PharmaSUG 2019 - Paper LD-288

Project Management Fundamentals for Programmers and Statisticians

Jennifer Sniadecki, Ultragenyx

ABSTRACT

In the clinical trials environment, nearly everyone from individual contributors to team leaders and managers have the need to utilize project management skills. Because most programmers and statisticians lack formal training in the project management field, it can be challenging to apply and effectively integrate project management principles as part of their day-to-day job duties. This paper will extract the fundamental concepts from *A Guide to the Project Management Body of Knowledge* (*PMBOK® Guide)* that are most relevant to programmers and statisticians and provide real world examples where applicable.

INTRODUCTION

Starting in the mid-20th century, project managers began to lobby for project management to become a veritable profession. One stepping stone in this quest was obtaining agreement on the content of the body of knowledge (BOK) that is known as 'project management'. As a result, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* was produced. The *PMBOK® Guide*, now in its sixth edition, is generally recognized as good practice for the field.

Project management is comprised of extensive guidelines, techniques, standards, definitions, mathematical concepts, communication skills, and leadership elements. Many programmers and statisticians work, at best, tangentially with project managers, and others work, at worst, without any substantial exposure to project managers on a regular basis. The understanding and appreciation for the total breadth of the field can be lacking. In fact, it may not be unusual for non-project management professionals to falsely believe that project management is simply the mastery Microsoft Project!

This paper is presented independent of any project management or scheduling software packages. Instead, it focuses on the key ideas that can help assist you individually as a programmer or statistician without the assumption of a background in project management. Regardless if you are an individual contributor who sometimes struggles with competing priorities, or if you are a lead or manager who is responsible for resourcing decisions and projecting timelines months in advance, the techniques and tips presented below will provide you with a framework for how to more effectively manage your project.

COMMON TERMINOLOGY AND BASIC CONCEPTS

As with any profession, it is essential that we all speak the same language, and project management is no exception. Before proceeding any further, let's establish some basic definitions and examine a few elementary concepts, beginning with the most basic concept of…what is a *project*?

PROJECT

A project is a *temporary* endeavor undertaken to create a *unique* product, service, or result. Since projects are temporary in nature, they must have a defined beginning and end. Developing a single pharmaceutical drug can be thought of as a project. It has a start point (at discovery) and a corresponding end point (either when the drug gets brought to the marketplace or at failure somewhere along the drug discovery process), and it clearly creates a unique product. However, projects can also be much more granular and simpler in nature. An example would be developing a single document, such as a PharmaSUG paper. It is temporary in nature, as initiation is at the abstract submission, and the project concludes at the deadline set forth for final submission. Once again, a unique result is produced.

LIFE CYCLES

The environment in which a project operates is known as its *life cycle*. Until the end of the last millennium, most projects operated under a single approach known as a *predictive* life cycle. This approach is fully plan-driven and fixed. It relies heavily on up-front planning and changes are constrained as much as possible. At the opposite end of the spectrum reside *agile* life cycles. They developed out of the need to function efficiently within an ever-changing environment (such as the within the software industry). Planning, execution, and testing are fluid and adaptable. The specifications may not always be known early and are likely to evolve. In between these extremes are both *iterative* and *incremental* life cycles. The characteristics of each are presented in Table 1: The Continuum of Project Life Cycles.

For example, suppose that you hire a contractor to help you redecorate a room in your house. Working within a predictive approach, you would provide all your personal preferences to the contractor before any work begins (e.g., wall paint color, floor type, window treatment patterns, etc.). The contractor would provide you with a cost and timeline estimate. Once an agreement is reached, the contractor then follows your specifications and completes the room. The advantage with this approach is that you have a high degree of confidence in the cost and timeline before the work begins. However, what happens if after the work is complete, that you realize you don't like the wall color? You will have to pay the contractor to repaint it. Then once it is repainted with a new color, what if you want to change the design of the window treatments? You can see that a predictive life cycle works well if the plan is well-defined, but changes can be extremely costly in terms of timing and financial implications.

Contrast this with a more agile approach of breaking down the redecoration into smaller steps. Working within this approach, you will have the contractor paint one wall first to ensure the color looks as expected. If you like it, the project proceeds, and if not, you can test another color before advancing with other tasks. Because you see the progress as it happens, changes are much easier as manage. Agile environments embrace change and flexibility. They rely on short, iterative planning and execution cycles. However, they require a high degree of stakeholder involvement to refine work products with frequent delivery.

Table 1: The Continuum of Project Life Cycles

	Predictive	Iterative Incremental	Agile
Requirements	Are defined up-front before development begins	Can be elaborated at periodic intervals during delivery	Are elaborated frequently during delivery
Delivery	Deliver plans for the eventual deliverable. Then deliverable only a single final product at end of project timeline	Can be divided into subsets of the overall product	Occurs frequently with customer-valued subsets of the overall product
Change	Is constrained as much as possible	Is incorporated at periodic intervals	Is incorporated in real-time during delivery
Key stakeholders	Are involved at specific milestones	Are regularly involved	Are continuously involved
Risk and cost	Are controlled by detailed planning of mostly knowable considerations	Are controlled by progressively elaborating the plans with new information	Are controlled as requirements and constraints emerge

Reprinted with permission of PMI from A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Sixth Edition Copyright 2017. Project Management Institute, Inc.

An iterative life cycle allows for feedback of in-process work to be incorporated as to improve and modify that work. An example of an iterative process can be thought of when a custom, post-hoc figure is first prepared for an upcoming presentation. In this scenario, there will be requirements created in advance. However, the team will likely want to review a draft version of the figure and provide their feedback before it is finalized. This process could necessitate a few, or many, iterations until all stakeholders agree and are satisfied with the content, design, and message portrayed by the figure. Each of these iteration cycles provides improvements to the final figure.

An incremental life cycle provides finished deliverables in batches. Each batch successively adds functionality. The deliverable is considered only complete after the final iteration. One way to think about an incremental approach is if a package of safety and efficacy TFLs (table, figures and listings) are required for a manuscript, and the efficacy outputs contain a fair amount of complex statistical analysis that take longer to create and validate. In this case, the safety portion of the TFLs could be delivered first so as not to hold up the medical writers from initiating the composition of the manuscript. The efficacy batch would then be delivered later and thus complete the entire deliverable.

Since we as programmers and statisticians operate within an environment that upholds clear procedures and necessitates requirements with certainty (e.g., SOPs, SAPs, Protocols, etc.), most of our projects reside on the left side of Table 1. Even though a large volume of our work has defined deliverables within firm timelines, it is possible that the planning or execution of supporting tasks or partial deliverables will reside within incremental or iterative approaches.

BASELINE

A *baseline* in project management is a clearly defined starting point. It is a fixed reference point for the measurement and comparison of your project's progress. It is utilized within predictive lifecycles where low levels of change are expected. For example, if your project is slated to finish in five weeks, is this good or bad? If your schedule baseline has a four-week completion plan, you can tell there is a problem and your team will need to work faster. A project baseline typically encompasses components from schedule, cost, and scope. When these three components are integrated, this is known as a *performance measurement baseline* (PMB). The performance measurement baseline is a tool that you can use to monitor and manage how a change to one baseline affects the others. For example, knowing that you need to accelerate the schedule to get the work done one week early will likely have an impact on both scope (possible reduction) and/or cost (possible increase). Baselines should be documented and controlled. It becomes more difficult to effectively measure progress when frequent changes to baselines occur.

