

Given what you learned this week, as project managers, **how do you deal with scope creep?**

The PMBOK describes the process of scope management and project scope with the deliverables of service and process.

The PMBOK also showcases adaptive lifecycles for responding to levels of change (4.6).

- However, when do we get to that tipping point?
- Is there ever a period in which even the most adaptive lifecycle begins to suffer from too many changes in the project scope?
- **Describe how you, as the primary project manager, would fight scope creep in your project and what you would do to ensure the project is delivered on time.**

Remember to use research, citations, and your class readings to bolster your arguments. This is a hypothetical scenario, so feel free to use your Portfolio Project or any other project as the basis for your discussion.

If possible I would make use of an Agile approach to prevent scope creep.

A traditional approach is plan driven. The approach encourages the project team to design a product, with all its features to be delivered, without an awareness of potentially implied requirements, the users' responses, and competitors future actions. Subsequently, some features will likely need to be adjusted or eliminated completely.

On the contrary, Agile frameworks are value-driven and as a result manage change well. In other words, Agile prepares for the *unknown*; traditional approaches plan for the *known*. Agile can only plan for a short timeframe in the form of rolling wave planning. When product delivery is incremental and frequent, teams can account for what is often scope creep, newly added features. Features that generate the most value are developed first to ensure that the client has its current business needs addressed. These features are atomized for more manageable work and grouped in a minimally marketable release.

Agile methodologies vary in how they mitigate against scope change. Scrum tasks the product owner with prioritizing features; the team works on the top ranked features during a timeboxed delivery cycle in which no changes are allowed. In XP the customer

provides the team with more detailed requirements if necessary during each iteration. With Kanban, customers can add feature requests to the team's work queue.

References

Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge*. Project Management Institute.

Sliger, M. (2010, October 12). *Goodbye, scope creep—hello, agile!* Project Management Institute.

<https://www.pmi.org/learning/library/project-scope-creep-agile-methodologies-6552>

Yatzeck, E. (2012, August 14). *How to Control Scope Creep in Agile*. ThoughtWorks.

<https://www.thoughtworks.com/insights/blog/how-control-scope-creep-agile>