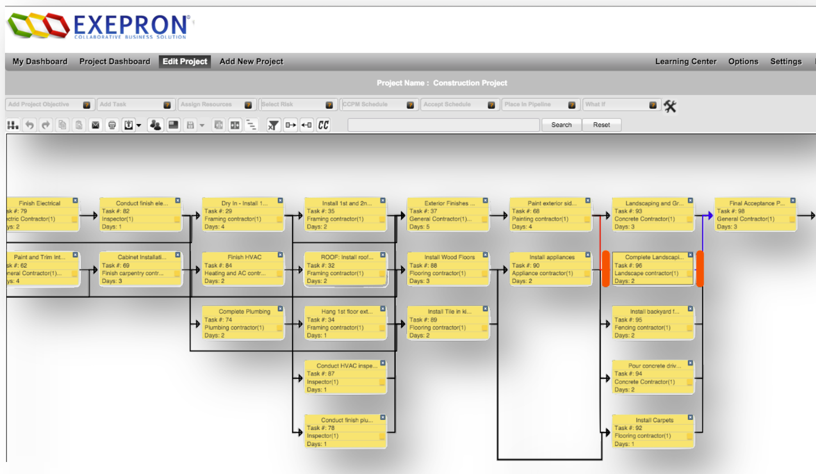
Building Executable Project Plans



Introduction to Project Planning

There seems to be two extremes to project planning:

- "Seat of the pants" methodology: little to no documentation, possibly a list of things "to do"
- Hundreds to thousands of tasks, at a level of detail that would make a detailed work instruction look like a high-level document (with times down to minutes or fractions of minutes).

What's a project?

According to the Project Management Institute (PMI):

"It's a temporary endeavor undertaken to create a unique product, service or result.

*A project is **temporary** in that it has a defined beginning and end in time, and therefore defined scope and resources. And a project is **unique** in that it is not a routine operation, but a specific set of operations designed to accomplish a singular goal. So a project team often includes people who don't usually work together – sometimes from different organizations and across multiple geographies.*

The development of software for an improved business process, the construction of a building or bridge, the relief effort after a natural disaster, the expansion of sales into a new geographic market — all are projects.

And all must be expertly managed to deliver the on-time, on-budget results, learning and integration that organizations need.
(https://www.pmi.org/about/learn-about-pmi/what-is-project-management)

Why do we need to plan a project?

PMI includes in their definition "*a specific set of operations designed to accomplish a singular goal*". Implied in the PMI definition is that a project takes more than one person to accomplish (note the "project team").

During project planning, that set of operations must be defined and properly sequenced before team members and resources to accomplish each set of operations and time estimates for each set of operations can be defined.

Without a good project plan properly communicated and managed during project execution, no one knows what to do when (or worse, decides on their own what to do, when, and to what extent!), which adds significant risk to meeting project scope requirements, staying at or below the project budget, and finishing at or before the (original) agreed-upon project due-date.

What's a project network?

In our book, ***Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*** (Gerald I. Kendall and Kathleen M Austin, J.Ross Publishing, 2012), we defined a project network as a model of the major work needed to meet the stakeholder needs and drive some part of the organization's goals. Regardless of whether people are using PERT networks, lists, GANTT charts or some other format, the network is the raw material of projects. In some form, it describes tasks, the correct sequence of the tasks (what must come before something else), the effort expected to complete the work, the resources, and other notes about the work that may not be understood from a simple diagram).

Some people distinguish between a network and a project plan. It is true that when you are using a particular software package or methodology, this terminology can represent two physically different things. For our generic purposes, it is sufficient to use the definition as stated above.

The project network impacts everything – the time and skill demands on the resource, the understanding of the work needed, the ability to monitor execution against a valid plan, the insulation against variability, the ability to meet the organization's and sponsor's goals. Therefore, like resources, project networks are at the very heart of the multi-project management system.

Good project networks are like good roadmaps. They allow you to chart a high-speed course. They provide warning signs for curves and bumps before you reach them. They clearly lay out the boundaries over which you are traveling. AND they enable the project manager to focus their energy on dealing with poor road conditions, detours, construction, instead of charting the course as they go. They will have had a chance to pre-plan alternative courses should a roadblock be encountered. In this way, they help you get to your destination safely yet at high speed.

What causes poor project networks?

The two problems that we've seen universally in building project networks are:

1. No formal process is used. Just like doing pre-flight checks, network building requires a series of precise, well-executed steps. Miss one step and the project will crash.
2. Scrutiny of the network is missing. Pilots and co-pilots cross-check each other. No one person has all the information necessary to understand all of the work of a project. Without scrutiny, a project network is a fantasy on paper.

Hmmm…sounds like we need a process – what needs to be included?

It is much easier to define the content of a task if you know who is going to use the output of the task and what they need before they can start their task. In other words, it is the input needed by the next task that defines the predecessor task's work content. Therefore, it makes sense to construct a project network working from the end to the beginning.

However, once a network is constructed, what is the best way to scrutinize that series of tasks and milestones? For most people, the answer is in the sequence in which the work will be performed. Therefore, build the network from the end to the beginning, but check a network working from the beginning to the end.

One analogy is having an architect design a house. They start with the end in mind. How big will the house be, how many bedrooms, how many floors? Then, for each floor, how many rooms, which rooms must have windows, what view must the window have. Then for the windows, how big should they be, what insulation factor do you want, what type of material is needed? For the builder doing the construction, they work from step 1 forward. They need to know that they must first clear the lot, then put in the footings, then lay the foundation, etc.

Task estimates need to be done by people with expertise in the given collection of work. With an architect's design and with the help of a general contractor, an electrician can estimate how much work is involved and how long it will take to wire the house. The HVAC expert can estimate the number of units required and the work and cost to lay the duct work to heat and air condition the house. Just as I would expect a general contractor to rigorously check the estimates for building a house by validating with skilled tradespeople, we need a project manager in collaboration with subject matter experts to validate each collection of project work, by skill, for both correct content and estimated time and cost.

Another key issue in planning is determining how much and what types of uncertainty to account for. Since any individual estimate can be wrong, the people building the network must understand the amount of variation possible in the work. The problem is, that at the time the network is being built, there are a lot of unknowns. However, when you remove the uncertainty about resource availability, caused by confusing priority systems and bad multitasking, you are able to focus much more on the task content. When you further remove the uncertainty of how long it will take to get help and decisions from management and other groups in the organization, it is easier to judge the unknowns.

It's also crucial to understand *where* the biggest variability is likely to occur – more towards the beginning or end of a project. If the biggest variability is towards the end of the project, we must have a healthy buffer of time left at that point in the project in order to account for and absorb that potential variability.

As an example, a web designer tells me that it will likely take five days to complete the design work I've asked for. After discussion, we agree that it is possible to do all the work in 3 days. If I check with the designer at the end of the first day, and he/she says "10 more days to go", I have the right and obligation to ask what happened. And if I am a skilled task manager, I can also offer to help in some way that will bring the time closer to the likely time. Has the content changed? Is the designer stuck waiting for a user decision? The original estimates framed the work in a way that we can now manage it effectively during execution.

Using the same example, if we are undertaking a project where the majority of the variability in changing a web site is at the front end, then the fact that we have used up the majority of the buffer half way through the project is not alarming. It is expected.

Why are current project planning efforts usually a waste of time?

Many project plans are not used once a project begins to execute. This means that either:

- they were created only to satisfy some policy of the organization or
- they are obsolete upon starting execution or
- they were not structured in a way to facilitate execution (e.g., dependencies incorrectly mapped, not organized in a fashion that aligns with how the work will be accomplished, etc.).

If our experience is common, that over 75% of the value of a project plan only comes during execution, then no matter which above case is true, the effort to plan was almost a total waste of time.

To avoid this waste, the first thing to do is to not make the common mistakes. There are four such mistakes in building project plans and networks, which make them difficult to use during execution:

1. Assigning named resources (i.e., specific people) to tasks at planning time. People leave companies. People get sick. People get tied up on other projects longer than expected. People get assigned to other projects. This approach simply does not work. Note that this does not absolve a manager from assigning a named resource or thinking about who can be assigned to a task *as it comes ready to execute*. The issue is about timing and predictability. Also, if there is only one resource in the organization who has a certain, unique skill set, then the skill set and the resource are one and the same. This is a huge red flag to an organization about their vulnerability to project delays.
2. Using the wrong level of detail to construct the plan. A project manager must focus on the important few tasks that really govern the project outcome. It is a huge mistake for a project manager to try to manage many hundreds or thousands of tasks. Breaking work down, during a planning stage, to its lowest level of detail or

work breakdown structure is unnecessary and error-prone. At the same time, a single task that is estimated to take more than two weeks probably needs to be broken down further.

3. Not rigorously checking every task and the collection of tasks against the key stakeholder's needs. Some tasks may not be needed because they add no value to the key stakeholder. Others that are essential to meet the needs are missing. In many cases, the stakeholder needs are simply not well understood until it is too late. The key stakeholder should have a project goal in mind which is tied to the company goals. Often, other stakeholders see the project as a way to get other needs met, and it becomes like members of congress attaching pork to a bill intended for something totally different.
4. Not rigorously checking for additional and/or missing dependencies. A missed dependency can mean the entire schedule is wrong. For example, we have seen a product launch delayed for weeks because the Legal department due diligence was not included in the plan. Subject matter experts can catch most of these mistakes before they happen. But they often are not included in the advance scrutiny of project networks.

Why use project networks at all?

Within any organization, there are at most a handful of people that love to build project networks. For most people, building project networks seems to be too complex and gives them a headache, literally. Therefore, it is tempting to not build project networks, or to put in a token effort to be able to show a plan, even though it is not useful. Another common practice is to let the project network expert build the plan almost independently, in a way that only this person understands. For this reason, it is also common practice that networks are not rigorously checked with key stakeholders.

We have personally witnessed organizations that consistently get more than 95% of their projects completed on time, on budget and within scope. Every one of them will tell you they could not have done it without the use of networks during both planning and execution. Therefore, you must find those people in the organization who love building networks – who think of it as a hobby, e.g., solving crossword puzzles, that is a lot of fun. They must be coached to construct a plan with terminology that is easily understood by other humans who are not into the lingo of project networks. They must also be coached to communicate their assumptions behind the network and its dependencies in simple language.

In our opinion, about 25% of the value of a project network is from planning. It is almost useless for predicting short term resource loading, because projects never execute as planned. Some tasks are executed more quickly. Others take much longer. However, the plan provides an up-front prediction of the approximate workload. It is a sanity check that the work required to deliver the project results makes sense when compared to the benefits the project will deliver to the organization.

But there is another key value to the plan itself. The project plan allows the organization to know when the project can be released and approximately when the benefits will start to accrue.

We believe that 75% of the value of the plan comes during execution. During execution, it is the yardstick against which progress is measured. If a plan contains strategic buffers (blocks of time placed at the end of a project equivalent to at least a third of the total time of the project), these buffers provide the real truth about how effectively the project is being executed.

When you compare how fast a buffer is being eaten away, relative to how fast the most critical tasks in the project are being completed, you have a compelling and proven story about the real time status of execution. However, most organizations wait too long to act on this story. *The network buffer story is only valuable if it is analyzed and acted on daily.*

Therefore, we claim that it is impossible for any organization to have their projects under control (meeting their goals better than 95% of the time) without properly constructed project networks used frequently during project execution.

What's a good project planning process to follow?

To us, a process is a series of steps that can be repeated by different people and will generate essentially the same results. This is the intent of the process that we describe, but do not detail, below. The details and associated examples and diagrams are included in an upcoming blog post. <u>These steps must be followed in sequence</u>. None of the steps can be omitted. The steps embed five different ways to avoid risks in the project plan – risks of missing steps, risks of missing dependencies, risks of including unnecessary scope, risks of missing key stakeholder needs, and overall risks of the project. The 10 steps we recommend are:

1. Define the project's measurable goals, tangible scope and sponsor criteria.
2. Define the tasks required for the backbone of the project network (one main path), starting at the end of the project and working towards the beginning.
3. Add the tasks required to build the skeleton (other paths), working backwards from the end, completing all other paths.
4. Read the network forward, from the beginning, rigorously looking for additional dependencies (First risk avoidance).
5. Check every task against project goals, scope and sponsor criteria (Second risk avoidance).
6. Determine resources (skill level and maximum number) that could be assigned to perform the task.
7. Scrutinize the network logic using subject matter and/or skill set experts (Third risk avoidance).
8. Define time estimates, with range of variability (Fourth risk avoidance).
9. Seek ways to reduce overall project duration without compromise.
10. Complete a final overall project assessment (Fifth risk avoidance).

When the organization has such a rigorous process, and every project network is built using the process, two excellent results are achieved:

- Templates that describe a type of project (e.g., new product development, I.T. service implementation or upgrade) can be developed and re-used, saving a lot of time in understanding the tasks involved in constructing a new project plan in the future.
- Projects become more predictable, with the consistency of rigor and validity across all projects.