KNOWLEDGE AREAS

The *PMBOK® Guide* defines ten Knowledge Areas as depicted in Figure 1: Ten Project Management Knowledge Areas. Each is an area of project management that is defined by its requirements and is described in terms of its processes, practices, inputs, outputs, tools, and techniques. These ten Knowledge Areas are used in *most projects most of the time*; hence they may not be applicable to all projects every time. They are interdependent, but they are often presented academically as discrete topics. With a combination of everyday life examples and the support of a hypothetical working project as an example, the remainder of this paper will touch on each one of the ten Knowledge Areas.

ILLUSTRATION THROUGH EXAMPLE

In order to provide a grounded and substantive orientation to the ten Knowledge Areas, we need a framework defined to help illustrate and apply ideas. Bear in mind that, as a programmer or statistician, you are a specialist in your field and not a project manager. You are likely reading this paper because you have found yourself as an 'accidental project manager' who requires a basic understanding of selected principles within the project management field to help you succeed. Success in this case can be measured by knowing you are on the right track with how you set-up your current projects. It can also be measured by learning a completely new concept that you can apply to your daily life, or it may be just the increased awareness of the project management profession.

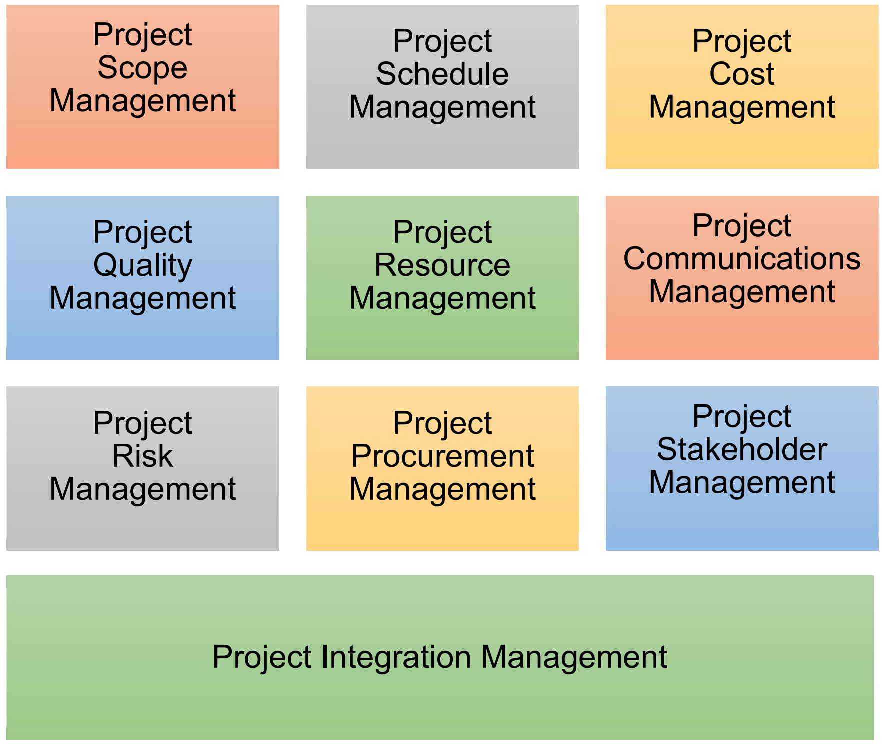

Figure 1: Ten Project Management Knowledge Areas

Let's imagine that you lead a small team of programmers and statisticians. Based on a recent planning meeting, you are notified that a database lock is approaching, and your team is expected to deliver the TFLs and CRT (Case Report Tabulation) package within 60 calendar days from the database lock. You have a project manager who is assigned to oversee the clinical trial in its entirety. However, the project manager comes to you and notifies you that at the next planning meeting, which will be held in a few days' time, you are expected to voice any concerns that you have with the execution of this timeline from your team's perspective.

As you hear this news, a slight panic begins to set in as you realize that you are at a loss with how to assess the feasibility of what has been asked of your team. You recognize that your responsibility is to create a pathway for your team to succeed, and that the responsibly of adherence to the timelines and the quality of the work rests on your shoulders. You feel there are too many unknowns to have much confidence so early in a large undertaking.

Fortunately, you can utilize techniques from the Knowledge Areas to help you develop not only a solid foundation of what needs to be accomplished, but also a mechanism by which you can monitor the progress of your team once the work begins. Your anxiety will hopefully be alleviated once you turn many of the unknowns into a working plan, including the early identification of risks and a well thought out communication strategy.

For this example, you will focus on the needs of your team. Thus, you will define the project as the narrowed scope of only what *your* team needs to accomplish to fulfill the requirements of meeting the required deliverable. This this upholds the definition of a project, as it is a temporary endeavor (60 days) with a unique product (the creation of TFLs and a CRT package).

You also know that you are operating within a predictive life cycle (as based on the *PMBOK® Guide*), where the scope, schedule, and cost are determined early in the life cycle. Scope is defined by the study protocol, SAP, and regulatory agency requirements. The timeline is also driven by a combination of your company's' internal needs and possibly any requirements from regulatory agencies. For simplicity, we will assume that cost is fixed for your role.

There is no set order in which the Knowledge Areas should be approached. Remember that some projects may not require all Knowledge Areas. For this example, I recommended to focus first on scope, schedule, stakeholders, and communication. Planning will likely be an iterative process. It is natural to think about each Knowledge Area separately, but it is highly probable that some knowledge areas will need to be revisited as planning becomes more elaborate and nears completion.

SCOPE

The *scope* is a detailed definition of the project. It collects what is included and excluded from the project. Not surprisingly, it is the foundation needed to perform accurate scheduling and time estimations. Defining what is included in the scope is also essential to prevent *scope creep*. Scope creep is the uncontrolled expansion to the project without adjustments to time, cost, and resources. Scope creep can negatively impact quality, overload the project team, and jeopardize the entire deliverable.

Defining the scope can be a highly iterative process. The first iteration may involve identifying the major deliverables, while the next iteration would work on eliciting more details. Each successive iteration would then describe with greater specificity the requirements as they become known. Fundamentally, this can be considered a simple two-step process. This first step is to list all requirements, ensuring nothing is omitted. The second step is to develop a hierarchy of those requirements. In practice, collecting all requirements and subdividing them into similar groupings often happens seamlessly instead of as two discrete steps.

For example, consider the task of having to prepare a traditional Thanksgiving dinner for some family and friends. Before you set out to the grocery store to purchase the needed ingredients, you will create a shopping list based on a planned menu. Your menu is likely organized by entrees (e.g., turkey, ham), side dishes (e.g., mashed potatoes, green bean casserole, rolls), desserts (e.g., apple pie, pumpkin pie) and beverages (e.g., soda, wine, coffee). By creating a menu to organize your needs, you have developed a hierarchy.

The same principles apply to our 60-day working example. First, you will need to create a list of everything required from your team. This list should encompass the *deliverables*, not the activities, required from your stakeholders. As discussed above, this project is operating within a predictive lifecycle. And you know that in a predictive life cycle, the requirements are known. As the team leader, you may have a very good, if not entirely accurate, understanding of all the requirements. However, it is also entirely possible that you may also not know everything that is required by your team. Regardless which category applies to your situation, you should plan to meet with your stakeholders to confirm exactly what is needed by them so there are no unexpected surprises later.

