

Printable Version: Use the print command from the menu above to print this item.

[Close This Window](#)



Relating PMBOK Practices to Agile Practices - Part 1 of 4

By Michele Sliger

Like the authors of the Capability Maturity Model (CMM), professionals at the Project Management Institute (PMI) state that the Project Management Body of Knowledge (PMBOK) is a guide to best practices and that organizations must use their own discretion when implementing the practices. This is important to note, because the association that many IT professionals have with the PMBOK is that it advocates the waterfall methodology, when in fact it does not. Years of traditional plan-driven software development using waterfall methodologies have created an assumption that a waterfall approach is the only way to adhere to the practices prescribed in the PMBOK. But I've found that many of the PMBOK practices can be followed (albeit it sometimes quite differently) in other methodology models, including those of the agile family. And like the PMBOK guidelines, agile methodologies have their own invalid assumptions to overcome, the most common being that they promote "cowboy coding" and are just shy of total chaos. In fact, agile methodologies, if followed with discipline and rigor, are compliant with CMMI and PMBOK best practices, just as traditional waterfall is compliant. What's different--beyond the obvious command-and-control dictatorship vs. self-organizing teams--is when and how these practices are executed and the new lexicon used by agile practitioners.

The Project Management Process Groups

The first translation is from the PMBOK naming conventions for the project management process groups to Jim Highsmith's Agile Project Management framework is the project management life cycle (Figure 1). The PMBOK identifies initiating, planning, executing, controlling, and closing as the process phases within project management. Highsmith reconfigured these phases to better reflect the reality of how software is truly developed, and defined them as envision, speculate, explore, adapt, and close. The envision phase is where the product vision is defined enough to provide a bordered sandbox within which to work. During speculation, that vision is translated into a set of features and an expected timebox within which to deliver those features. The explore phase is the iterative and incremental development of potentially shippable code. Production transitions into the adapt phase by scheduling stopping points along the way to inspect and adapt both the process and the product. Finally, in the close phase, teams have the opportunity to reflect on their achievements and make decisions based on what they've learned.

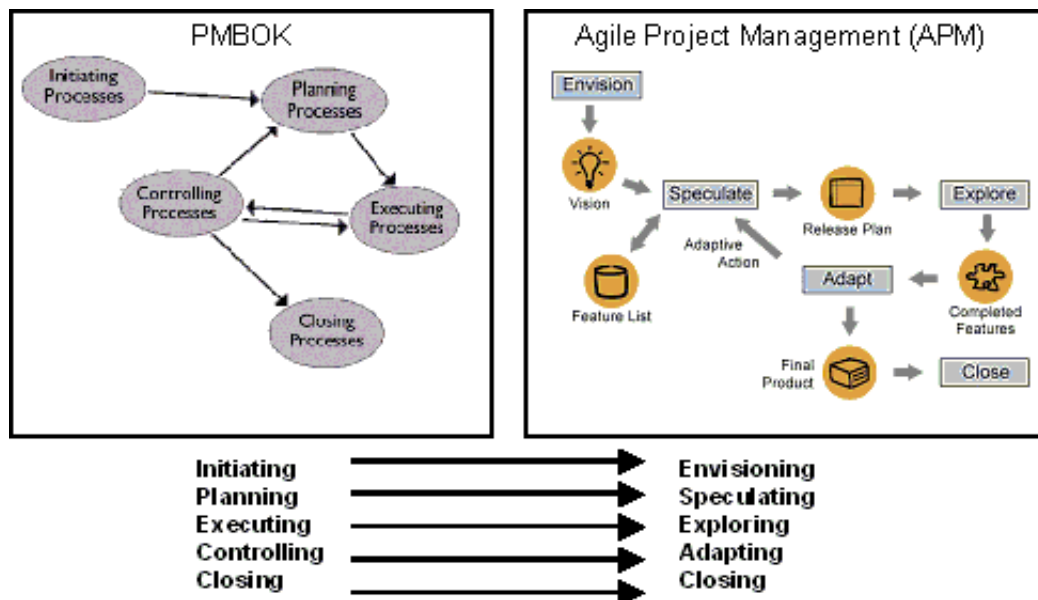


Figure 1: PMBOK Project Management Process Groups mapped to Jim Highsmith's Agile Project Management framework

A further translation of some of the activities performed in each phase is also warranted. Six key knowledge management areas defined in the PMBOK and deserving special attention are Project Integration Management, Project Scope Management and Project Time Management, Project Quality Management and Project Risk Management, and Project Human Resource Management. For each of these areas, the practices advocated by the PMBOK have their cousins in agile, but they are significantly different in their implementation.