This 10-step process sounds like it takes a LONG time

Personally, we have spent up to two days working on a network. There was no pre-established template to use, and the company had not done formal project plans before. Much of the time was spent discussing scope and not adding tasks to the network. We have heard of cases where it took a week or more to finalize a project template, for a very complex project. However, the next time a similar project comes up, it typically would require between an hour and a day to customize an existing template.

Remember, even in the most complex of projects, where there may be thousands upon thousands of actual tasks performed, you are looking for ONLY the 200 tasks that the project manager must focus on. For example, in building a large ship, there are hundreds of rooms that must be finished, involving different skilled contractors – painters, plumbers, electricians, dry-wallers, etc. The project manager does not need to know when Joe, the painter, will be painting room 127. In the project plan, he needs to know that the first 30 rooms are scheduled to be painted over a period of 1 week. Before the painters arrive, he must make sure the plumbing and electrical work is complete, and the dry-wall has had time to dry.

Do not be disheartened if you already have a project plan, and you rebuild it using this process and find it requires a longer duration. All that this means is that the original plan was destined to not work, because it was missing pieces.

Conclusions

Project networks are the building blocks to complete a project successfully. If you don't have enough building blocks and you have to get more during execution, the project likely fails to meet its time and cost projection. If the blocks are made of poor material (poor understanding of the work), they will crumble during execution. If we don't understand which blocks go where, and which blocks are needed before we can put the next ones in place, we'll have a lot of rework. From experience, it is worth the effort to have and use a formal process for both constructing and for scrutinizing networks.

We have not heard of a single case of any organization getting predictable results from projects without having a rigorous project plan, constructed using a disciplined and consistent process. This implies that the process cannot be left up to each individual project manager to determine from their own experiences. This chapter overviewed a 10-step network building process that included five risk avoidance techniques. Details for each of these steps are found in Part III of this text. To end up with a good end product, you must not only follow the process, but also have the right people involved in the network building process, as described within this chapter.

Upcoming Topics

- Ensuring the correct level of detail in the project network

- Details for each of the 10 steps to building a good, executable project network

Building Executable Project Plans – Part 2

Part 1 provided an introduction to projects and the need to plan a project.

Summary: Project networks are the building blocks to complete a project successfully. If you don't have enough building blocks and you have to get more during execution, the project likely fails to meet its time and cost projection. If the blocks are made of poor material (poor understanding of the work), they will crumble during execution. If we don't understand which blocks go where, and which blocks are needed before we can put the next ones in place, we'll have a lot of rework. From experience, it is worth the effort to have and use a formal process for both constructing and for scrutinizing networks.

We (Kendall & Austin*, Advanced Multi-Project Management*, J.Ross Publishing, 2012) have not heard of a single case of any organization getting predictable results from projects without having a rigorous project plan, constructed using a disciplined and consistent process. This implies that the process cannot be left up to each individual project manager to determine from their own experiences. To end up with a good end product, you must not only follow the process, but also have the right people involved in the network building process.

Part 2: How to Ensure the Correct Level of Detail in a Project Network

Hint: It's not the lowest level of the Work Breakdown Structure!

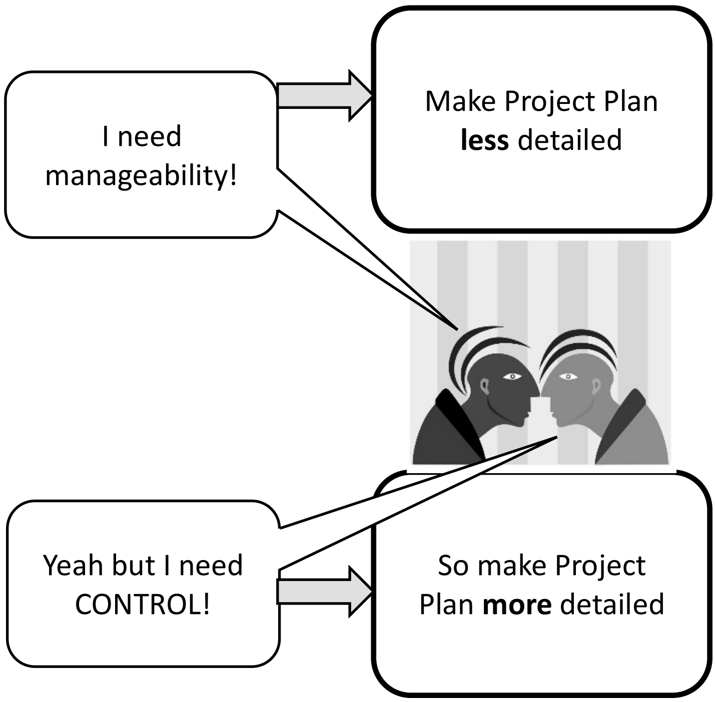
> Oh, I remember the days as a brand-new 2nd Lt working program control and project management on an Air Force weapons acquisition system. I was so sure I was right, following the requirement to insist defense contractors plan the project work to seven levels and report monthly (in the earned value system) at level three. And then, not being able to answer my boss's questions about how the projects were really doing in terms of schedule and cost variance. What did "green" mean in terms of project due date and budget? What was "yellow" and what was "red"? We had a great reporting system, metric-rich and full of detail – but it didn't help us at all to manage the work, nor could we really tell, based on the reports, where the projects were in terms of completion. Truly it was like driving a car forward, on a busy freeway (Atlanta or LA) with only a rear-view mirror. YIKES!!

What a dilemma planning a project can be! Expectations for project plans usually include many of the following:

- Usable for costing the project,
- Structure to track while executing the project,
- Doing resource loading,
- Calculating earned value,
- Detailing exactly what every task should be and/or should include,
- Providing information about inputs for the task,
- Defining each task's exit criteria,

- Describing any specific notes or details about each and every task,
- Enabling managers to know who are the primary performing resources (people, equipment, facilities) and who are supporting resources (those not used for the entire time of the task, but required to achieve the tasks' exit criteria, etc.)

Don't forget, the project plan must also include all the work required to meet the stakeholders' consensus of project scope. Oh yes, a project plan should also be easy to manage!



The Unavoidable Conflict in Project Networks
(from *Advanced Multi-Project Management, Chapter 12)*

It is easy to see the conflict for project planners:

On one hand, in order to assure project success, the project plan must be manageable, which means there's pressure to not have a very detailed project plan (because otherwise it would not be manageable – the focus would be diluted on too many tasks).

On the other hand, in order to assure project success, the project plan must provide all the data needed during project execution (to manage resources, work, costs, timelines, estimates to complete, etc.), which means there's pressure to have a very detailed project plan. It sure sounds like a project plan is being used for more than planning, scheduling, executing, and managing a project – it's also required to be the entire project database! Is that reasonable?

It's not unusual for project organizations to provide a way out of the conflict for project planners: no more than 350 tasks; no task is longer than 80 hours; plan at a very low level of detail of the Work Breakdown Structure (WBS), but manage at a much higher level (see box above!). Have these approaches been effective for all projects? Our answer is, "Definitely not." (Sounds more like coping mechanisms and compromises, doesn't it?)

In our opinion, there is a huge problem with using the WBS as the project plan, the way that it is implemented in most of the circumstances we have seen. WBS is defined by the Department of Defense as an organized method to breakdown a product into sub-products at lower levels of detail.[1] These sub-products do not represent the interdependencies of work required to create the product. However, one of the big risks in doing project work is in the handoffs between resources and the interdependencies between tasks. When there is so much detail with sub-products, you lose focus on these interdependencies.

We recommend doing project planning by creating a project network – an interdependent relationship of tasks (boxes) and flow of work (arrows) that are required in order to achieve the goals, scope, and sponsor criteria of the project. (More on actual building of the project network in upcoming blog posts)

The less detail/more detail conflict exists for project planning. Our solution is to approach project planning from this perspective: *Start planning at a high-level, then "explode" the plan into more detail only when and where needed.*

The 10 Step Process

The 10 steps to build a robust network at the right level of detail to meet stakeholder needs with minimum risk are:

1. Define the project's measurable goals, tangible scope and sponsor criteria.
2. Define the backbone of the network.
3. Expand the skeleton of the network
4. Define additional dependencies
5. Check the network against project goals, scope and deliverables
6. Scrutinize with subject matter experts
7. Resource the project tasks
8. Estimate time durations
9. Reduce duration without compromise
10. Perform a final risk assessment

Conclusion

There are two commonly used approaches to project planning, both of which do not work well. One is "seat of the pants" where projects are run without a formal, written, scrutinized plan. The other is a plan worked to the lowest level of detail of a work breakdown structure. Such a plan is so detailed that the underlying problems are masked and the plan is very difficult to scrutinize. In the 10 step process outlined here and defined in the upcoming posts, detail is only advocated where absolutely necessary because there are task interdependencies or other crucial elements of scope. We believe excessive detail does not help control a project – in fact, the outcome is often the opposite. This belief is backed up by years of experience with organizations who have used this process to build project networks with much greater than average project success.

Reference:

1. Department of Defense Standard Practice, MIL-STD-881C, *Work Breakdown Structures for Defense Materiel Items*, 3 October 2011, 4.

Next Post

- Step 1: Define the project's measurable goals, tangible scope and sponsor criteria.

Building Executable Project Plans – Part 3

Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Basis for these posts: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J.Ross Publishing, 2012.

Finally, we begin the 10 steps to building an executable project plan!

Step 1: Define the project's measurable goals, tangible scope and sponsor criteria

Background: Users often complain about project outcomes. Scope creep is one of the most voiced issues in project management. No wonder. In the vital "giving birth" stage of a project, once again, we witness two extremes. Either we see a 20+ page document, sometimes called a Project Charter, that has so much detail in it that it is almost guaranteed to put you to sleep, and further, to not be very useful in gauging whether or not the project will meet all of the key stakeholder needs. The other extreme is brief project scope and/or objectives statements that are so vague as to be proclaimed an open invitation for scope creep within days of project initiation.

This step is analogous to having a blueprint for building a house. Before you start building, you must have a pretty good idea of how big the house will be, and its dimensions, or you will not know where to put the footers nor how strong the footers must be. Everything else flows from this key front end definition of stakeholder needs. Do not skip this step nor any of the elements described below!!! Have your attention? Good! There are too many anecdotes about a project's budget and timeline being almost totally gone when the project team finds out that what's being produced is not what the stakeholders expect. Oops!

What is the desired result of Step 1? The project planners and all stakeholders understand and agree upon the goals, scope, sponsor criteria, functional criteria, and boundary conditions for the project.

There are two parts to this step:

1. Preparing for the Project Stakeholders' meeting;
2. Holding the Project Stakeholders' meeting;

Note: The material below assumes you are responsible for project planning

Preparing for the Project Stakeholder's Meeting

Meeting Attendees: First identify the project stakeholders and other key people who should be invited to the meeting. If you're not sure, these questions may help in developing a checklist of meeting invitees for this and future projects:

- Who does the Project Manager report to for this project?
- Who is sponsoring this project?
- Are there customers for the project? Please note that a project's customers may be internal or external to the organization; at times there can be both internal and external customers.
- Who is providing funding for the project?
- Who provides resources for the project (human, equipment, facilities)? Again, these resources may be internal or external – who represents each of their interests? Do unions represent any of the resources? If so, are they represented?
- Are there key functional areas that should be represented? Engineering (and those sub-sets), production, marketing, sales, contracting, legal, safety, product development, technical documentation, distribution, etc.?
- Are there additional people (internal or external) that require project progress reports?

Meeting Logistics: After determining who should attend the meeting, determine when and where the meeting will take place. It may also be important to set up video-conferencing for some attendees.

Meeting Self-Preparation: This is not so that you can provide all the answers at the meeting! ☺ It is so you begin to understand what you will hear from the attendees as well as giving you the "big picture". This preparation will also help ensure nothing is left out at the meeting.

- Gather and study any existing project documentation, including any planning documents, trade studies, etc.
- Ensure you understand how each of the customers plans to use the output of the project - similarities and differences.
- What is the scope of this project? How many, what is involved, which locations? Is the scope the same for each customer?
- What are the tangible deliverables and what functional requirements must they meet?
- What is the project's output – an analysis, a prototype, a production-ready unit, a quantity of product delivered to multiple distribution warehouses with technical

support fully-staffed and ready to answer questions, a new product available at all locations with full advertising campaign and fully-trained sales staff?

- What impact does this project have on your organization: What is the predicted bottom-line positive garnered from delivering the project on-time, within budget, and with full scope? When does the organization expect to begin seeing those bottom-line impacts? If there is not a bottom-line positive impact, why is the project being done? There must be a benefit statement of some kind – expect it to be measurable, even if it is qualitative rather than quantitative.
- What are the budget(s) allocated to the project (remember, a financial budget is not the only type of budget)?
- What are the risks to the project's success? Include known significant technical, schedule, and budget risks. Will the major items that the project needs (e.g., equipment, prototypes, software, long-lead-time items) be available when promised? Will these items likely be according to required specifications? Have those risks been documented and their impact calculated?
- What assumptions are being made about the project (by any of the stakeholders, including you)?