Let us suppose that your high-level (early iteration) list includes the following: SDTM datasets, ADaM datasets, all TFLs in the SAP, a placeholder for post-hoc TFLs, both the SDTM and ADaM reviewer's guides, metadata, Pinnacle 21 reports, Define-XML, subject profiles, and annotated case report forms. As mentioned at the top of this section, the scope will clearly define what is excluded. In this case, we will exclude all PK analyses from this project, as these are performed by the pharmacokinetic group and considered out of scope for your statistical team.

Now, as you refine your list, think about organizing it into a deliverable-oriented hierarchy. In project management, this hierarchy is known as the *Work Breakdown Structure* (WBS). The WBS contains every piece of work required. As one moves from the top to the bottom, the project work is broken down into smaller more manageable parts. This technique of dividing the scope and deliverables into smaller parts is known as *decomposition.*

An abbreviated WBS is shown below in Figure 2: Partial Work Breakdown Structure. The top level shows a breakdown between datasets and TFLs, with the illustration of decomposition occurring within the datasets branch. The datasets are further divided at the next branch between SDTM and ADaMs. Decomposition continues down the ADaM branch that further details its associated deliverables.

How you choose to organize your hierarchy is completely up to you. Just as in planning Thanksgiving dinner, instead of a breakdown by course, it could have been organized by vegan, vegetarian, and meat/dairy choices. For our working example, it can be divided at the top between source and validation work. It can also be organized by deliverables based on your team structure. The options are endless, but the main importance is that all work is included.

The *8/80 rule* is a good rule of thumb for knowing when to stop the decomposition. This rule suggests stopping when tasks become shorter than 8 hours and keep going if the task takes more than 80 hours. The 8/80 rule prevents the creation of too small a task, which can lead to micromanaging. It also protects against the creation of a work package that is too large to easily manage.

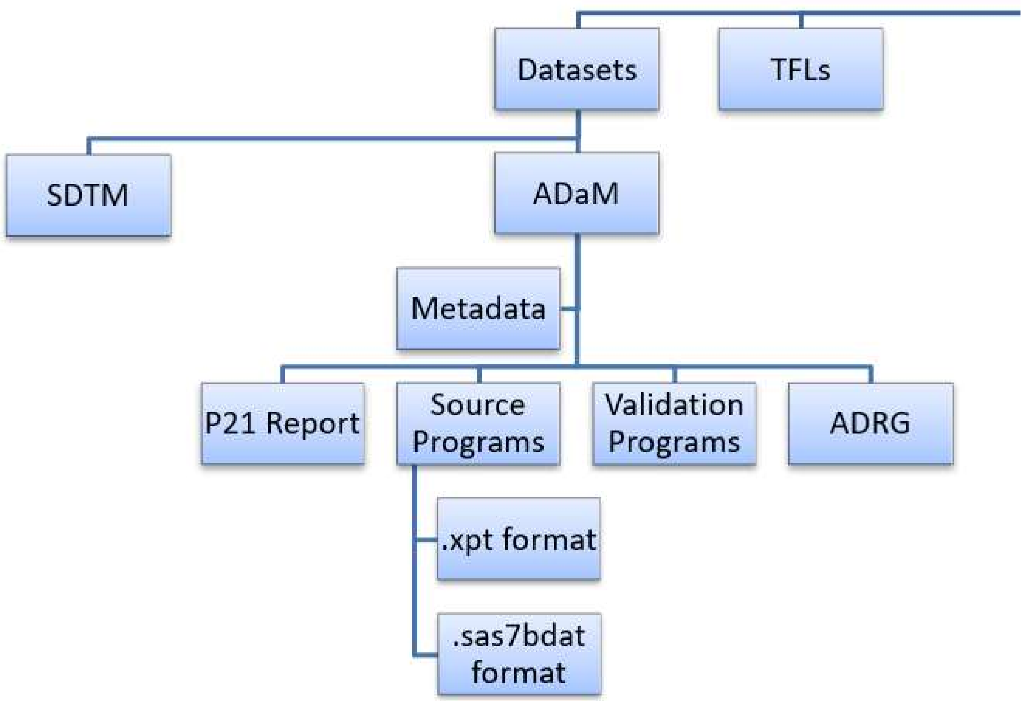

Figure 2: Partial Work Breakdown Structure

Lastly, when starting out in the initial planning stage, consider creating an *assumption log*. An assumption log is a record of all constraints and assumptions throughout the project. It spans all Knowledge Areas, and it is used as the official record for the assumptions about the project, the environment, and the stakeholders. The larger the project and the higher frequency that a project undergoes changes (especially to the PMB), the more important the assumption log will be in assessing an impact to those changes. A few assumptions that would be recorded under this Knowledge Area include the Implementation Guide (IG) versions, dictionary coding versions, and the version of the software used to support the project (with the assumption that the software has been properly tested and qualified).

SCHEDULE

DEFINE YOUR ACTIVITIES

Once the requirements are identified and the scope is defined, the next step is to arrange the schedule of activities for your team. In order to do this, you will need to define all the *activities* associated with your project. Activities are the specific *actions* required to complete the project work. This can seem overwhelming at first, but it quickly becomes more manageable once it is broken down into smaller steps. In order to begin, it is best to start with the lowest levels of the Work Breakdown Structure created above and think about the actions needed to create those deliverables.

For example, let us select the ADRG (Analysis Data Reviewer's Guide) deliverable that displayed in the lower right in Figure 2: Partial Work Breakdown Structure. Some likely activities to support this deliverable include obtaining the latest ADRG template, generating the Pinnacle 21 (P21) conformance report, creating a first draft of the ADRG, having a thorough review of the content, and verifying all the hyperlinks work as expected. Activity lists, as shown in Figure 3, are different from the Work Breakdown Structure in that they will contain action verbs (e.g., generate, verify, gather, etc.).

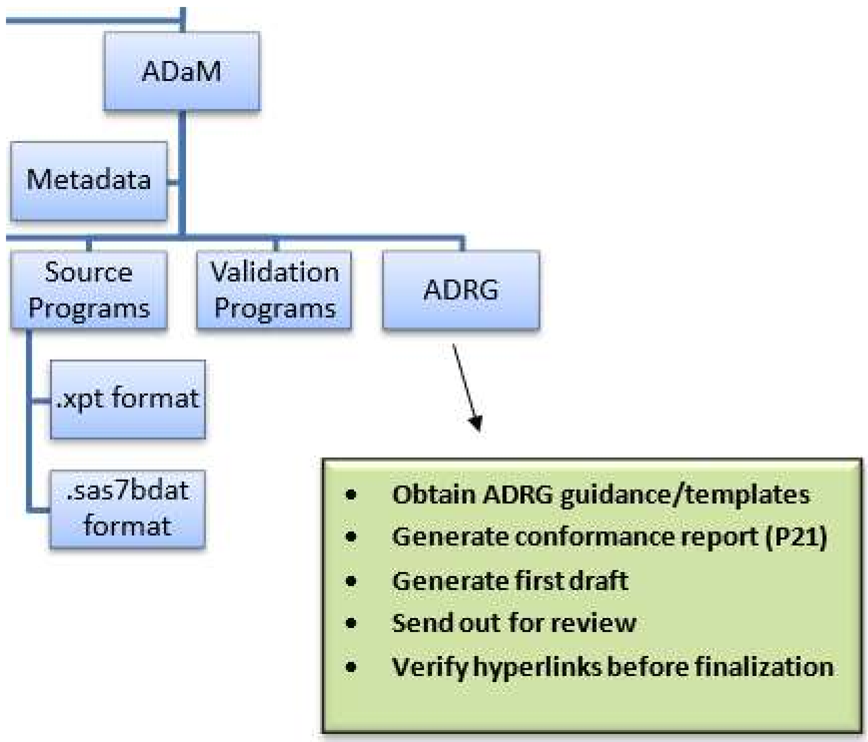

Figure 3: Example of an Activity List

SEQUENCE YOUR ACTIVITIES

Once you have an activity list created for each of your deliverables, the next step is to sequence them. In project management, a technique used for analyzing the logical relationships between activities is known as the *precedence diagramming method* (PDM). PDM will be utilized throughout the remainder of this section to produce *project network diagrams*. Project network diagrams are simply a visual representation of the dependencies between two activities or milestones.

