

## PMBOK Approach Insights for Software Development Projects

Dr. Ronald J. Wojcik, PMP  
Principal Consultant  
Pragmatic! Solutions  
ronw@pragmatic--solutions.com

### Introduction:

The Project Management Book of Knowledge (PMBOK)<sup>1</sup> provides a broadly accepted world standard for the project management of any project type. The PMBOK model encompasses 44 different processes that interweave with one another. Each process is organized into a three stage flow, with inputs being processed using techniques and tools appropriate to the activity and project stage to generate outputs that are typically project artifacts as well as inputs to subsequent processes.

For a practitioner who is not schooled in this approach, the sheer size of the PMBOK can be daunting. Furthermore, in the case of a software development project, the industry has placed much greater focus on software development life cycle (SDLC) methodology. This paper focuses on the project management of a software development process. It “breaks down” the full PMBOK approach to the critical inputs, processing steps and outputs that most impact a software project manager.

The net effect is to reduce overall complexity in order to enable the process basics and their flows/interdependencies to be quickly grasped. All of the major process groups, from the Initiating group through the Planning, Executing, and Monitoring & Controlling groups to the Closing group, are then addressed in detail, specifically in the context of a software development project. Key insights derived from application of the PMBOK approach to software development are noted for each major process group. Furthermore, the sales process, not within the PMBOK scope, is also discussed, given the constraints that sales approaches typically place on a software development project manager prior to project initiation.

### Getting Started: A Thumbnail Sketch of the PMBOK Approach

Figure 1 below provides a break down of the overall PMBOK approach. In particular, it highlights the crucial outputs that a software development project manager needs to pay special attention to. The “Processing” column is accomplished using appropriate tools and techniques as spelled out in the PMBOK.

While there are myriad PMBOK inputs, tools, techniques and outputs, for software development projects, critical attention must be paid to a limited subset, namely:

- Initiating: Project Charter (includes Preliminary Scope Statement)
- Planning: Project Plan (includes Activities, Resources, Schedule, Cost & Risk)
- Executing: Functional Delivery Plans (consistent with SDLC Methodology)
- Monitoring & Controlling: Performance Reporting & Integrated Change Control
- Closing: Acceptance Criteria

The following figure captures this view.

---

<sup>1</sup> A Guide to the Project Management Body of Knowledge, Third Edition, ANSI/PMI 99-001-2004

## Simplified Capture of the PMBoK Approach

#	Process Group	Inputs	Processing	Outputs	Stakeholders
	Selling (Non-PMBoK)	Customer Needs RFX	Proposal Preparation High-Level Effort, Schedule, Risks Negotiations	Contract Solution Architecture	Sales Manager – Owner Senior Mgmt Project Manager Dev Manager
2	Initiating	Contract Preliminary Scope Statement	Detailed Analysis Contract Clarifications	PM Assigned Project Charter Constraints Assumptions	Project Manager – Owner Development Manager(s)
21	Planning	Contract Customer Requirements <i>Plan Updates</i>	Contract Review Work Breakdown Staffing Plan Detailed Schedule Risk Assessment .....	Project Plan	Project Manager - Owner Development Manager(s)
7	Executing	Customer ATP Delivery Artifacts as Execution Proceeds <i>Corrective Actions</i>	Specifications Project Development Integration Test System Test Site Test Patch Deliveries	Solution Development & Delivery & Validation & Acceptance Plans <i>Execution Results</i>	Project Manager – Owner Project Team Development Manager(s)
12	Monitoring & Controlling	<i>Execution Results</i>	Rich Collection of Reviews <i>Performance Reporting</i>	<i>Corrective Actions</i> <i>Project Plan Updates</i>	Project Manager – Owner
2	Closing	Final Software Delivery	Defect Closure Plan	Customer Acceptance	Project Manager – Owner

44


1

### Project Initiation:

The key Initiation input is the Preliminary Scope Statement, which provides the Customer's view of the project scope. If this scope has been formally captured as the Statement of Work (SOW) in a Contract, then significant constraints and assumptions have already been placed on the project. We will revisit this concern at the end of the paper to discuss steps that can be taken to mitigate problems and minimize issues that can arise when this is the case.

For a software development project, the Project Charter should capture both the scope of the project and the game plan for its pursuit. The latter typically identifies the SDLC methodology practices that will be followed, including any tailoring for this particular project. Personal experience has shown it best to conduct a kickoff meeting to review this charter to ensure all internal parties have a common understanding. The following figure illustrates its typical content.