Now that the pre-meeting homework is done, you're ready to conduct the meeting: gathering data from the experts in the room in a manner that fully accomplishes the meeting goals without wasting time or money.

TIP: Consider whether the pre-work will be similar for all projects. In many cases, it makes sense to follow a checklist in preparing for every project meeting. Here's a sample:

Project Name
Meeting Attendees? List required and optional
Project Access, Information, Considerations, Security, Sensitivities?
Project Due Date(s)? What is required on this date/each date?
Project customers (internal & external)?
Project Documentation Required? Specify both internal and external requirements
Project Reviews? Specify type, quantity, and dates, if already scheduled
Customer furnished data, drawings, materials, specifications, etc.?
Required Tests? Specify types, locations, requirements, attendees
Minimum or Maximum intervals between tests?
Test data? Sent to whom? In what form and format?
Required Permits or Certifications? Specify types and sources
External Resources or Facilities? Specify

Long-Lead items?
Public Affairs/Legislative constraints?
Marketing/Sales/Legal constraints?
Global manufacturing constraints?
Government involvement/constraints? Specify level, agency, involvement/constraint
Assumptions?

PRE-STAKEHOLDERS' MEETING SAMPLE CHECKLIST TOPICS

Project Stakeholder's Meeting

Ensure that the person whose role is to facilitate the Project Stakeholders' Meeting uses this standard approach. Remember, in a multi-project environment, if some project networks and scope are defined poorly, this will create chaos for ALL the projects during execution.

TIPS:

- All the attendees need to be able to see what's written in its entirety. Have available multiple whiteboards or chart paper that you can stick on the walls. Since this will be a multi-page outcome, using a computer is only practical if you are able to print what you capture for all attendees as you go.
- Remember: The facilitator's pre-work is to get in the right mindset to listen and gather data at this meeting, not to tell the attendees what the project should be!
- Follow the meeting outline!

1. Present this meeting's goal: "The project planners and all stakeholders understand and agree upon on the goals, scope, sponsor criteria, functional criteria, and measurable outcomes for this project."
2. Identify the scope and goal(s) of the project. What IS this project? There may be one or multiple statements of scope elements. Identify whose goal (perspective) it is as you go, since different stakeholders/key people may state the same goal in different ways or it may sound the same but have a very different meaning to some attendees. This also helps determine whether there are conflicting goals (or whether they are just stated from a different perspective). The facilitator is the "clearinghouse" – making sure everyone understands what's been said and that terms have the same meaning for all.
3. Make sure that the goals are expressly and measurably related to the goals of the organization. How much new revenue will be generated? By how much will operating expense be reduced? How much will this project add to the organization's profits? How much faster will patients get through an emergency room? How many more aircraft will be ready for missions and testing? How much shorter will customer lead-times be? Etc.

4. Ensure all customers/types of customers are listed, including what needs this project meets or issues this project resolves for each of the customer types. Reminder: Customers may be internal, external, or both.
5. Define the tangible deliverables – the project's output(s). How many, in what form, delivered to which customer, delivered where (to the loading dock, delivered to a distribution facility, etc.)? Are there any interim outputs that must be provided? Are any of the interim outputs date sensitive? Are the outputs products, drawings, an analysis, a prototype, a production ready unit, a fully-ramped up production line, etc.? What must accompany the output? Ensure the functional requirements or specifications of the project's output(s) are defined as well.
6. Are there any major items needed for this project (e.g., equipment, software)? When will they arrive? From whom? What are the specifications required for those items?
7. When does your organization start making money (gaining bottom-line benefit) from this project? Do the goal and scope statements reflect that? For example, is the project to construct a new production facility or is the project to be at full productive capacity in the new production facility? There is a big difference between those two project scopes! The goal and scope should always be stated in a way that reflects when your organization realizes bottom-line positive value.
8. Are any customers imposing mandatory reviews? Are there contractual milestones?
9. What are your budgets to accomplish this project? Budgets can be financial, facilities, headcount, or a combination.
10. Describe the technical, schedule, and budget risks to the project.
11. Define any sponsor criteria that have not yet been defined.
12. Document any additional stated assumptions.

When you have achieved the meeting's objectives, summarize and close the meeting. After the meeting, review what's been gathered. Do a final check to make sure all statements and documentation are clear. Ensure all attendees receive a copy of what's been documented. Here's a sample format:

Scope/Goal(s):
•
Customer(s):
•
Tangible Deliverables/Functional Requirements of Output(s):
•
Major Items / Specifications Needed as Project Inputs:
•
Bottom-Line Impact to Organization:

•
Budget(s):
•
Risks:
•
Sponsor Criteria:
•
Assumptions:
•

TEMPLATE FOR STAKEHOLDER MEETING RESULTS

Food for thought: Kitting has become an important aspect of accomplishing projects smoothly. It is certainly too early to identify and evaluate all kitting opportunities, but some kits may be able to be identified already. Do not miss the opportunity to gather kitting ideas, but do not let this diffuse/dilute the meeting's purpose.

Summary

To avoid scope creep and its ultimate consequences – projects delivered late, over budget and/or not within scope, a careful process is needed. The process includes people who must be invited to a meeting to define the project, topics to be covered, and documentation of the outcomes. This documentation forms the basis of checking every task in a project plan, to see if it is needed and if it, in fact, helps the project meet these stakeholder criteria. Furthermore, this documentation allows the team to check that all criteria are met, once the network is finished. To be useful, such documentation must be organized, succinct and directly related to the goals of the organization.

Next Post

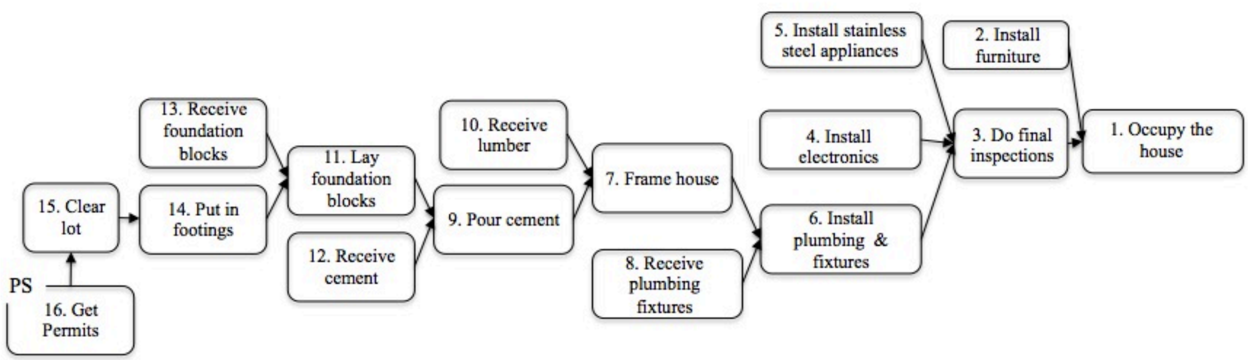
- ❖ Step 2: The Backbone of the Project Plan

Building Executable Project Plans – Part 4

Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Part 3 began the first of the 10 steps to building an executable project plan. Basis for these posts: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012. Note: Some figures in this part of the blog begin with "14-"; this indicates the book chapter for reference.

Step 2: Define the tasks required for the backbone of the project network (one main path), starting at the end of the project and working toward the beginning.

Network building begins only after Step 1, the stakeholder needs, are clearly documented and understood! Every project network has some long strings of tasks with many other tasks "feeding into" these strings. For example, consider the string of tasks in building a house. After the house is framed, other strings of tasks would be performed before the builder would arrange final inspections (e.g., electrical, plumbing, dry-walling, painting, etc.). But to start building the network, we only consider one of these strings, and we'll use the example below as a starting point as well as the results of the stakeholders meeting, also shown below.



Step 2.1 Example

Step 2 defines the tasks required for the backbone (or spine) of the project network, starting at the end of the project and working towards the beginning. This is where the leadership of the person who thinks network building is "fun" is crucial! In addition, include the project manager, and one or two other team members. The team members should be people who understand the overall project requirements, how the organization works, and all resource skill sets in general. If the organization has a project management office (PMO), someone from this office would be involved. These team members will remain for all following steps (important!)

Scope/Goal(s):
• Ready to occupy 3000 sq ft lakefront, ranch-style, 3-bedroom, two-office, brick home with attached 3-car garage no later than Oct. 15th
Customer(s):
• Mr and Mrs John Doe (no children, no pets)
Tangible Deliverables/Functional Requirements of Output(s):
• Hardwood floors in all but kitchen and bathrooms • Ceramic floors in kitchen, bathrooms, mudroom • Spa-like master bath • All granite countertops • Stainless steel high-end appliances, including icemaker, gas range, double ovens, and warming drawer • Custom birch cabinets and finishes • High-end finishes throughout • Lakeside: floor to ceiling windows, French doors • Full house electronics (TV, games, security, high-speed wireless, intercom and whole-house surge protection) • Summer kitchen on extended lakeside deck • Full formal landscaping including path to dock and boathouse
Major items / specifications needed as Project inputs:
• Hot tub on lakeside deck
Bottom-Line Impact to Organization:
• $ 50,000 net profit
Budget:
• $450,000
Risks:
• Permitting and inspection processes • Weather • Tile contractor • Electronics installer
Sponsor Criteria:
• Deck and dock of Timber-Tek like material

Output of a Stakeholder's Meeting

Before We Begin: Some Network Building Tips…

- Use sticky notes (Post-It®[1] type notes or pieces of paper and Post-It® glue sticks) large enough to write on and large sheets of paper to begin network building.
- We strongly recommend that you do NOT begin building your project network on the computer; build it on large paper so that you and your team can see the interconnections and add to them. Put the network into the computer only when you are ready to print or plot it.

First Box on the Network Diagram

2.1. Using the Stakeholders Meeting Results from Step 1, pick a tangible deliverable that is produced later in the project. At this point, do not worry about the other tangible deliverables; they will be built into the network in subsequent steps. Using the example above, we select the item of scope to have the house available to occupy by October 15th. The verb is:

> "Occupy the house."

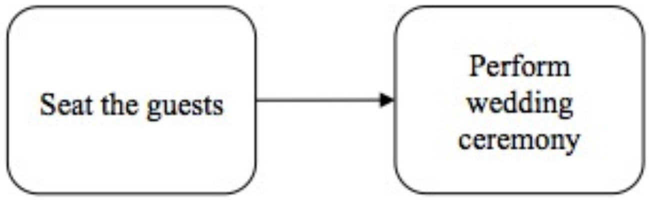
Write the selected starting point on a Post-It® and place it on the far right of a large sheet of paper. The tangible deliverable should be written as a sentence minus the subject, starting with an action verb. Note that the date is not included in the task name; this will come as a result of the scheduling. Note also that there are no resources or times associated with the task at this point. To start: (No subject) action verb object.

Some ask why we build a network backwards, starting from the end, rather than forward, starting from the beginning. Working backwards, we include ONLY those tasks that are absolutely necessary as inputs to the next step. By working this way, we find that much less detail is required, yet the end result still meets all stakeholder needs.

For example, assume we have a wedding project. If you try planning this from the beginning, you can probably think of dozens of tasks that need to be done, all muddled together in your brain. However, working from the end, the last task is for the clergy/official to marry the couple. What must we have immediately before we accomplish this task? We must have the couple in place, the guests seated, the clergy/official ready and the wedding music playing. We can continue working backwards on any one of these streams, and will find that there are typically only one or two items at a time we need to identify <u>immediately</u> before each task. The network is constructed in a logical, orderly way and we only identify those elements that we must have to meet the end goals.

Boxes and Arrows

A project network consists of boxes and arrows. The boxes contain the work of the project and will have resources and time estimates identified in later steps. The arrows are used to indicate handoffs of work between boxes.



An arrow indicates that a box on the right of an arrow **must have** the input from the box(es) to the left of the arrow before it can start work. Output from the left is input to the right[3]. For example, as shown above, we must have the guests seated before performing the wedding ceremony. There are actually two types of "must have", which are discussed later.