Project network diagrams are beneficial because they show the interdependencies of the activities. They help to identify areas where the schedule may be compressed, and they can help justify the time estimate for your project. PDM and project network diagramming can be extensive topics. Definitions are provided here to help orient you to the basic concepts. Examples of project network diagrams are shown below in Figures 5 and 6.

One concept central to sequencing involves the logical relationships that can exist between two activities. Understanding logical relationships will help you determine the order in which your activities need to be performed. The logical relationships are built upon a combination of what are known as *predecessor* (those that come first) and *successor* (those that come after) activities.

The four types of logical relationships are:

1. Finish to Start (FS) Relationship – Start Event of a Successor is dependent on the Finish event of Predecessor. This is the most common relationship.
2. Start to Start (SS) Relationship – Start Event of a Successor is dependent on the Start event of Predecessor.
3. Finish to Finish (FF) Relationship – Finish Event of a Successor is dependent on the Finish event of Predecessor.
4. Start to Finish (SF) Relationship – Finish Event of a Successor is dependent on the Start event of Predecessor. This is the least common relationship.

To better understand this, let's examine possible scenarios for the first three relationships. The fourth relationship, Start to Finish, is rarely encountered in most projects. To note, these relationships and the terminology presented here are commonly used in project management software packages.

Finish to Start is the most common (it is the default relationship in Microsoft Project). It is used to establish a sequential flow of activities. Such as once activity A (a predecessor) finishes then activity B (the successor) can begin. For example, as depicted at the top of Figure 4, the TFL specifications (the predecessor) must be completed before the programming of the TFLs (the successor) can begin.

For a Start to Start relationship, the activities initiate at the same time, but they do not need to end at the same time. For example, while drafting the SAP, a statistician can also begin drafting the TFL specifications, but the two activities do not need to be completed together.

The third relationship, Finish to Finish, is used when the completion of two activities should be coordinated. For example, validation programs can be written and tested, but they are typically considered final only after the validation checks are performed.

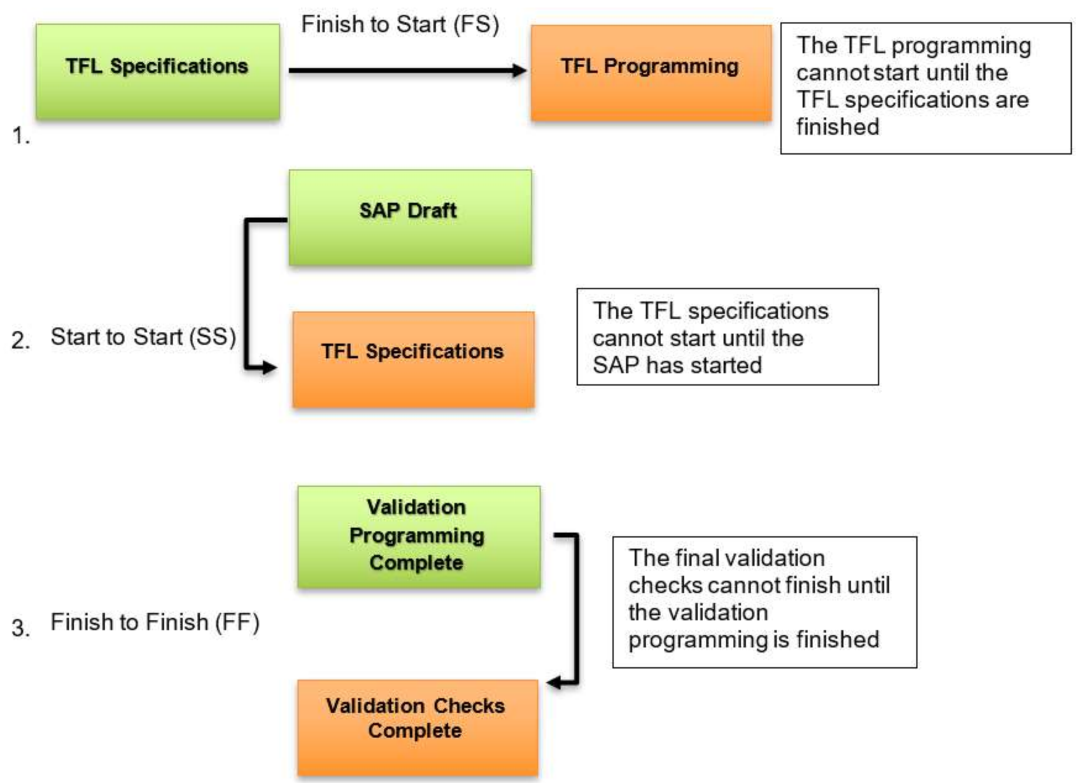

Figure 4: Examples of Logical Relationships

All activities and their logical relationships are then integrated into a network diagram. Figure 5 illustrates an abbreviated version of network diagram for our working example. It contains the probable activities for the final run of TFL programs after all development and testing has been completed. The start (on the far left) begins with the final ADaM datasets and it ends on the right with the completed TFLs. Predecessors are shown at the arrow tail and successors are shown at the arrow head. All relationships below are Finish to Start (FS).

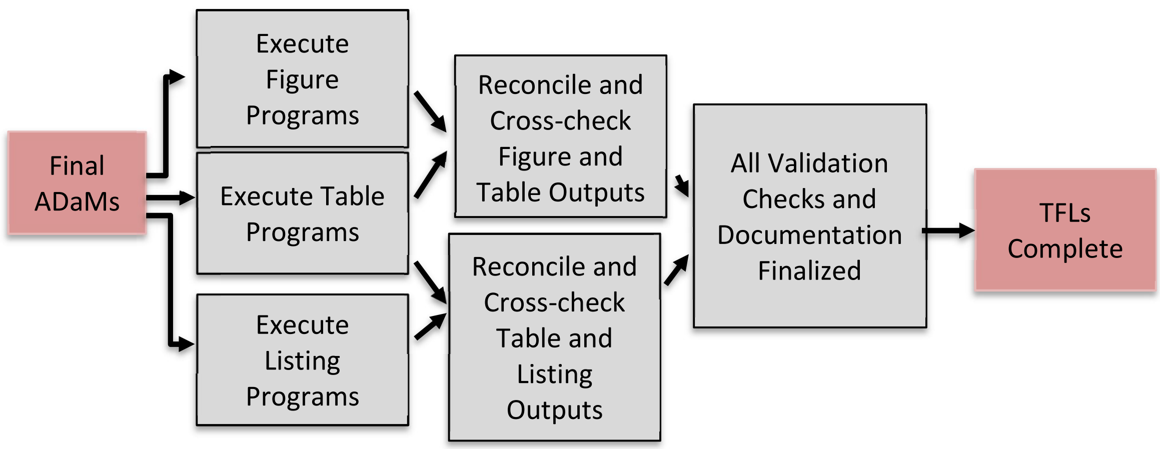

Figure 5: Abbreviated Network Diagram

ESTIMATE THE DURATION OF YOUR ACTIVITIES

The *PMBOK® Guide* presents a few estimation techniques that you may consider for calculating the duration of your activities. They are presented below in order of least comprehensive to most comprehensive (but also most time consuming).