Project Charter Components	
Project Charter Outline	
<i>Customer Background &amp; Motivation</i>	
<i>Project Scope</i>	
<i>Major Project Milestones</i>	
Solution Architecture	
Specifications Approach & Status	
Design/Development Challenges	
Testing Strategy	
Resource Requirements	
<i>Risks &amp; Mitigation Planning</i>	
<i>Constraints &amp; Assumptions</i>	
Project Economics	
<i>Project Communications Plan</i>	
Project Methodology Review	
Major Assumptions / Open Items	

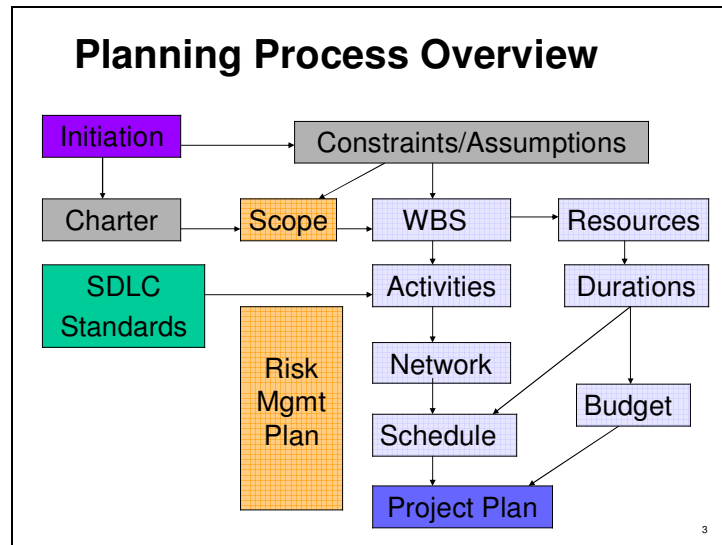


*Components in italics reviewed with Customer*

9

### Project Planning:

The Planning process group encompasses 21 different processes. The following figure emphasizes the deliverables associated with the key processes that drive the planning effort to its culmination in a Project Plan. The project plan is a detailed synopsis of the various development activities, their dependencies, their estimated durations, and the specific resources that are required for their execution. In many cases, this information is captured in Microsoft Project as the initial detailed plan for the development is established, and serves as a baseline for later monitoring. The project cost is also typically derived from this information.



In the process of creating the above detailed plan, the work breakdown structure reflects the major development pieces. These can usually be mapped to software modules. The activity decomposition often reflects the SDLC methodology, with each major software component spawning activities for requirements allocation, design, coding, unit test, and integration into a system-testable deliverable.

For success in this phase, a clear definitive project scope is crucial. While some development organizations think vague requirements are an advantage, vague specifications typically create enormous frustration for both customer and supplier, especially in fixed price contracts.

Also, it should be understood that the planning process above is highly iterative. This is the time when the project manager sows the seeds of success or failure. Techniques to optimize potential plans, for example resource leveling, crashing and fast tracking are often used. Risk analysis, avoidance and mitigation must be carefully performed as these techniques are exercised. A specific example illustrating the Microsoft Project output that results from this approach for a simple but non-trivial software project with six modules is available from the author.

### Project Execution:

Execution is the project phase where the bulk of the work is performed. Execution is performed by the project team; its organization can be functional or project-centric. PMBOK is silent on the procedures for how a project is executed.

The deliverables identified in the Project Plan are produced per the agreed upon methodology of the organization. This methodology drives the project plan deliverables, and is the place where the SDLC and PMBOK approaches particularly complement one another. Within the PMBOK

approach, Quality Assurance Performance addresses the project mechanism for review and continuous process improvement activities. This should be consistent with the organization's SDLC procedures.

Project Monitoring and Control:

Once the project plan has been established, the project manager spends most of their energy and time in the Control phase, the second focus of the admonition, "Plan the Work. Work the Plan!"

To monitor a complex project, it is essential that excellent communications are in place. Given this, the project manager is often best served by a "lightweight" approach so that accurate status can be determined without placing an undue burden on the people actually doing the work. Definitive deliverables go a long way towards facilitating this approach.

In addition, it is important to have quantitative measures of status. This is often a major challenge, as people often respond to status questions with "I feel we're in a good place." or "I think we're going to be fine" or the proverbial "We're 95% complete", the latter sometimes being the status for a month or longer.