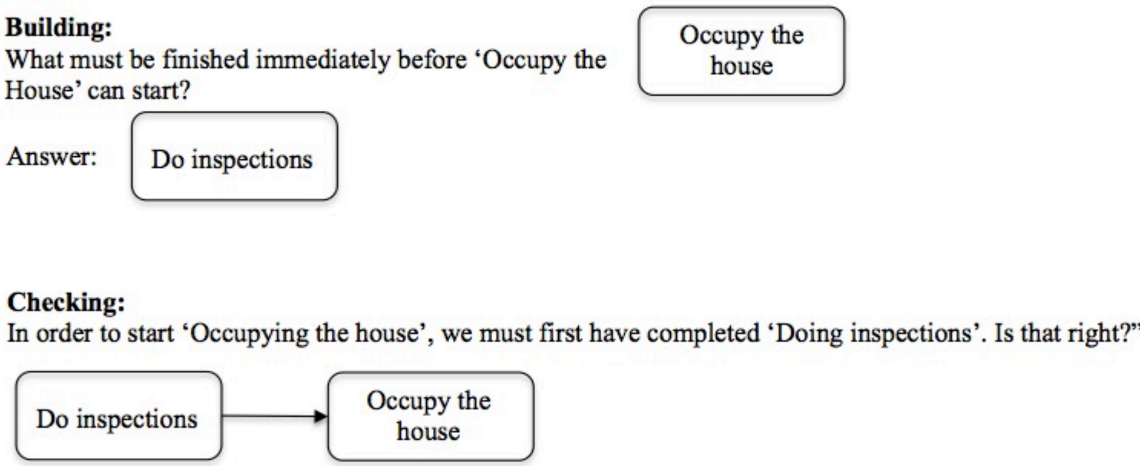
Building the Backbone

Building the backbone requires knowing the first box you will use on the network diagram and understanding how boxes (tasks) and arrows are used. We have that information so it's time to understand the process for building the backbone.

2.2. Ask, "What must be completed (or finished) immediately before *the task on the right* can begin?" E.g., "What must be completed immediately before 'Occupying the house' can start?"

Tip: [Asking out loud helps you to listen/think about the question more closely. Fill in the underlined words with the actual wording of the task on the right – the task at the tip of the arrow.]

Another way of asking the question is, "What input is required before the Does can begin occupying the house?" The answer should be written on a Post-It in task format (starting with the action verb) and placed immediately to the left of the first task. Check the logic by saying, "In order to start *(the task on the right)*, we must first have completed *(the task on the left)*". If the team agrees, write the arrow connecting the two tasks. Note that sometimes "ing" must be added to the verbs to make the building and checking sentences flow. See below.



Building & checking task dependency

2.3. Ask, "Is there anything else that must be completed immediately before *(the task on the right)* can start?" or "Is any other input required to be able to start *(the task on the right)*?" Use a Post-It to capture that task, ensuring it is written in the correct format (starting with an action verb, no subject, sentence that describes what must be completed). Check the logic by saying, "In order to start *(the task on the right)*, we must first have completed *(the task on the left)*". If the team agrees, write the arrow connecting the two tasks (tail of arrow at predecessor, tip of arrow at successor). Repeat this until all tasks that must be completed immediately before the starting task have been identified. Don't forget to build and check every dependency using the wording above. This is also a good time to start uniquely numbering each task for identification purposes. This will make it much easier for people who have a comment or question on one of the many boxes to describe which box they are referring to (see below).

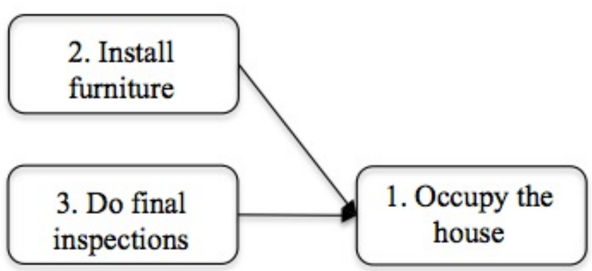


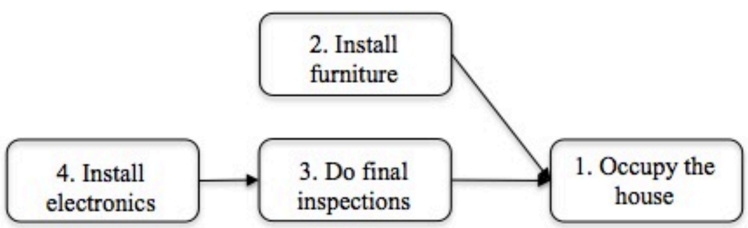
Figure 14-4: Result of Steps 2.2 and 2.3

Sometimes there can be disagreement among the team members about whether or not a task belongs immediately before the task on its right (whether its output is required as an input to the next task or whether it is needed earlier in time). If this occurs, leave the task where it is for now; if it truly is needed earlier, the process will identify that and the task can be moved earlier at that time.

Notes: This process can seem counter-intuitive. E.g., "Install furniture" must be complete before "Occupy the house" can start, and the next step for many is to look for what must be completed immediately before "Install furniture" – what we would call looking for the linear flow. Do not fall into this trap! Identify all the required tasks required immediately before the task on the right – that is the best way to ensure all the inputs that the task on the right needs to begin will be available. There is no fixed number of arrows required for a task. Remember to identify ONLY the inputs needed to begin the task on the right. This step is to identify all the significant necessary dependencies (predecessor tasks) to the task on the right (successor task).

As a final check before moving to the next step, read the connections as "In order to achieve (task on the right), we must first complete (task on the left) and (task on the left) for as many predecessor arrows as exist. In the example above, it would read, "In order to occupy the house, we must first install furniture and do final inspections." If any additional necessary dependencies come to mind (that must be accomplished immediately before the successor task can start), please add them appropriately.

2.4. In order to continue building the backbone of the network, examine each of the tasks to the left that you just inserted. Based on the team's understanding of those tasks, which one requires the most work to be done? That will be the task to build the rest of the backbone from. Repeat steps 2.2 and 2.3 to identify first one task to the left of your newly selected "right" task and check it and then identify all other required tasks to the left. See below.



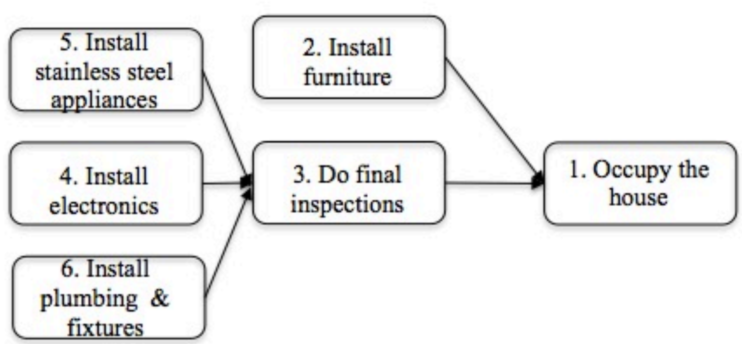


Figure 14-5: Result of Step 2.4

2.5. Repeat Step 2.4 until you have reached a task that could be started today, if the project go-ahead was given. That task is called a path start and should be designated as such. Typical conventions are to put a "PS" on the task. (Some put "EP" for Entry Point; either is acceptable. Pick one and use it consistently.) Congratulations! Your team has completed the backbone! See below.

Note that we have only worked back to the beginning on <u>one</u> path. Even though other tasks have no tasks to the left of them (e.g., task 13, receive foundation blocks), we have not yet determined that this is, in fact, a path start task. Further steps in the process will complete the network.

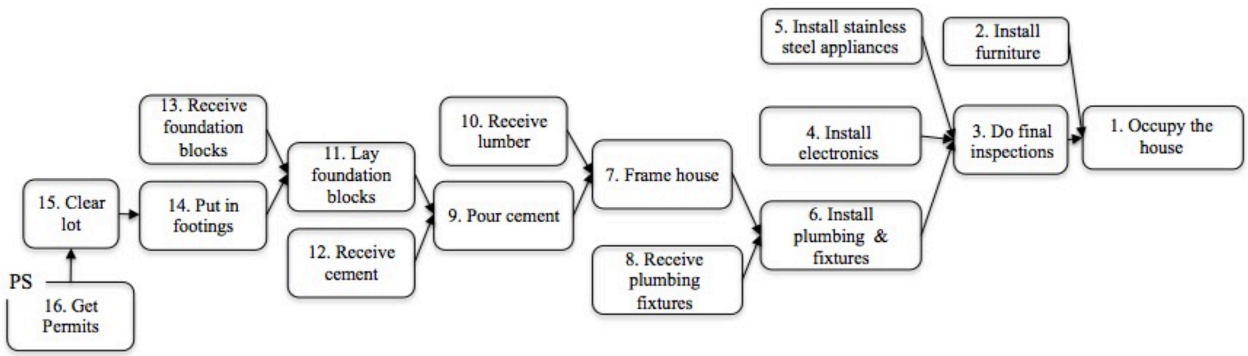


Figure 14-6: Result of Step 2.5

Some organizations require the project network to look like a "football" (American-style) – that is one starting point and one ending point. While we agree on having one ending point, there is not a requirement for a single starting point. Check to make sure that a single starting point isn't really an administrative or overhead task – these should not be in the project network. <u>Only tasks that are necessary to create or produce the tangible deliverables of the project should be in the project network</u>.

As the backbone emerges, it becomes tempting to begin adding tasks that "you know" must be in the final network for the project. Do not do that – yet. It is very important to keep network building at a high level, breaking down into more detail only when and as required. Too often, network builders go too detailed, too early; this makes for networks that are very difficult to manage and can significantly lengthen the time it takes to build a project network. It really doesn't matter what you "know" must be in the network, right now. The required level of detail will make itself known through the process of network building – which does have multiple steps and multiple risk avoidances/risk mitigations built in. This is the beginning of the first pass! Staying at a less-detailed level can be very difficult for experienced project planners and managers. This can be hard to "un-learn"! Please be patient and let the network develop over the required steps!

Conclusions

Network building must begin only when the stakeholder needs have been clearly identified and documented, from Step 1 of this process. Once this is complete, the following steps are performed by a very small team (2-4 people) who know both the overall project goals and the organization very well. The steps to build the backbone are:

Step 2. Define the tasks required for the backbone of the project network (one main path), starting at the end of the project and working towards the beginning.

2.1. Using the Stakeholders Meeting Results from Step 1, pick a tangible deliverable that is produced later in the project.

2.2. Ask, "What must be completed (or finished) immediately before (the task on the right) can start?"

2.3. Ask, "Is there anything else that must be completed immediately before (the task on the right) can start?" or "Is any other input required to be able to start (the task on the right)?"

2.4. In order to continue building the backbone of the network, examine each of the tasks to the left that you just inserted. Based on the team's understanding of those tasks, which one requires the most work to be done? That will be the task to build the rest of the backbone from. Repeat steps 2.2 and 2.3 to identify first one task to the left of your newly selected "right" task and check it and then identify all other required tasks to the left.

2.5. Repeat Step 2.4 until you have reached a task that could be started today, if the project go-ahead was given. This task is called an entry point or path start task and should be designated as such. Typical conventions are to put an "EP" or "PS" on the task.

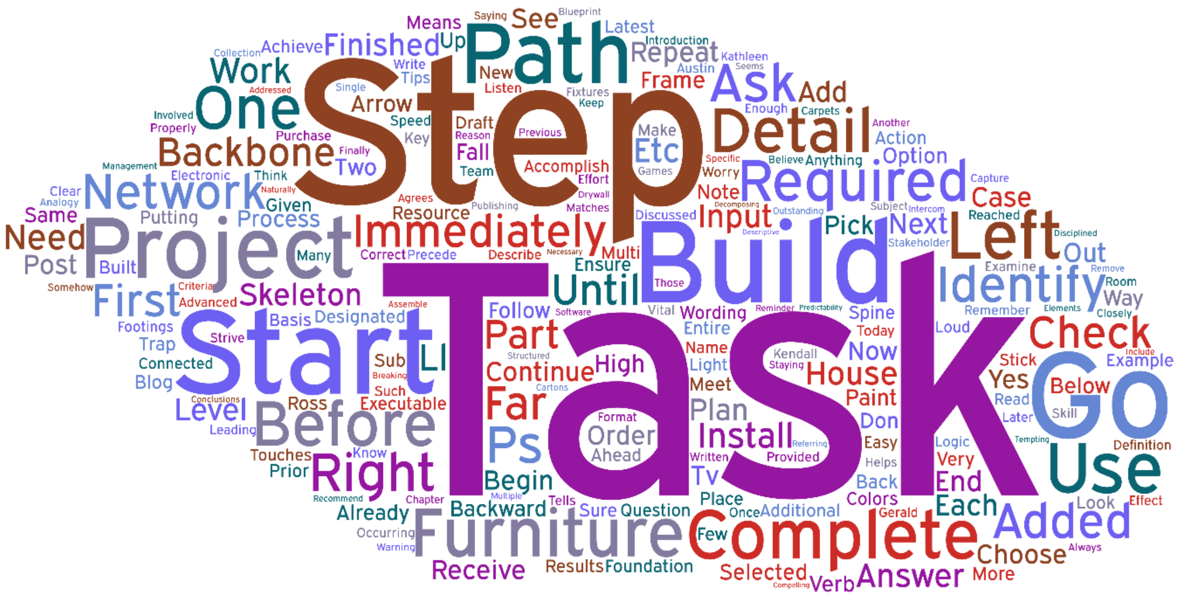
Endnotes

1. Post-It® is a trademark of 3M.
2. Dr. Stephen R. Covey
3. For those readers who are familiar with the jargon of project networks, this means the only arrows used in the project network are "finish to start" arrows.