The first is known as *analogous estimating*. The analogous estimating technique involves taking a similar project or activity as reference. It is the least accurate but most expedient way to forecast the duration required. For example, if your current project requires 200 TFLs and you know that your last deliverable required 100 TFLs and took three weeks to complete, you would estimate your current project to take six weeks. Even though analogous estimating is a gross value estimating approach, it does consider differences in complexity, but only at the project level. Therefore, if you know that your current project of double the size also has more complex programming involved, you can take that into consideration and safely extend the time to longer than six weeks.

The next estimation technique is *parametric estimating*. It calculates duration by also considering historical data, but it utilizes a statistical relationship to calculate the durations with more specificity. Typically, this is done by multiplying the work to be performed by the number of labor hours per unit of work. The labor hours are taken from the historical reference. For example, if you have historical data that tells you, on average, it takes five hours to create a unique table program, and your current project requires 40 unique tables, the duration required to program the unique tables for the current project is 200 hours.

The last estimation technique is *three-point estimating*. Three-point estimating considers uncertainty and risk into the single-point estimates to develop a range for an activity's duration. The three "points" are:

1. Most likely (M): This is the mostly likely estimate, given the resources likely to be assigned, their productivity, and realistic expectations of availability for the activity.
2. Optimistic (O): The activity duration based on the best-case scenario for the activity.
3. Pessimistic (P): The activity duration based on the worst-case scenario for the activity.

Estimates for most likely (M), optimistic (O), and pessimistic (P), are based on the information that you have at the time.

Let's assume that you have a good idea of who the programmers are that will be assigned to these 40 tables. You know that they are solid programmers with whom you have no concerns of being able to meet the five-hour benchmark. However, you also know that a new employee will be on-boarded during the table programming activities. Because of this, you have a strong suspicion that those table programmers may need to assist intermittently to help on-board that new employee. In this case, you may feel that a most likely estimate (M) of 6.5 hours would accurately depict realistic expectations.

For your optimistic estimate, the best-case scenario is that you will be able to find other programmers who can spend time to help on-board the new employee. If this occurs then the assigned table programmers can remain fully dedicated to the table programming, so your optimistic estimate (O) is five hours.

For the pessimistic estimate, you need to plan for the worst-case scenario. Suppose that not only do the table programmers all assist with intermittent on-boarding tasks for the new staff member, but that also their task takes longer than expected because the tables are more complex than initially anticipated. If the table programmers are interrupted for longer than expected, and their task is more complex than expected, you make your pessimistic estimate (P) at 9.5 hours.

Now, with your three estimates, M=6.5, O=5.0, P=9.5, you are ready to apply an estimation technique known as the *Beta Distribution* (or sometimes called the PERT formula). The formulas for the expected value and standard deviation are provided in Table 2. If you notice, the Beta Distribution weights the most likely (M) estimate more heavily than the optimistic and pessimistic estimates, thus simulating a bell-shaped distribution.

Table 2: Three Point Estimating Formulas

Beta Distribution Expected Value (E)	E = (O + (4 x M) + P) / 6
Beta Distribution Standard Deviation (σ)	σ = (P – O) / 6

You calculate an expected duration of (5.0 + (4 x 6.5) + 9.5) / 6 = 40.5/6 = 6.75 hours. The standard deviation is (9.5 – 5.0) / 6 = 0.75 hours. You can then estimate the programming for the creation of a unique table for this specific project to take 6.75 +/- 0.75 hours (or 6.0 to 7.5 hours) within one standard deviation. Although this estimating technique took a bit more effort than parametric estimating, it produces a more accurate estimate.

FIND YOUR CRITICAL PATH

The critical path method assists with keeping your project on track. It allows you to identify activities that *must* be completed on time in order to complete the whole project on time. It shows which tasks can be delayed and for how long. It also tells you the earliest and latest you can begin each activity to maintain the schedule.

To begin, apply the duration estimate (calculated above) to each activity in your network diagram. The *critical path* is the sequence of activities with the longest duration. Figure 6 displays a hypothetical example where the duration of each activity is presented inside of its box. The red circles, S and F, represent start and finish, respectively, and have no duration associated with them.

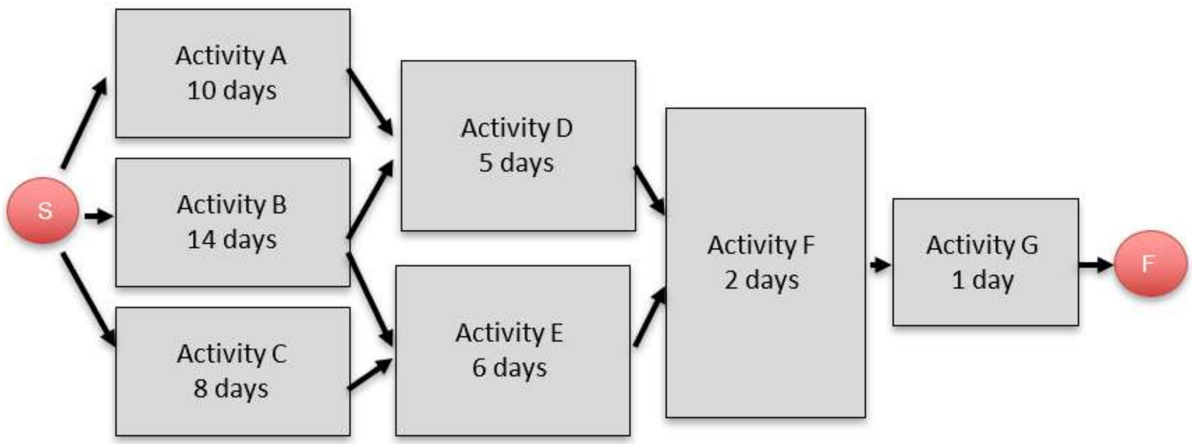

Figure 6: Network Diagram With Durations

In order to identify the critical path, list each unique path from start to finish and sum the durations for that path. There are four paths shown in Figure 6:

Path 1: A, D, F, G = 10 + 5 + 2 + 1 = 18 days

Path 2: B, D, F, G = 14 + 5 + 2 + 1 = 22 days

Path 3: B, E, F, G = 14 + 6 + 2 + 1 = 23 days

Path 4: C, E, F, G = 8 + 6 + 2 + 1 = 17 days

Since the third path is the longest duration, it is the critical path. This tells you that it is important to carefully monitor the activities along this path (B, E, F, G). The float, *or slack*, for the activities along the critical path is zero. This means that if any of these activities are delayed, the entire project will be delayed. The next longest path is Path 2, requiring 22 days for the completion of activities (B, D, F, G). Since activities B, F, G are on the critical path, only activity D is unique and thus has a float of one day. This means that if activity D is delayed one day it will not cause the project to be delayed, but anything more than one day will affect the project.