Integration Management

A key deliverable in integration management is the project plan document, prepared and owned by the project manager. In agile software development, with its emphasis on just-in-time design and participatory decision making, this activity translates into several different envisioning and planning exercises done on an iterative basis. Rather than defining all of the elements of a project plan at the beginning of the project--scope, work breakdown structure, schedule, assumptions, and controls--the agile project manager will focus on planning using a level of detail that's more appropriate and realistic for the time horizon. Waterfall's big up-front planning translates into agile's continuous planning, where gross-level features and estimates are planned at the beginning of a release, and more detailed planning of tasks and effort is done at the beginning of every 1-4 week iteration. The big up-front documentation effort that went hand-in-hand with the planning is also drastically reduced, and low-ceremony documentation appropriate for the type and number of the communication nodes involved is embraced instead. Low-ceremony, or low-tech documentation of agile release and iteration plans can consist of several white boards in a war room with color-coded markings, or it can be flip-chart paper taped to the wall with brightly colored notes on each. (For help visualizing this, watch Jim Highsmith's free Webinar on agile project management. http://www.rallydev.com/register_for_webinar_series.jsp#jimPresentation.)

Release plans indicating the expected release date and the features to be worked, as well as iteration plans indicating the level of effort in implementing a set of features within the defined timebox, are defined in planning meetings involving the entire team. The team must create, own, and commit to the plan. They cannot be ordered to meet plan specifications written by an individual directing the project. Release plans, iteration plans, and other planning outputs from the envision and speculation phases are then shared with all stakeholders in the most highly visible fashion possible. For co-located teams, this may mean something as simple as posting the plan on the wall. Technical solutions for communication are required for geographically dispersed teams; SharePoint, wikis, and other third-party tools are well-equipped to provide this. Project managers go from writing a large, detailed document defining the plan for the entire project, to facilitating the team in their ongoing iterative planning efforts and sharing that information in the most visible way possible.

Integrated change control also changes dramatically in agile methodologies. In keeping with the idea of minimum process to achieve maximum value, the change control process is streamlined and integrated into the daily routine of agile teams. Product change is managed via the ranked backlog of features. This product backlog is managed by the customer or customer proxy (product manager, subject matter expert), who is responsible for maintaining the list of items to be worked, with the features that provide the most business value to the customer ranked highest.

This backlog contains items beyond functionality requests; technical supporting work and defects are also placed in the backlog and ranked. During release and iteration planning, the highest ranked items move from the backlog into the iterations to be coded, tested, and accepted by the customer. During the iteration, daily 15-minute stand-up meetings are held to apprise each other of status, a key communication element that keeps the team in sync and better able to tackle unforeseen issues.

At the end of each iteration, the working code that was developed is demonstrated, and feedback that may affect future

decisions about the items in the backlog and ranking is gathered from stakeholders. Process changes are also made at the end of the iterations, allowing the team to make course corrections not only in the product, but also in the way they work. The team--customer, developer, tester, analyst, technical writer, and project manager--becomes the equivalent of a change control board. They streamline the process so that decisions are made quickly, collaboratively, and with little to no ceremony.

Want more articles on agile and other best practices? Subscribe to *Better Software* magazine for FREE today at www.BetterSoftware.com/agarts

Further Reading

Highsmith, Jim. *Agile Project Management*. Boston: Pearson Education, 2004.
Project Management Institute. *A Guide to the Project Management Body of Knowledge, Third Edition*. Newtown Square, PA: PMI Publications.

Look for my next article in this four-part series, in which I'll discuss scope and time management.

[Click Here to Read Relating PMBOK Practices to Agile Practices Part 2 of 4](#)

[Click Here to Read Relating PMBOK Practices to Agile Practices Part 3 of 4](#)

[Click Here to Read Relating PMBOK Practices to Agile Practices Part 4 of 4](#)

About the Author

Michele Sliger has worked in software development for over fifteen years. She currently works as an agile coach for Rally Software Development, where she trains software development teams in agile methodologies. Sliger has served as a project manager at Qwest Communications, a consultant for Fortune 500 companies, and was a founding member of the engineering teams at two biotech start-ups. She is a certified Project Management Professional and a Certified Scrum Master. In addition to her service at Rally, Sliger is also an adjunct faculty member of the University of Colorado, where she teaches Software Project Management to graduate engineering students.

StickyMinds.com Weekly Column From 02/20/2006

[Close This Window](#)

[Home](#) | [Resources](#) | [Topics](#) | [Community](#) | [PowerPass](#)

© 2010 StickyMinds.com. All rights reserved.

StickyMinds.com is a division of [Software Quality Engineering](#).
[Privacy Policy](#) [Terms & Conditions](#) [Link to StickyMinds.com](#) [Feedback](#)