Next Post

- Step 3: Add the tasks required to build the skeleton (other paths), working backward from the end, completing all other paths

Building Executable Project Plans – Part 5



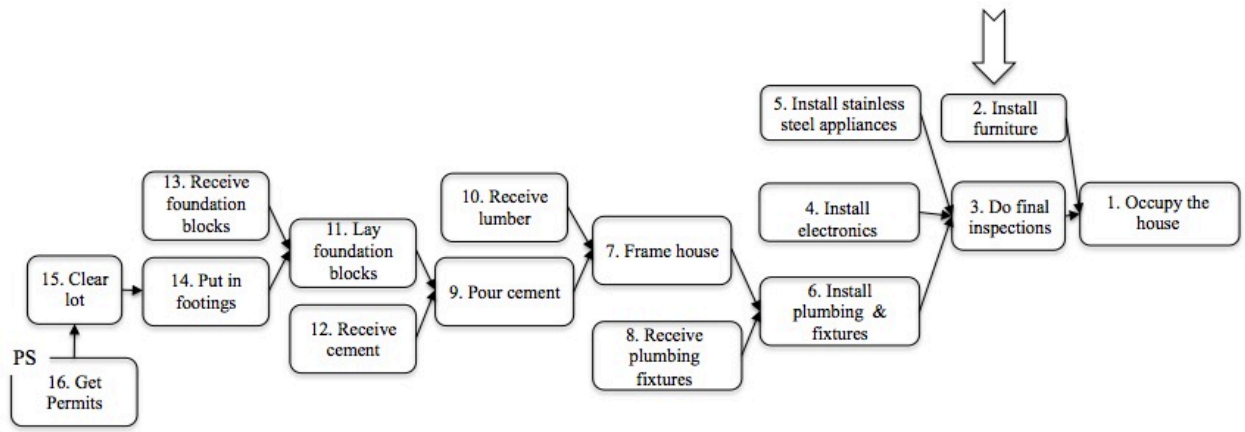
Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 &4 detailed Steps 1 & 2 of the 10 steps to building an executable project plan. Basis for this posts: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 15.

Step 3: Add the tasks required to build the skeleton (other paths), working backward from the end, completing all the other paths.

The spine or backbone of the network was completed in the prior step. Continuing the process, every task or collection of tasks that you add must now be somehow connected to this spine. All of Step 2, the backbone, must be complete before starting this step. The analogy is that in building a house, you must have the footings and foundation before you start framing the house. The backbone of a project network (Step 2) is the foundation.

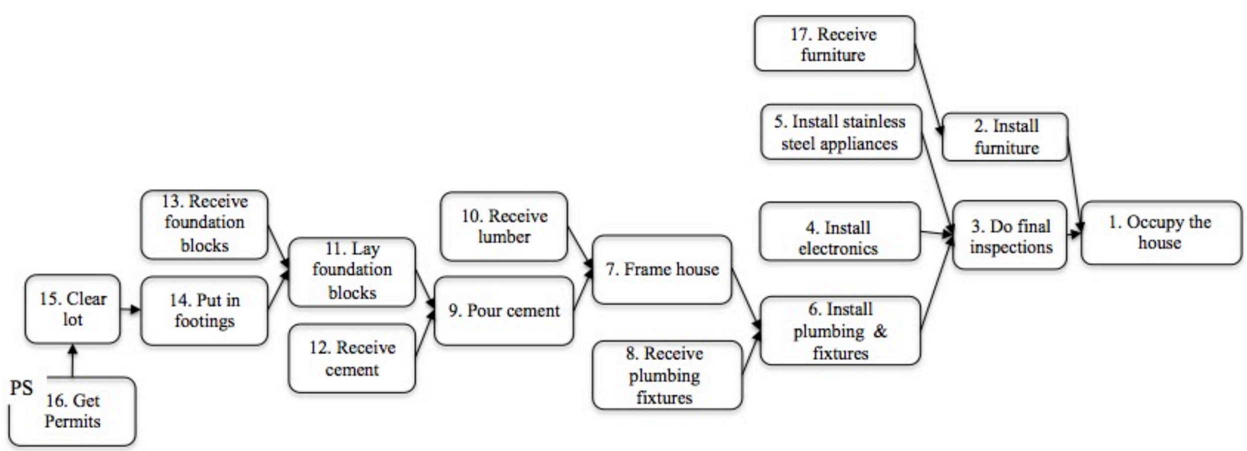
Building the skeleton means adding the additional required tasks and paths of the project, following a structured, disciplined process. Remember to stick to a high level of task definition – you are not putting all the finishing touches on the house. Don't worry about choosing carpets, light fixtures, paint colors, etc. Just make sure that this skeleton matches the blueprint definition of step 1. In this step (Step 3), the same process steps are repeated until the entire project skeleton is complete, meaning the first draft project network has been completed.

3.1. Using the project backbone already completed, go to the far right of the backbone and pick one of the latest occurring tasks. Continuing the example from the previous blog post, there is only one option, Task 2, since we have already built the backbone from Task 3.



3.2: Ask, "What must be completed (or finished) immediately before the task selected in Step 3.1 can start?" [Asking out loud helps you to listen/think about the question more closely. Always read the entire task wording.] Another way of asking the question is to ask, "What input is required before we can begin (the task selected in Step 3.1)?" The answer should be written on a Post-It in task format (starting with the action verb) and placed immediately to the left of the task on the right. Check the logic by saying, "In order to start (the task selected in Step 3.1), we must first have completed (the task on the left)". If the team agrees, write the arrow connecting the two tasks.

See below. What must be completed (or finished) immediately before "Install furniture" can start? The answer is the new task, 17, "Receive furniture." In order to start "Installing furniture" we must first have completed "Receiving furniture."

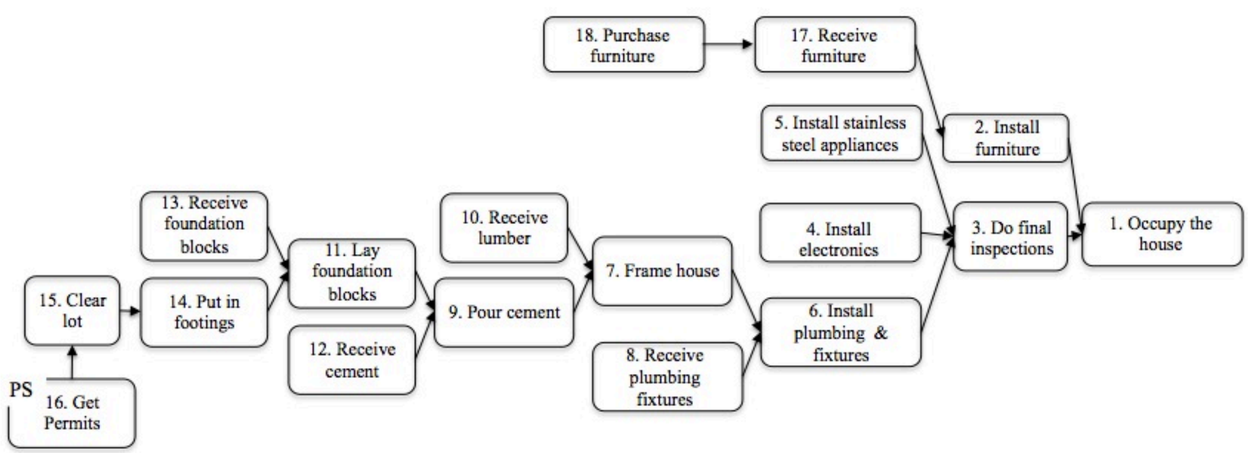


3.3: Ask, "Is there anything else that must be completed immediately before 'Installing furniture' can start?" or "Is any other input required to be able to start 'Installing furniture'?" In this case the answer is no, so no additional tasks are added to the left of Task 2. If the answer had been yes, you would have identified each of the required tasks to the left, using the building and checking wording.

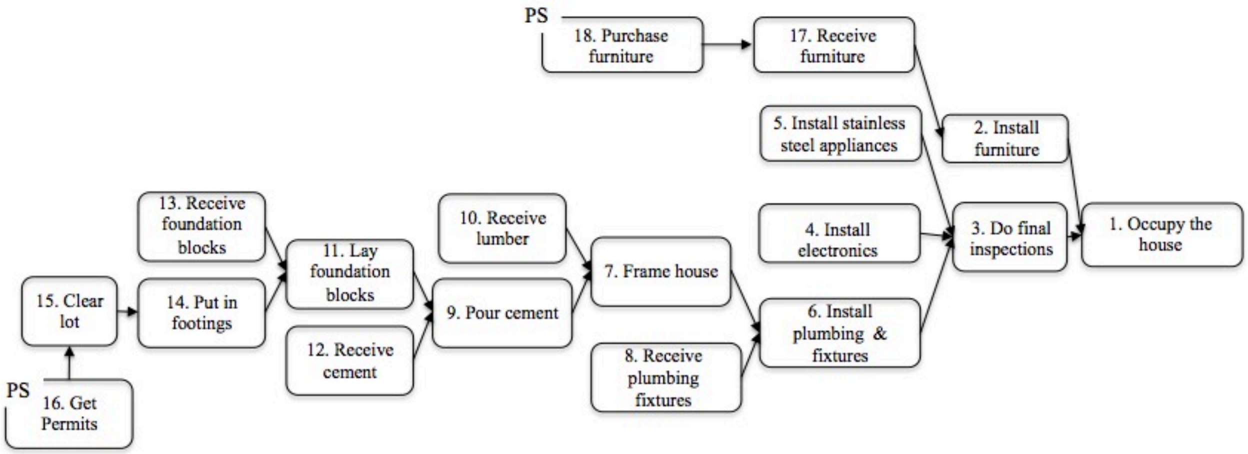
Warning: There is a trap that is very easy to fall into. Many begin adding tasks that describe the steps involved in accomplishing the task on the right, in effect, decomposing or detailing what is needed to accomplish the task on the right. What we are looking for are the tasks that must be completed immediately before beginning the task on the right, not how we are going to accomplish the task on the right. For example, for the task "Install Furniture", it is tempting to detail that task by adding tasks such as "Remove Furniture Cartons", "Assemble Furniture", etc. This is the trap we are referring to, which is describing the sub-tasks of installing furniture rather than what tasks must precede it.

3.4: In order to continue building the project network, we next examine Task 17 and repeat steps 3.2 and 3.3 to identify first one task to the left of Task 17, check it, and then identify all other required inputs to Task 17.

What must be completed (or finished) immediately before "Receive furniture" can start? The answer in this case is task 18, "Purchase furniture." In order to start "Receiving furniture" we must first have completed "Purchasing furniture". Is there anything else that must be completed immediately before "Receiving furniture" can start? No.

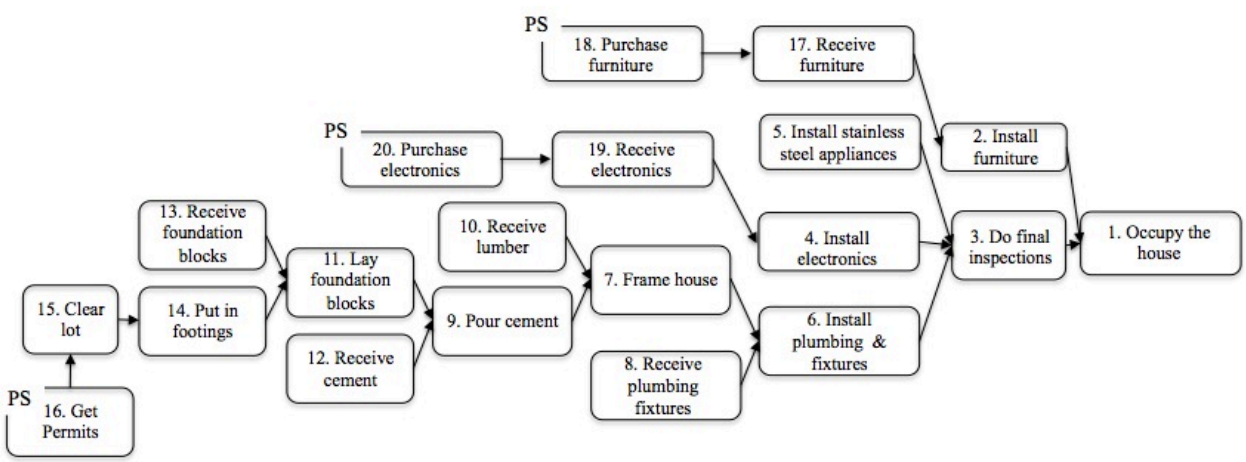


3.5: Repeat Step 3.4 until you have reached a task that could be started today, if the project go-ahead was given. That path start task should be designated with a "PS".