As the project in Figure 6 begins with activities A, B, and C all commencing at the same time, if it becomes clear that activity C needs to be delayed for three days, will this cause a problem? Since activity C is not on the critical path, an additional three days will put it ending at day 11 and increase the duration of Path 4 to 20 days. This will not delay the project if the other activities on the critical path (E, F, G) stay on track. Calculation of float should be performed for all activities as an easy way to gauge potential impact to the overall deadline. This can be tedious, especially for a large project. This where the use of project management specific software packages can help.

Note that Figure 6 is drawn using the Precedence Diagram Method:

- There are 7 activities (A, B, C, D, E, F, G)
- There are 2 milestones, Start and Finish (red circles)
- Durations are included
- Predecessors are shown at the arrow tail and successors are shown at the arrow head
- Activity A has one predecessor (start) and one successor (D)
- Activity B has one predecessor (start) and two successors (D and E)
- Activity C has one predecessor (start) and one successor (E)

- Activity D has two predecessors (A and B) and one successor (F)
- Activity E has two predecessors (B and C) and one successor (F)
- Activity F has two predecessors (D and E) and one successor (G)
- Activity G has one predecessor (F) and one successor (finish)

To close out this section, note that there are two techniques that you can use if your schedule starts to slip. If you find yourself needing to accelerate your schedule, but you cannot comprise the scope, consider *fast tracking* or *crashing* as depicted below in Figure 7.

Fast tracking is where activities that would have been performed sequentially in the original schedule are performed with some degree of overlap. For example, your original plan may call for the SDTM domains to be completed before the ADaM dataset programming begins. However, if your schedule slips, you could feasibly start the ADaM programming a few days earlier, before the SDTMs are fully completed. Fast tracking only works when the activities are on the critical path. The forced overlap of ideally sequential tasks typically leads to increased coordination and communication efforts on your part. It can also lead to an increased risk in quality.

Crashing is a technique that adds additional resources to the critical path tasks. This option results in an increased resource cost. It also assumes you have trained resources ready and available, and it is only feasible to a certain limit. The law of diminishing returns will apply at some point, and the productivity of each resource will become marginal.

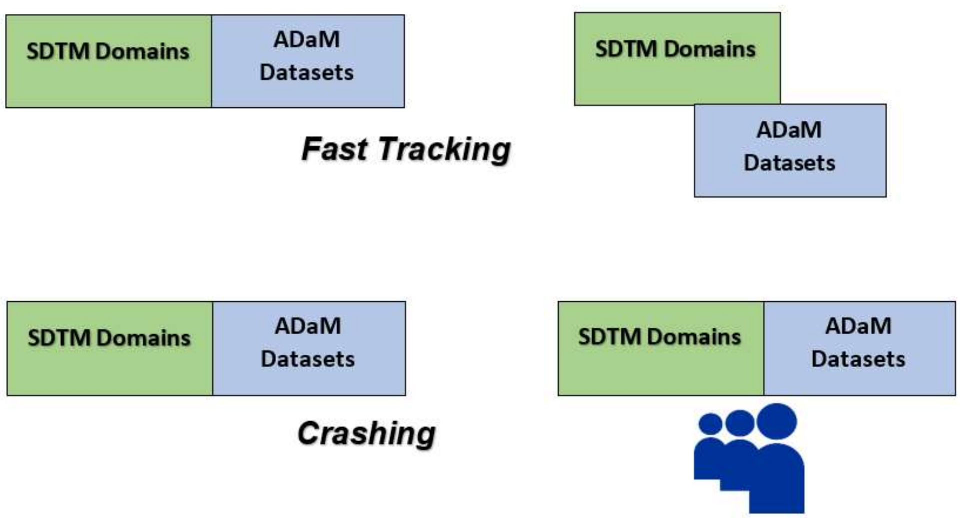

Figure 7: Fast Tracking and Crashing

STAKEHOLDERS

Often in the clinical trials environment, there can be the natural tendency to perform insularly and only focus on your immediate team and customers. However, this can be detrimental when working on a larger deliverable that may have input, feedback, and constraints set by various teams.

Stakeholders are the people (or groups) that *could impact* or *be impacted by* the project. This impact can be either positive or negative. Identifying stakeholders early helps increase your chances of success by ensuring that you engage them in an appropriate way. It also helps capture the value that each can bring to your project. Some stakeholders may support you, others may hold information that you need, and yet others may have the power to derail your project simply due to their level of influence.

There are many approaches that you can consider while organizing your stakeholders. The *PMBOK® Guide* offers different ways of classification for consideration. One method is to identify the degree of influence and level of interest of each stakeholder. The intersection of these two criteria is then plotted on a stakeholder matrix, shown in Figure 8: Influence and Interest Stakeholder Matrix.

Mapping where each stakeholder lands in the matrix will help you develop an approach to optimally engage each stakeholder in an appropriate fashion. You do not want to risk neglecting those who have a high degree of influence and a high degree of interest in your project. Likewise, it will be a wasted effort on your part to over-engage with those who have little influence and interest in your project.

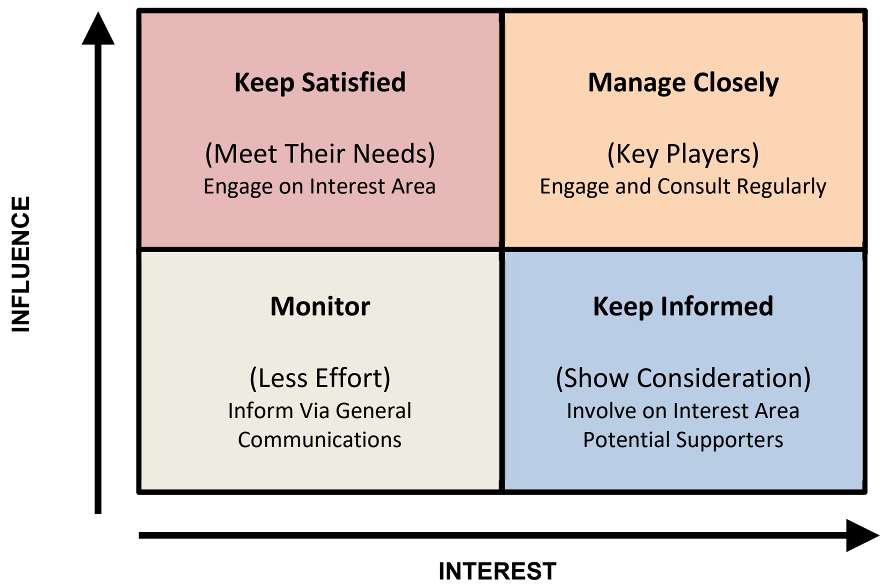

Figure 8: Influence and Interest Stakeholder Matrix

There are variations of the Influence/Interest stakeholder matrix that can be created for different dimensions as well. Mapping options include those for Power/Interest, Attitude/Knowledge, or Power/Support. For a power and support matrix, stakeholders who fall on the top half of the matrix help to identify those who are crucial to the project's success. If you have them available, templates and assessment matrices are great tools to use (or to consider developing) if you find utility in their functionality, especially when working on larger deliverable packages.

