope kill."

Inconsistent Process for Collecting Requirements

Findings

Sponsor view: Sponsors expect that project teams will follow a consistent method for collecting requirements and don't think much about requirements elicitation. Again, they have dwelt on a particular business problem or opportunity, and they are not eager for the project team to take the time needed to define detailed requirements.

Project view: If scope is not decomposed enough, we have already seen that it can lead to people misinterpreting the high-level scope and adding in extraneous requirements. High-level features and functions tend to have fuzzy process or product boundaries, which in turn can inadvertently lead to:

- **Scope drift**. This means too much scope is included if the fuzzy boundary drifts beyond what was actually intended. (It can also lead to under-defining of scope). In the example, the team omitted any scope modeling, and they opened the door to drifting scope. The boundaries of the solution were not firm enough, allowing for the inclusion of the General Ledger (G/L) work.
- **Too many stakeholders**. Drifting scope can often result in ever-increasing numbers of stakeholders included in requirements elicitation sessions. If included, these additional stakeholders will naturally want to provide their requirements. If the additional stakeholders are not really part of the scope, then their requirements are likely to be scope creep. Chris had a "hidden agenda" and caused the scope to drift into adding a web interface.

Recommendations

1. Establish and follow requirements processes for scope modeling, analysis, prioritization, traceability, and change management.
2. Create scope models early in a project to foster discussion and agreement on scope. Context Diagrams, Use Case Diagrams, SIPOC charts, etc. are simple tools that visually clarify scope. They can also be used to build on as the project's product is progressively elaborated.
3. Using the concept of progressive elaboration, requirements should be collected and analyzed in "layers" to ensure a complete and consistent set. We suggest multiple iterations to do this, using progressively more detailed layers: the scoping layer, an expanded layer, and a detailed layer.

Summary: Often, the boundaries around processes that a project affects are not clear. Fuzzy process boundaries lead to fuzzy scope, unnecessary stakeholders, and extraneous requirements in a project.

Lack of Sponsorship and Stakeholder Involvement

Findings

Sponsor view: Sponsors are typically busy with many initiatives. Like other workers, they usually have too much to do, and are not aware they are not "involved." If a sponsor does not receive regular communication in their preferred mode about a project, he or she may become disinterested. A scheduled milestone or other event like a missed deadline then tends to bring the project back into focus.

Project view: Lack of sponsor and stakeholder involvement both have long been ranked the #1 and 2 reasons for project failure according to research (Standish Group, 2004). Part of the challenge relates to scope creep:

- A disengaged sponsor is more likely to abdicate project decisions to the team. When a vacuum exists, the team, SMEs, and end-users tend to fill it. The less interested the sponsor, the more likely scope creep will occur because the nexus of activity moves away from the sponsor and toward the team. Like many sponsors, Don did not prepare a charter, and relied on his project manager to prepare the project plan. Don also withdrew from the project in the crucial installation stage. He let the project team make decisions in his absence, such as adding the interface to General Ledger.
- Stakeholders often don't devote enough time to project work. Because of ongoing operational duties, SME and end-user involvement with project work frequently suffers. Yet, their input is needed for detailing and reviewing requirements, and for testing. If they don't participate enough, then either a project stalls, or the project team fills the gap. Whenever teams fill voids created by under-involvement, scope creep can occur. Bob might have had more success in engaging his SMEs using interactive techniques like prototyping.

Recommendations

1. Sponsors should develop project charters to keep ownership where it belongs. The sponsor is in the best position to fully articulate the project vision, describe its benefits, and set boundaries.

2. PM: Use project status reporting that engages sponsors, focusing on progress towards the deliverables. Quick, visual displays of status work well.
3. PM and BA: Use a tool like a RACI matrix to manage roles and responsibilities on projects. Use it to clarify decision making and to confirm SME participation.
4. PM: Be sure to include lack of stakeholder participation as a risk, and build in contingencies for the likely occurrence of this risk.

Summary: Lack of sponsorship and stakeholder involvement are two factors that top the list of why projects fail or are challenged. They also contribute to scope creep, which is a significant component of challenged or failed projects.

Length of Project

Findings

Sponsor view: The longer a project runs, the more time sponsors have to refine their ideas, to compare notes with the competition, or simply for the business to change. All of these are potential causes of scope creep, and sponsors don't inherently think in those terms. As sponsors ourselves, we can attest to having added additional requirements to projects the longer the project continues.