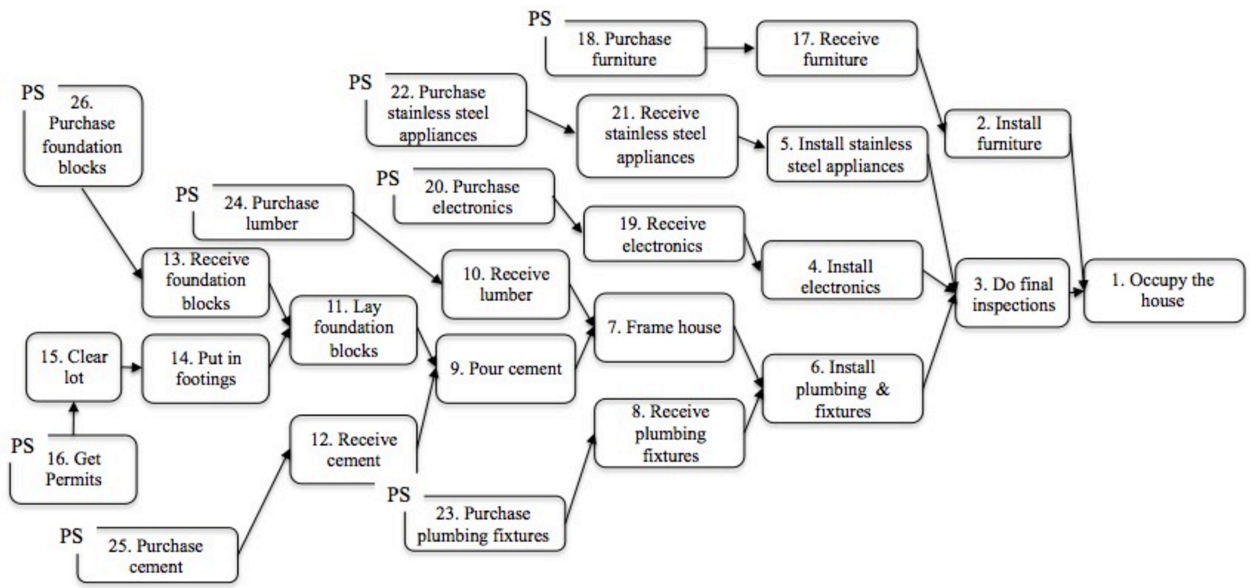


3.6: Follow Steps 3.1 – 3.5 until all tasks with no arrows leading to them have been properly designated as path start tasks (PS).

Continuing our example, now that Task 18 has been designated a path start task, go back to the far right of the project network to look for the next task to build backward from. The two options are Tasks 4 and 5, since Tasks 1, 2, 3, 6, and 17 have already been addressed. We recommend choosing the task you believe has the most work preceding it. For our example, we'll choose Task 4, using the same building and checking questions. See below, where Tasks 19 and 20 have been added. Note that we are staying at a high level, "electronics", rather than breaking the tasks into the specific electronic elements (TV, games, intercom) that are in the Stakeholder Meeting Results. Why? There has been no compelling reason so far, to do so. The process steps will let us know when and where we need more detail in the project network.



Since Task 20 is a path start task, we go back to the next option for building backward, Task 5. Next will be Task 8, then 10, then 12, and finally 13. See below.



Reminder: There is not a requirement to have one single starting task in a project; in a few cases it may naturally work out that way, but do not strive to achieve it.

Tips

- Ensure each task begins with an action verb
- Each task should be descriptive enough that the completion criteria are clear. When using software to capture the tasks: If there is not enough room to capture all the details necessary in the "task name", include those additional details in a task note.
- There should not be a subject (i.e., name of a resource skill) for the task. The subject (resource or resources) will be added in a later step.

Conclusions

Building a skeleton of a project network is like continuing to build a house, once the foundation and footings are in place. It is vital to remember to keep this effort at a high, not very detailed level. You are framing the house – not putting up drywall, not choosing paint colors, not putting in all the detailed, finishing touches. The frame tells you that you will meet all of the key stakeholder needs. The steps are:

Step 3: Add the tasks required to build the skeleton (other paths), working backwards from the end of the project, completing all other paths.

3.1: Using the project backbone already completed, go to the far right of the backbone and pick one of the latest occurring tasks. If there are multiple options, choose the one that seems to have the most work required to accomplish it.

3.2: Ask, "What must be completed (or finished) immediately before the task identified in Step 3.1 can start?" Use the building and checking wording.

3.3: Ask, “Is there anything else that must be completed immediately before the task identified in Step 3.1 can start?” or “Is any other input required to be able to start the task identified in Step 3.1 In this case the answer is no, so no additional tasks are added to the left of Task 2. If yes, identify each of the required tasks to the left of the task identified in Step 3.1, using the building and checking wording. If no, go to Step 3.4.

3.4: Examine the task selected in Step 3.1 and repeat steps 3.2 and 3.3 to identify first one task to the left, check it, and then identify all other required inputs to the task selected in Step 3.1.

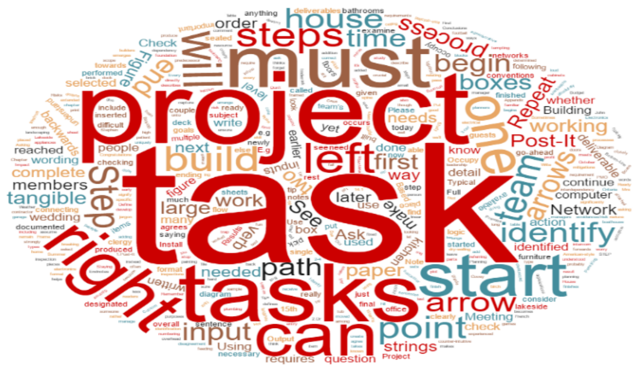
3.5: Repeat Step 3.4 until you have reached a task that could be started today, if the project go-ahead was given. That path start task should be designated with a “PS”.

3.6: Follow Steps 3.1 – 3.5 until all tasks with no arrows leading to them have been properly designated as path start tasks (PS).

Next Post

- Step 4: Read the network forward, from the beginning, rigorously looking for additional dependencies (first risk avoidance).

Building Executable Project Plans – Part 6



Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 5 detailed Steps 1 - 3 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 16.

Step 4: Define additional dependencies

One type of chaos that occurs when executing projects is to assign a resource to a task, only to discover after starting the task that some prior task's output was needed and is not ready. The resource manager had scheduled to use that resource in order to ensure that the project continues progressing. Now, the resource manager has two problems to overcome – first, what to do with this resource, and secondly, when will I actually be able to assign this task and what impact will this have on the project. If this happens once a month, everyone can live with this. But if project network teams do not follow step 4 in the network building process, this type of chaos could well become a daily occurrence. Step 4 significantly reduces this risk.

Finding Additional Task Dependencies

Step 4 is the forward pass of building the project network. Specifically, it is reading the network forward, from the beginning, rigorously looking for missed additional task dependencies. In our analogy to home building, it is like saying "This wall must be built before the fireplace goes in upstairs because it is a load bearing wall for the fireplace. But we never noted before that this is also a load bearing wall for the indoor hot tub. We better make sure that this wall is complete before we move in the hot tub!" This is the first risk avoidance in that we are ensuring that no tasks will be started, during project execution, with missed dependencies.

Before beginning this step, ensure all previous steps have been completed; the tasks, dependencies, and all required notes have been entered into the computer; and a fresh plot / printout of the network diagram is available.

Pick one path start task (denoted by "PS"). You will see that it is already connected to one or more successor tasks. Ask, "Is the output of (this task) needed to begin any other task (for which we have not shown the arrow or dependency)?" An alternative wording would be "What other task cannot start without the output of this task?"

The two keys to this step are to

1. Identify any missing inputs and
2. Ensure no incorrect inputs are added.

See below for an example of an incorrect input. There already is a pathway from Task 25 through Task 12 to Task 9. Adding an arrow from Task 25 directly to Task 9 is **incorrect** because the output of 25 (purchased cement) is not the immediate input for Task 9; the cement must be received (Task 12) before it can be poured (Task 9). The key is to show the correct flow of work as it is passed along the arrow!

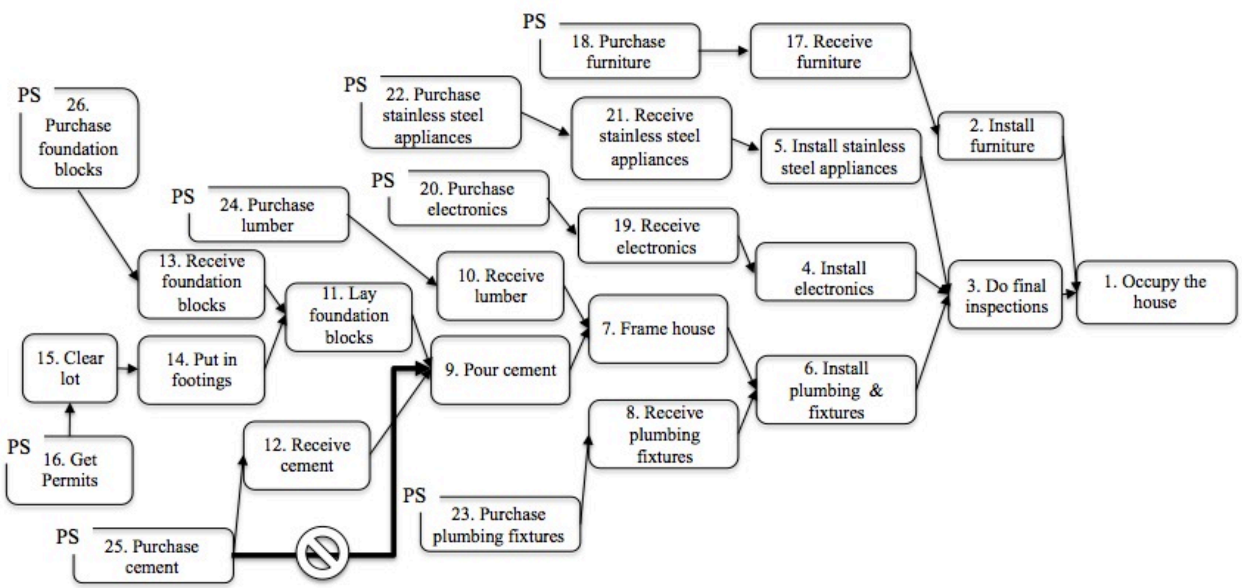


Figure 16-1: Forward pass – incorrect arrow (25 to 9)

Continue along this pathway, using the checking wording above and a colored pencil to help identify the pathways completed, until you reach the last task on the right of the project. Continue the process, going back to an unchecked path start task, until the forward pass has been accomplished for the entire project network.

As the forward pass is being accomplished, there are two typical modifications that are made to the project network:

1. Adding an additional arrow going to another task on the right from an existing task on the left. This means that the output of the task on the left is required as an input to an additional task not previously identified. For example, the output of Task 4 below is an activated debit card. An activated debit card is needed to purchase supplies (Task 3), as shown by the existing arrow in the upper part of the example. An activated debit card is

also needed in order to withdraw cash from the ATM (Task 2), so the new arrow is drawn from Task 4 also to Task 2.

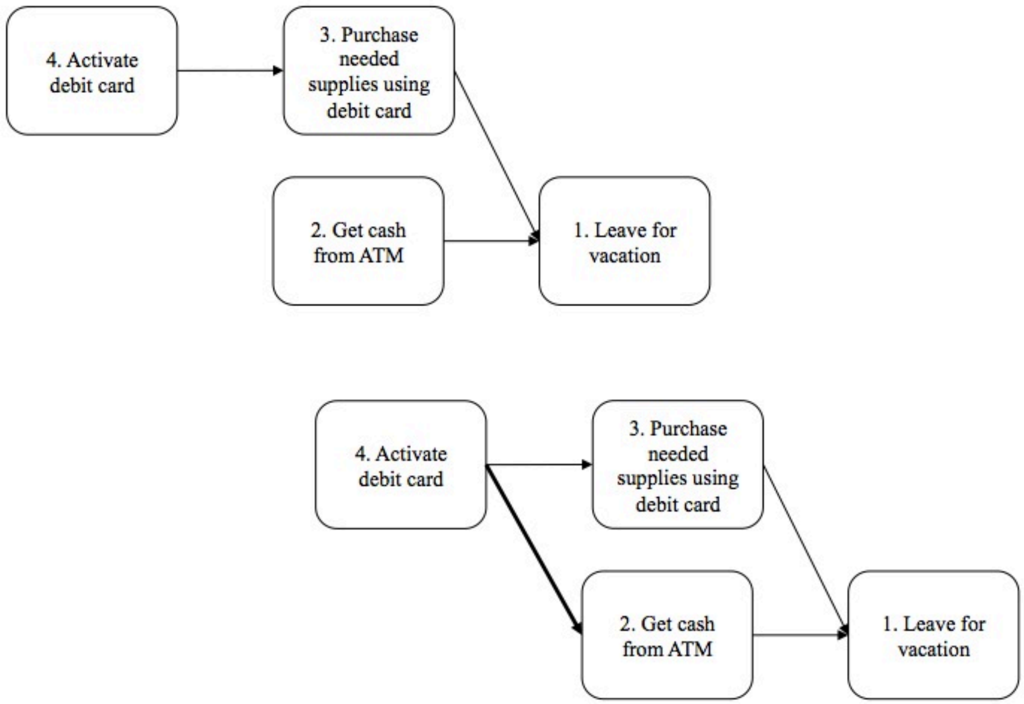


Figure 16-2: Forward pass, adding an arrow

2. When it looks like there should be an arrow from one task to another (below, from Task 4 to Task 2), but the output of one is not the IMMEDIATELY required input of the other, it is likely that there is a task missing between the two. It seems that “Reserving the church” must be an input to “Seating the guests”. Otherwise where would you seat the guests for the wedding ceremony? But the church reservation (output of Task 4) is not what is required in order to begin seating the guests. However, reserving the church is an input to decorating the church (Task 5) and decorating the church is an input to seating the guests (Task 2). So in this example, the forward reading of the network has helped identify the missing task and arrows, as shown.