Stakeholder influence can also be classified by directionality. The perspective of the directionality is typically presented from that of a project manager and not a team specialist. Be aware and watch out for that perception if you read further about this.

Directionality is classified in four ways as:

- Upward: Senior management, sponsor, etc.
- Downward: Team members – managing *your* team.
- Sideward: Peers. This one is thought to include those with whom you may have competition for resources. Also includes others with whom you collaborate.
- Outward: Anyone outside of the project team. Includes regulatory agencies, the public, etc.

Thinking back to our working example, at a minimum, stakeholders likely include data management, clinical operations, medical writing, quality assurance, clinical science, project management, senior management, your team of programmers and statisticians, and regulatory agencies. Each of these stakeholders either directly impact your project or are impacted by your project. Personally, I find that a straightforward but fruitful practice is to simply create a list of all known stakeholders, and then add abbreviated notes on your engagement and communication level with them. A condensed example is provided in Table 3 for illustration only.

The level of detail is dictated only by your own comfort level and what is warranted for your project. Use your judgement and modify accordingly with what works best for your organization. The main importance is that you adequately identify all stakeholders and that thought is given with how they can impact your project. The appropriate level of engagement will flow naturally from these activities.

Stakeholder identification and monitoring has interdependencies with many other Knowledge Areas, but principally with communication and risk. It is not only conceivable, but highly likely that there will be a high degree of overlap amongst these Knowledge Areas. In this sense, it is efficient and completely appropriate to incorporate risk, communication, or other dimensions into your stakeholder list or engagement plan.

Table 3: Example of a Stakeholder List

Stakeholder Functional Area / Main Contact (s)	Communication	Interest/Involvement	Direction of Influence
Data Management (DM) / <Lead Data Manager Name>	• DM hosts weekly meetings at beginning then daily meetings as database lock approaches. Participation is expected. • Enrollment trackers are sent by this team (push communication). • This stakeholder requires from you that any data findings or concerns are documented in email.	High/High during preparing for database lock.	Sideward *(you will collaborate with this group)*
Programming and Statistics <Your Team>	• You host daily check-in meetings (interactive) • Email notification of timelines, monitoring of progress is constant and high. • Communication needs to be managed well.	High/High - Maintains a high level throughout your entire project.	Downward
Project Management / <Lead Project Manager Name>	• This stakeholder expects your input at weekly meetings. • Milestone status is tracked via a dashboard (pull communication). • You need to alert them if any scheduling delays arise.	High/Low to Medium. Involvement is primarily oversight of the high-level timeline.	Upward

COMMUNICATION

Effective communication, including the proper timing, the appropriate audience, and a suitable level of formality, is essential for a successful project. You want to consider not only the communication amongst your team but also that between yourself and your stakeholders. According to the *PMBOK® Guide*, Project Communications Management consists of two parts. The first part is developing a strategy to ensure the communications are effective. The second part is carrying out the activities necessary to implement the communication strategy.

There are [n (n-1) / 2] communication channels within a project, where n represents the number of stakeholders. As an example using a team size of five for your team, if you have only one project manager and one data manager, for just this small subset of seven stakeholders you will need to manage (7*6) / 2 = 21 communication channels! A bit of awareness and organization will set you up for success.

When considering the communication with your stakeholders, your focus will be on what each team needs to know, when they would need to know it, and how best to get the information to them. Overloading stakeholders who have low interest and low influence in your project may cause them to become completely disengaged. On the flip side, communicating insufficient or unclear information to a stakeholder who has an upward direction of influence can lead to false assumptions and may endanger the entire project.

When thinking about your immediate team of programmers and statisticians, you will likely utilize *informal* communication exchanges. Informal exchanges are unofficial, spontaneous, and flexible. They include email exchange, ad-hoc discussions, and instant messaging. You will also want to consider frequent, but brief, check-in meetings. These are effective tools that allow you to be informed of progress in real-time. They also facilitate team discussions to identify possible issues early. If your team is virtual, you need to ensure that the team maintains effective communication throughout the duration of the project. Not having a team that is co-located can be subjected to an increased likelihood of miscommunication, a tendency of under-communication, and team disengagement. Virtual teams are efficient and have many advantages and merits, but (lack of) communication can be a challenge. If it is not managed and monitored carefully, it can result in a negative impact to your project.

When planning the proper communications with upward (and sometimes with sideward or outward) stakeholders, a more official approach is likely necessary. These forms of communication include presentations, reports, and formal meetings with an agenda, and subsequent meeting minutes to document discussion points, decisions, and outstanding action items. If most of the formal meetings are orchestrated through a project manager or senior leadership, you should still have a plan developed for when you may need to communicate with upward influencing stakeholders. For example, you may encounter sensitive data issues, or you may need to document and communicate the unblinding of a trial. Each of these situations require a formal communication plan and exact implementation.

The second column in Table 3: Example of a Stakeholder List, mentions *push*, *pull* and *interactive* communications. Push communications are one-way communications sent from the sender to a receiver. Examples of push communications are memos, emails, voice mails, or press releases. They are preferable when you do not expect a response back from the party to which the communication was passed. For example, if you need to notify one of your statisticians that a draft figure is ready for review, a push communication via email would be an optimal choice.

Pull communications are used when the receiver needs information on demand. Information is typically stored in a shared repository such as a website, dashboard, or SharePoint site. For example, your project management group may keep a milestone timeline on a SharePoint site so that any involved party can access it when needed.

Lastly, when you require more interaction and dialog, interactive communications such as video-conferencing or meetings are the preferred methods. They allow you to observe non-verbal communication, deal with time-sensitive decision making in a more efficient manner, and they enable the clarification and confirmation of ideas in real time.

COST

Project Cost Management is primarily concerned with the cost of resources needed to complete the project activities. Many of the same tools and techniques illustrated above in Project Schedule Management can be applied to this area for the planning, estimation, and control of costs to your project. For example, analogous, parametric, and three-point estimating techniques all apply to cost forecasting just as they did for duration estimations.

QUALITY

Failure to understand and follow quality requirements can have serious negative consequences, such as decreased profit and increased levels of employee attrition, errors, and rework. Poor or low quality can also have a negative impact on your team's reputation.

One important project management concept to understand is the *cost of quality*. It refers to the cumulative costs required to bring poor quality products to standard. This includes the prevention of non-conformance, assessing deliverables for performance, and the costs of rework. Cost of quality can be separated between cost of conformance and cost of non-conformance as illustrated in Table 4: Cost of Quality.

Table 4: Cost of Quality

Cost of Quality	
Cost of Conformance	**Cost of Non-Conformance**
Prevention Costs *(costs of preventing defects from arising in the first place)* Training staff, documentation, following standards, quality assurance activities	Internal Failure Costs *(cost when defects in deliverables are detected internally)* Rework or defect repair
Appraisal Costs *(cost of activities for inspection)* Testing and inspection, quality control activities	External Failure Costs *(cost when defects in deliverables are detected by customer)* Recalls, loss of reputation

Likely you already have validation procedures and quality checks in place (e.g., independent programming, review and sign-off of the SAP by different functional teams, training on SOPs, etc.). These are all examples of costs of conformance. Cost of conformance is more inexpensive than cost of non-conformance. You may have heard the concept, "Prevention over Inspection". It originates out of this idea that resource efforts are better appropriated and most cost effective for activities that fall on the top left of Table 4.