While Microsoft Project and other tools provide mechanisms for tracking detailed project status, considerable effort can be required and sometimes place too heavy a burden on the team. As an alternative, the baselined project schedule for major project deliverables can be used to provide a lightweight monitoring approach that is nearly as effective. A sample output for the lightweight approach follows:

Stage	Activity	Dates			Items		
		Baseline	Current	Actual	Done	Left	Late
Planning	Draft Customer Requirements	1/19/06	1/19/06	1/19/06	100%		
Planning	WBS Structure	2/07/06	2/07/06	2/07/06	100%		
Analysis	Approved Customer Requirements	2/14/06	2/14/06	2/17/06	100%		
Analysis	Baseline Project Plan	2/17/06	2/17/06	2/17/06	100%		
Design	Draft Functional Specs	3/10/06	3/10/06	3/14/06	100%		
Design	Approved Functional Specs.	3/20/06	3/20/06	3/20/06	100%		
Design	Design Specs. Approved	3/27/06	4/03/06	4/06/06	100%		
Develop	Project Development Delivered	5/22/06	5/29/06		50%	30%	20%
Develop	Integration Complete	5/26/06	6/02/06				
Testing	System Test Planning	5/16/06	5/16/06		80%	20%	
Testing	System Test Execution	6/30/06	6/30/06				
Delivery	Package Documentation	6/26/06	6/26/06				
Delivery	User & Technical Documentation	6/26/06	6/26/06				
Delivery	Software Delivery	7/03/06	7/03/06				
User Test	Customer Installation	7/05/06	7/05/06				
User Test	Customer Acceptance Testing	7/06/06	8/01/06				
User Test	Acceptance & Production Cutover	8/02/06	8/02/06				

This monitoring, when combined with traditional reporting, such as Project Progress, Risk Mitigation Status, Change Request Status and Budget Tracking, can enable Senior Management to quickly and succinctly understand project status and be able to probe areas where further discussion is warranted.

While monitoring is crucial, change control is equally important. It should also be clear that change control is management of change, not simply prevention of change. It also needs to be

understood that tradeoff analysis must be in place so that the impact of a change on the project is explicit and properly managed. Too often, such interactions result in statements that the Customer will accept the added risk; this seldom turns out to really be the case when problems arise.

A mechanism that has customer buy-in and support is essential. The effectiveness of such a mechanism is greatly facilitated when detailed specifications, chained to customer requirements and acceptance tests, have been created. Change control is often an issue in initial projects between Customer and Supplier, where common expectations between the parties have yet to be established in practice.

#### Project Closeout:

Closeout on a project involves two major elements. First and primary is the Customer's acceptance of the project. The best advice here is to ensure that success criteria are explicit, genuinely commonly understood, agreed on in advance, and quantitative. Groundwork laid at project initiation pays off handsomely at this stage.

The other major element is the conduct of key learning sessions, ideally jointly with the Customer. Frustrating experiences during a project are almost universally a two-way street, whether the customer is internal or external. Insights from key learning sessions can not only improve future project efforts and the general SDLC procedures; if conducted in an open and constructive manner, they can strengthen the relationship between the Customer and the Supplier as both parties see the positive impacts that refinements and improvements can effect. This can bring substantial business value long-term.

#### Sales Process:

While the sales process is outside the scope of the PMBOK, in most cases, a project manager is well advised to participate in the sales process and ensure that sufficient input from project management and development groups is incorporated into it. Often "high-level" schedules and costing are assumed in the Sales Process. Unfortunately, these typically become either non-negotiable or difficult at best to change, once the project has been initiated.

For most significant projects, the planning process should be executed at a high-level with sufficient contingency to ensure the resulting Contract terms or project expectations are achievable. Major scope and requirements assumptions, project constraints, first cut architecture, resource availability and tradeoff assessments, ideally in dialogue with the Customer's business and technical staff, is a wise investment. This is also the place to ensure that acceptance criteria are defined and acceptable to all.

The key insight here is that Sales Processes often blur the line where Project Initiation truly begins. The aforementioned steps can help mitigate this. The project manager needs to be a full participant, and in fact is a key stakeholder, in the Sales Process when this is the case.

#### Summary:

Approaches for both project management and software development have typically been independently addressed. The approaches in fact can explicitly complement one another, and at the same time reinforce each other in areas where one of the two is silent. Recognizing the need for addressing sales process realities further places the management of software development projects on a solid base and improves the chances for project success.