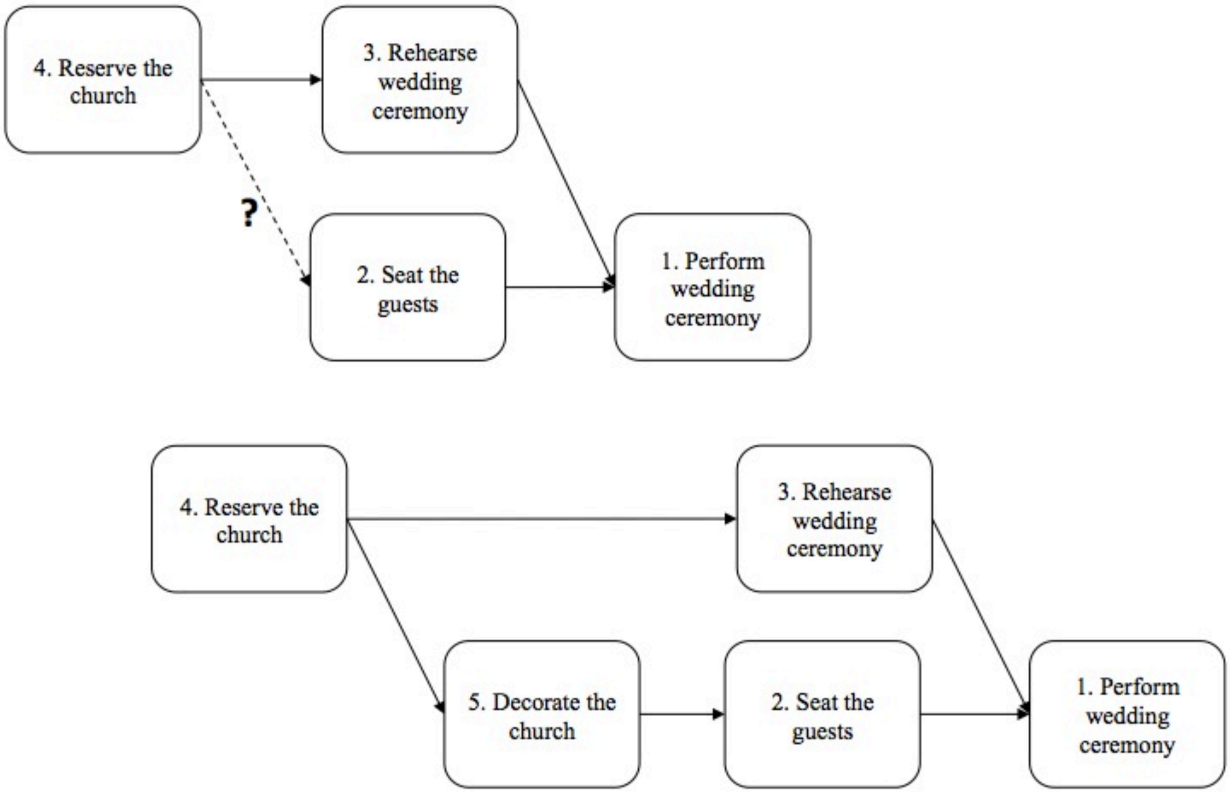


Figure 16-3: Forward pass, adding an arrow and discovering a missing task in between

Step 4 is not complete until all pathways have had the forward pass performed.

Tips

- Do not add extra, incorrect pathways. If tempted, ask yourself if the output of the task on the left is needed in the exact same form/format as an input for the candidate task on the right.
- Colored pencils can be useful for indicating where you have and have not done the forward pass on your plot. They can also help highlight when you've accidently created an extra incorrect pathway.
- This is not a step used to dive into deeper detail in the network. <u>This step's only purpose is to identify missing dependencies.</u>

Conclusions

Every project encounters some surprises during execution. The fewer the surprises, the more likely all projects will complete on time, on budget and within scope. While some surprises are unavoidable, this chapter helps eliminate or at least drastically reduce the surprises from missed task dependencies. This is the kind of surprise where a resource starts a task, only to find that they needed the output of another task in order to progress on the current task. This could mean that they encounter a wait period, forcing them to multitask on some other task, or they have rework on the current task. Further, the completion of the task is delayed and this forces resource managers to waste precious time rescheduling. Step 4 of the network building process avoids this risk by having the team read the network from beginning to end, rigorously looking for these missing task dependencies.

Next Post

- Step 5: Check the network against project goals, scope, and deliverables.

Building Executable Project Plans – Part 7

Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 6 detailed Steps 1 - 4 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 17.

Step 5: Check the network against project goals, scope, and deliverables – 2nd risk avoidance

This step provides three types of risk avoidance:

1. Avoids the risk of missed scope, by ensuring that all the tasks required to meet the project's scope and goals are included in the task interdependency diagram.
2. Avoids the risk of wasting resources, by ensuring that no extra tasks are included that exceed the project's scope and goals!
3. Prevents some risk of scope creep, by ensuring that the tasks are sufficient to meet goals and deliverables that were discussed during the initial team meeting.

Specifically, Step 5 checks every task against the project goals, scope and sponsor criteria identified in Step 1 (Project Stakeholders' Meeting). One way to think about this step is that it ensures that what the stakeholders require has been translated into work specifics for the projects' resources and managers.

Ensuring that the Project Meets All Stakeholder Needs

Do not begin this task until all previous steps have been fully completed. Gather the documentation created during and after the Project Stakeholders' Meeting. Make sure there is a clean plot / printout of the task interdependency diagram as well as all task notes available.

Figure 17.1 Check Network Against Scope

Carefully check the Project Stakeholders' Meeting results template against the project planning done so far. See Figure 17.1 – Did we remember to include tasks to put electrical outlets in the kitchen island? Do we have a task for ordering the microwave? Does the task have the correct microwave specifications? Some items from step 1 will be explicitly shown (the tangible deliverables), while others are implied by the work that will be accomplished. Where criteria have been specified, ensure that is documented in the task description or task notes so that proper exit criteria for a task or pathway reflects that criteria.

If items of scope or tangible deliverables are missing, add those tasks and build in the required dependencies using the established processes for building and checking. Ensure both a forward and backward pass are done to mitigate the risk of missing any required dependencies.

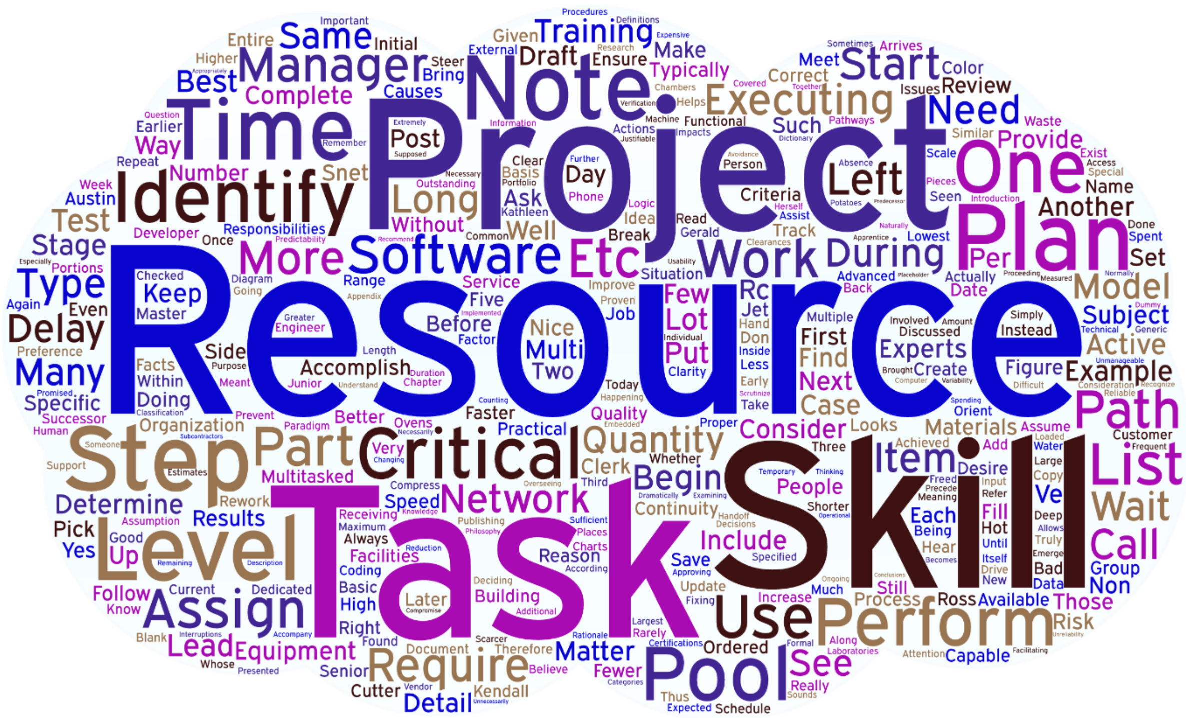
When tasks are in the network diagram that are not needed to meet the project scope defined in step 1, delete them. Use the standard building and checking processes to ensure correct relationships with the remaining tasks.

Conclusions

Any multi-project environment deteriorates quickly when project after project yields unpleasant surprises. No one expects a project network to be perfect, but when every network is full of holes, the results are predictable – many projects will finish late, over budget and not within scope. Step 5 plugs the hole caused by missing key stakeholder criteria. It also avoids doing unnecessary work by checking every defined task in the network against the definition of project scope from Step 1. The result is a far more robust network with a much higher probability of meeting stakeholder expectations without big surprises.

Next Post: Step 6: Determine resources (skill level and maximum number) that could be assigned to perform the task during project execution

Building Executable Project Plans – Part 8



Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 7 detailed Steps 1 - 5 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 18.

Step 6: Determine resources (skill level and maximum number) that could be assigned to perform this task

The high-level network diagram of the "work" of the project has been completed and checked. The next step (Step 6) is to determine resources (skill level and quantity) that could be assigned to perform the task. Note that this step does not assign named resources (specific people) to perform tasks. Instead this step identifies the number and type of resource skill(s) capable of performing to the required level of quality and task completion criteria.

In keeping with the desire to increase the speed of executing the project, we will identify as many resources as practical to each task in this step. The more resources we can identify to a task, without causing unmanageable waste in executing the task, the fewer tasks we will have active. The fewer the number of active tasks, the more management and support group attention can be given to active tasks, and therefore the shorter each task will wait for decisions and actions by any other group in the organization. Also, if proper consideration is given to the number of resources we can identify to a work on a task, the faster the task will complete.

“Identify” is not the same as “Assign!” For clarity, during project planning we **identify** the types and quantities of resources for tasks. During project execution, resources are **assigned** by name to work on a task; the assignment happening according to specific procedures.

Some Basic Facts About Resources

While people are the most common type of resources that perform task work, they are not the only project resources. Other resources can include equipment (ovens, cutters, test equipment, etc.) and facilities (laboratories, chambers, buildings, ranges, etc.). We have even seen (only rarely, and for well-documented reasons) clearances and certifications used as resources. This is almost always better left to execution.

Not every resource needed to perform a task is identified. Critical resources, meaning those that the task would wait for, must be included. For example, in a research project, assume there is a task to “Review test results”. While five people are involved in the review, it is three days of a senior engineer’s time that this task will wait for. Much of this person’s time is spent overseeing the review and approving the results before proceeding to a next step. But, because there is technical information presented during the review that is good for other resources to hear, other resource skill levels believe they must also be listed for the task. Be very clear and specific as to what is required for the task; if it is a task that only a few resources are actually required, the other resources can be listed in the task notes as “Nice-to-have” resources. A deciding factor as to whether a resource is critical or ”nice to have” is to ask if the task would wait/be delayed if the resource skill level was not available. If yes (would delay), the resource skill is critical; if no (no delay), the resource is only listed in a task note and not modeled in a project plan.