Failures or non-conformance are further categorized as either internal (those found by the project team) or external (those found by the customer or regulatory agency). Consult with individuals or groups in your organization who have expertise or specialized knowledge in quality assurance, quality control, and quality systems if you need more guidance on this topic.

From an applied view, preparing for quality should be performed alongside the other planning processes that you undertake. Be cognizant of ensuring that you have prevention methods in place for your activities. Also ensure that you and everyone on your team fully understands and complies with your quality related activities.

RESOURCES

Project Resource Management includes the identification, acquisition, and management of resources needed for the project. Resources include both personnel and physical resources (e.g., equipment and facilities) In our working example, there is likely not enough time to identify, interview, on-board, and train new personnel.

Note that the *PMBOK® Guide* details not only resource planning (where resources include team members, supplies, materials, equipment, etc.), but it also includes techniques to aid in conflict management, tips to work in a virtual environment, skills to help with decision making, and tools to help with the development and management of your team.

RISKS

Risks are quite simply "the unknown". The uncertainty can be neutral, with no effect on the project. It can develop into a positive risk, known as an *opportunity*. Or, it can develop into a negative risk, also known as a *threat*. The amount of time spent on identifying risks, quantifying them, and planning risk mitigation should be directly correlated with the importance of a project. Be aware that risks will continue to emerge even after the planning stages of a project are complete. Therefore, the planning, identification, assessment, and monitoring of risks should be an iterative process that is conducted over the lifetime of the project.

Event risks are possible future events that may or may not occur. As discussed above, they can be classified as either positive (an opportunity) or negative (a threat or unfavorable outcome). In order to identify potential risks to your project, brainstorm what may occur based upon your experience from other projects. Additionally, you can consider reaching out to your project manager, your team members, or a subset of your stakeholders who may be subject matter experts to help you determine all potential risks.

An example list is provided in Table 5. Note that Risk Number 2 is classified as an opportunity or a threat. It is possible that some risks could be either, depending the situation. For example, if part of your deliverable package becomes outsourced, this may present an opportunity to your team, as it allows resources to be freed up that could help with the other tasks. However, if your team needs to provide a high degree of oversight to the vendor, and if the timetable for the deliverables moves up because of a need to provide the vendor with time to complete their task, this risk becomes a threat to the project.

Table 5: Potential Event Risks

Risk Number	Potential Risk	Risk Classification
1	Other team members (resources) become available to you	Opportunity
2	Management informs you that they prefer to use a partially outsourced model for cost purposes. Now the Define-XML creation will be performed by an external vendor (CRO)	Opportunity or Threat
3	Someone from your team resigns	Threat
4	A stakeholder resigns	Threat
5	Your team identifies major issues upon running the Pinnacle 21 report	Threat

If you have the time and ability, consider performing a risk analysis, also known as a *risk assessment*. You do not need to be a risk expert or have any special training to accomplish this task. In fact, a *qualitative risk assessment* is a very simple method that does not rely on any complicated tools or specialized software. A qualitative risk assessment examines the likelihood and impact of each risk. It helps you prioritize your risks to determine upon which risks to focus.

Table 6: Example of a Likelihood Scale Definition

Rating	LIKELIHOOD
Low	Highly unlikely to occur. Would only occur in an exceptional situation.
Moderate	Possible
High	Likely, has occurred in the past for similar projects.

To begin, you first need to define the rating scales for likelihood and impact. With a qualitative risk assessment, a relative or ordinal scale is used. Presented in Tables 6 and 7 are three-level rating structures, but you can extend to five or more levels if you have the need for more granularity (e.g., Very Low, Low, Moderate, High, Very High). These ratings are dependent on the specific details of your project, so they should be customized to fit your needs. For example, on one project, an impact which causes a 10-day delay may be considered moderate, but for another project, it would be considered as high or critical impact.

Table 7: Example of an Impact Scale Defintion

Rating	IMPACT	
	Cost	Schedule
Low	No increase to budget	No change to schedule
Moderate	< XX % increase to budget	< XX days delay
High	> YY % increase to budget	> YY days delay

Once you have the likelihood and impact scales defined, you simply create a risk assessment matrix as in Figure 9. The risk assessment matrix is a presentation tool that provides you an easy way to assess risk prioritization. The early identification of risks, even in the context of your assumptions and constraints as a programmer or statistician, can better position your larger team for an efficient risk mitigation strategy should conflicts or changes arise during the project.

LIKELIHOOD				
	High	**Moderate Risk**	**High Risk**	**Very High Risk**
	Moderate	**Acceptable Risk**	**Moderate Risk**	**High Risk**
	Low	**Acceptable Risk**	**Acceptable Risk**	**Moderate Risk**
		Low	Moderate	High

IMPACT

Figure 9: Risk Assessment Matrix

PROCUREMENT

This Knowledge Area is concerned with the processes surrounding the purchase or acquisition of products or services from outside of the project team. Typically, these include contracts or legally binding documents such as purchase orders or service level agreements. It is not applicable to our example and is likely a rare occurrence in your day-to-day activities. If you find you need guidance in this area, please reach out to your legal and business departments before navigating procurement processes on your own.

INTEGRATION

Project Integration Management is the combination and coordination of all the various project management activities. It unifies, consolidates, and interweaves the nine other areas together. In contrast to the other Knowledge Areas, which may be individually managed by specialists (e.g., finance specialist, risk management expert, quality team), it is within this Knowledge Area where the main efforts of a project manager reside. The project manager is the one who combines the results of all other Knowledge Areas, monitors the overall project work, and oversees the complete view of the project. It is also within this Knowledge Area that the charter is developed, the baselines (i.e., scope, schedule, cost, and PMB) are maintained, and change requests are issued.

For our working example, where you are essentially your own project manager for what is in your targeted scope, the takeaway is that the interdependencies of the various Knowledge Areas must be managed. Not surprisingly, changes to one area impact another and a balance must be maintained. That balance can be struck by a myriad of possibilities such as examining alternatives, reallocating resources, or changing scope, etc.

CONCLUSION

Project management is a vast topic. This paper is a summary to bring an awareness of the existence of the *PMBOK® Guide* to an audience who may not be familiar with it. A survey of the concepts of project management, complied as ten Knowledge Areas of the *PMBOK® Guide,* are presented above, including tools and ideas to assist programmers and statisticians in their day-to-day activities.

REFERENCES

A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Sixth Edition (2017). Project Management Institute, Inc.

https://www.wrike.com/project-management-guide/faq/what-is-a-baseline-in-project-management/
https://www.cl3ver.com/blog/the-stakeholder-theory-and-matrix-for-the-aec-industry/
https://www.project-management-skills.com/qualitative-risk-analysis.html
https://study.com/academy/lesson/qualitative-risk-analysis-benefits-limitations.html
https://www.project-management-skills.com/critical-path-method.html

RECOMMENDED READING

PMBOK® Guide – Sixth Edition (2017). This is the flagship publication from PMI and the fundamental resource for effective project management in any industry. The sixth edition is the current release that includes expanded content on agile practices.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jennifer Sniadecki, MS, CAPM
Ultragenyx
lambertjen@gmail.com

Any brand and product names are trademarks of their respective companies.