Project view:
This final factor is less formal than the others, although experience tells us it is no less real. The factor is:

"The longer a project, the greater the chance for scope creep to occur."

You might also have experienced this same phenomenon. With a large project scope and long project schedules, it opens the door for more and more product features and functions to be added. Actually, this is true even with smaller projects, and it splits pretty evenly who causes the scope creep.

- Over a long project, some of the additional features requested are value-added and should be included, (using a change management process, of course). In the example, the long project delays increased the chances the business would change and provoked Don to add to the scope without changing the deadline.
- Project teams have more time to add unauthorized features with a longer time span. The primary culprits are a) the team being ahead of schedule, with additional time for adding new features or b) an SME asks a developer: "can you make this change 'as long as you are in there.'" Bill was "ahead of schedule," which contributed to his suggesting the expansion of scope and Mary approved it.

Recommendations

1. Educate sponsors to chunk projects into shorter sub-projects and focused on tight deliverables.
2. Decompose projects into smaller sub-projects. The case study could easily have been four separate projects. The installation might have been sub-divided even further.
3. Agile projects address this issue by shortening delivery cycles and tightly controlling scope. Consider this method for appropriate projects.

4. As sub-projects and phases of longer projects finish, formally close them out, conduct lessons learned sessions, and celebrate. This helps maintain momentum and reinforces the results and benefits achieved. That in turn keeps sponsors and other stakeholders engaged.

Summary: Longer projects don't directly cause scope creep. They open the door wider to it when the other components we have addressed in this paper are missing:

- Without a firm initial scope,
- When a requirements management process is lacking,
- When the requirements collection process is inconsistent,
- When sponsors or stakeholders are not engaged.

Summary

Scope creep is dreaded on projects, and it happens for a multitude of reasons. Most of them pertain to the fact that projects bring change, an unpredictable thing. Scope creep is not inevitable, though. Here is a summary of our top 5 reasons for scope creep and a recap of what to do about each of them.

1. **Ambiguous Scope Definition**.
    - Sponsors: Develop charters with specific product features.
    - PM: Create tight scope statements, with features in and out of scope.
    - PM: decompose deliverables into work packages using a WBS.
    - BA: Create scope models to align project team's mental models.
    - BA: Define detailed and complete requirements.
2. **Scope and Requirements not Managed.**
    - PM: include a change management process in the scope management plan and follow them both.
    - BA: Create a Requirements Management Plan to be included in the overall Scope Management Plan and follow it, including the use of requirements traceability.
    - Both: Formally communicate, review, and get all requirements approved; use traceability to manage the process.
3. **Inconsistent Process for Collecting Product Requirements**.
    - Both: Define requirements processes related to scope, analysis, prioritization, traceability, and new requests.
    - BA: Use context diagrams to clarify scope/stakeholders early (e.g., SIPOCs, Context Diagrams, etc.); add detail in layers as appropriate.
    - Both: Define scope and requirements iteratively in "layers" throughout requirements analysis.
4. **Lack of Sponsorship and Stakeholder Involvement**.
    - Sponsor: Develop Charters that communicate desired product features, emphasizing project benefits.
    - PM: Provide project status that engages sponsors, focusing on how deliverables are being realized.
    - Both: Use tools like RACI to get commitment for approvals and for providing input, review, testing, etc.

5. **Project Length**.
    - Sponsor: Keep projects as short as possible and focused.
    - PM: Decompose projects into smaller sub-projects.
    - PM: Close out sub-projects to maintain momentum and show results.

References

Bleiweiss, A., Bupp, J., Higgins, M., Johnson, D., Meister, J., Murphy, B., Temchin, M., Oldfield, P., respondents to a discussion on Linked.com during May, 2009 from http://www.linkedin.com/answers?viewQuestion=&questionID=470012&askerID=245516.

International Institute of Business Analysis (2009). *A Guide to the Business Analysis Body of Knowledge (BABOK®*, Version 2.0. http://theiiba.org.

Project Management Institute (2008). *A Guide to the Project Management Body of Knowledge (PMBOK®) – Fourth Edition*. Newtown Square, PA: Project Management Institute.

The Standish Group (© 2004). Standish Group Chaos Report.