We need to ensure that the identified critical resources are not multi-tasked (asked to perform more than one project task at a time). Typically, critical resources are required for the entire length of a task. In Step 6, we are still in a planning, not execution stage. Typically, all software available today has the capability, during planning, to prevent a resource from doing two project tasks at once, by following a process called resource leveling. However, what the software does not know is that a resource may have non-project responsibilities that require 1 day per week, for example.

Since it has been proven time and time again that multitasking delays projects and causes rework, it is important in the planning stage that resource and functional managers recognize the bad multitasking caused by frequent interruptions of project task work, and consider ways of facilitating more dedicated resource time. Sometimes, this can be accomplished by simply doing a temporary assignment of all other responsibilities of the resource to another individual. This is important to consider during the planning stage, because multitasking dramatically impacts the time required to complete a task (and thus increases project risk).

Critical Resource Skill Levels and Quantities

When determining the critical resource skill levels for project planning, do not use organization charts or phone lists; they will steer you to including all resources, whether critical or not. We have found that identifying the critical resource skills on the first projects planned using these steps is the best way to identify not only the resources that are truly critical, but also the "master" critical resource list (See Appendix C of *Advanced Multi-Project Management*, Kendall and Austin, for a generic example).

Resource Pools

Typically, an organization is more than one-deep in a resource skill. Resources of similar skills and levels are grouped into a "pool". Resource pools can have multiple levels, such as a junior or apprentice level, and a senior or master level. Do not have too many levels in your pools! Remember the purpose is to identify critical resources whose absence would delay the start of project tasks, not document the Human Resources classification of all the resources in the organization!

Keep track of the resource pool definitions. Many times, identifying a resource pool also includes additional resources and equipment that would normally accompany the resource. As an example, a "Software Training Developer" is assumed to have access to a computer with the software loaded as well as the current software training materials. This assumption may not be correct for subcontractors, for example. Keeping this "data dictionary" of the resource pools and assumptions and keeping it updated, can save a LOT of time and rework during project execution as well as other project's planning activities.

There is often a question as to how many pools should there be. We've covered the people resource pools, but have not yet discussed facilities and equipment. Typically, critical equipment is considered to be large, expensive pieces of equipment, such as a water-jet cutter or 3-D modeler – resources that are used a lot and are difficult to schedule; a copy machine or scale would not be considered critical equipment. Including facilities and equipment, we typically find that a range of 30-50 resource categories are sufficient to model a portfolio of projects. We have implemented with less than 20, and achieved outstanding results.

Don't start by spending a lot of time identifying the resource pools; they will naturally emerge as you perform this step.

Resource Continuity (RC)

In some types of projects, the specific resource that performs an early task in one part of a project (one path) can save time and improve quality later if brought back to work in the same path. This is called resource continuity. The thinking is that another resource from the same skill pool would take too long to get up-to-speed for task/project knowledge and quality if it's not the same resource that worked earlier in the path. In these few cases, note within the project plan that this task has a need for resource continuity in the task notes, along with the resource skill type, if there's more than one resource skill level identified for the task. For example, the task note could read: "Resource Continuity desired – software engineer".

Identifying Critical Resource Skill Levels and Quantities per Task

6.1. Identify which portions of the network will most likely be performed by which resource pools (color-coding is often useful). Identify which subject matter experts (functional managers, resource managers, etc.) are the best to identify resource skills and quantities to tasks. The preference is to start at the beginning of the project and work left to right. Bring these subject matter experts in one at a time and orient them to project network before beginning Step 6.2.

6.2. Begin at the left side of the project network. Pick a path start task. For that task, determine what resources are critical for accomplishing the task; that is, the task cannot start without those resources being available – note that there may be more than one resource skill level.

Let's use "Update Software Training Materials" as an example. There is a pool called "Software Training Developer". Are any other resources required for the task? Making a list of the steps or activities required to meet the task completion criteria helps to identify any other critical resources. (This is extremely useful to put in the task notes for project execution; the resources, task manager, and resource manager(s) will all have a better idea of what this task is really supposed to accomplish!) The subject matter experts also often provide task notes that will be useful during task execution – not only to understand what must be accomplished in greater detail, but also to assist in providing estimates of time remaining to accomplish all those steps. For example, draft training materials must be put together for the software testing. Since the software training developer creates the draft training materials, it sounds like creating the draft training materials is a step in accomplishing this task – we would make that a task note. (Since the software training developer would hand off the draft training materials to him/herself, we consider this one task; if there was a handoff to another resource, it may be a reason to break this task into more than one task. See Figure 18-1.

Update Software Training Materials

Resource: Software Training Developer

Steps:
1. Understand New Software Release
2. Determine "What, Where, How" For Each Training Module
3. Determine And Create Exercise(s) For Each Training Module
4. Create Draft Training Materials (Overview, Planning, Execution For Leadership, Execution For Users, Task Updating)
5. Review Results Of Usability Verification Testing For Any Required Changes To Materials
6. Create Final Version Of Software Training Materials

Figure 18-1: Resourcing

Examining the steps on this task, step 5 is performed after the usability verification testing, which is another task, performed by another resource. This is a justifiable reason for breaking this task into more detail: there is another task, performed by another resource, embedded inside it. Fixing this situation looks like Figure 18-2.

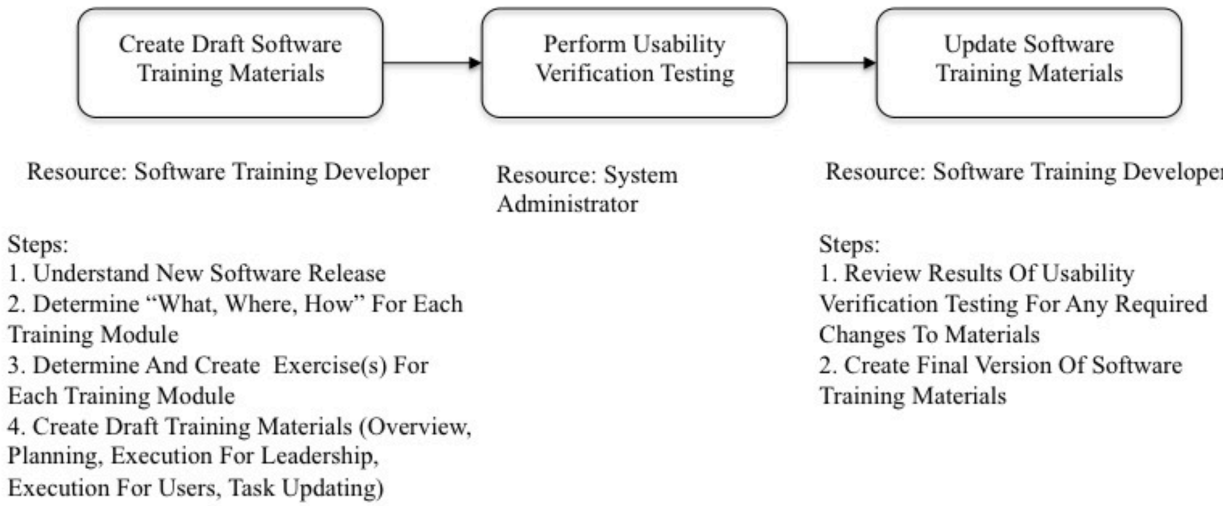


Figure 18-2: Breaking A Task Into More Than One Task
Justification: Resource Does Not Hand Off To Itself

During the initial planning, identify the lowest critical resource skill level capable of performing the job. The rationale is that later, before/during execution, if there is a need to further compress the schedule, we can identify the best places to put higher skilled resources as a way to do that schedule compression, rather than counting on scarcer resources unnecessarily. The philosophy is to plan initially with the lowest skill level capable of meeting the required task completion criteria and then add higher skilled resources only where and when necessary (see Step 9 – Duration reduction without compromise). Assign the largest quantity of resources of that skill set as practical.

6.3. After the resource skill level(s) and quantities are identified for the first task on a path, follow the same process for the successor tasks on the path, noting any resource continuity issues in the task notes.

6.4. Repeat for all pathways in the project.

Special Types of Tasks

Many times while identifying the correct resource skill levels and quantities for tasks, we find that there is no resource that is required for the entire time the task is going to be worked. Consider the situation when a long-lead part must be ordered. Actually doing the ordering requires a customer service clerk; however, looking at the task notes, we find that what is really meant by the task is ordering and receiving a specified long–lead item. In this case, we would actually have two tasks: "Ordering the __________ long-lead item" (fill in blank with the specific item) which is done by the customer service clerk, and “Wait or Delay for the _________ long-lead item”. This task is performed by what we would call a dummy or vendor resource (so called because it is not a resource we would necessarily track, but is not the same resource as the customer service clerk); in some software, there is no need to list a resource for the “Delay…” task. See Figure 18.3. There are other delays and/or wait tasks that exist in projects. We recommend modeling them similar to long-lead tasks, changing the task description appropriately.

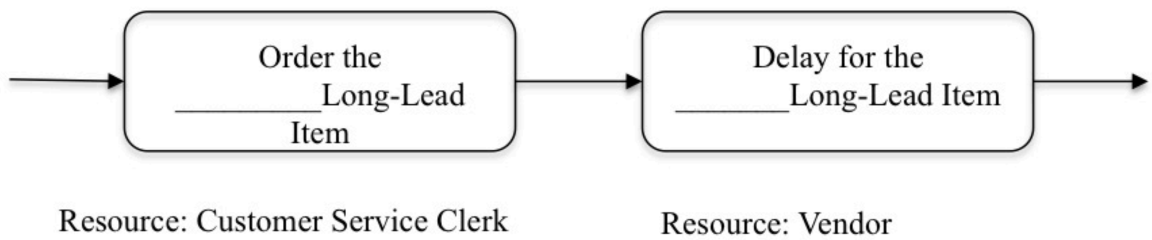


Figure 18-3: Modeling Ordering and Receiving _________ Long-Lead Time Item

In other cases, someone (external to your organization) has promised to provide you with an item on a specified date. We refer to this item as a “receivable” to the project. The receivable is a required input to a project task. Modeling this situation, especially if that external resource is not always reliable is also a modeling special case. The unreliability of the provider should not be measured within the task time itself; the task cannot start until the part arrives! Any project management software allows you to create a “start no earlier than (SNET)” task with that date and the amount of variability expected. By making that task precede the task that needs to use the part, this SNET task becomes a placeholder, modeling the expected arrival time. The predecessor, SNET task, is a path start. See Figure 18-4.

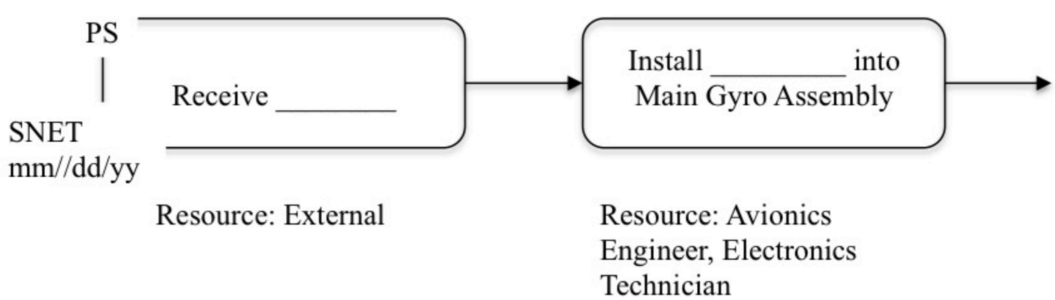


Figure 18-4: Modeling a “Received” Item

Conclusions

In most current paradigms, tasks are like hot potatoes. Resource managers assign tasks to resources at the planning stage. Every resource has multiple tasks to do, including non-project work such as operational responsibilities, ongoing improvement tasks, etc. In this new paradigm, instead, the resource manager plans to assign as many resources as practical to critical tasks, to drive projects faster to completion. The resource manager looks to have only one task assigned to a resource at a time during execution, and therefore considers, at this planning stage, as to how their resources can be freed from other responsibilities during the performance of a project task, so their time can be dedicated (not multitasked). The formal parts of step 6 are:

6. Determine resources (skill level and quantity) that could be assigned to perform the task.

> 6.1. Identify which portions of the network will most likely be performed by which resource pools (color-coding is often useful). Identify which subject matter experts (functional managers, resource managers, etc.) are the best to assign resource skills and quantities to tasks. The preference is to start at the beginning of the project and work left to right. Bring these subject matter experts in one at a time and orient them to project network before beginning Step 6.2.
>
> 6.2. Begin at the left side of the project network. Pick a path start task. For that task, determine what resources are critical for accomplishing the task; that is, the task cannot start without those resources being available – note that there may be more than one resource skill level.
>
> 6.3. After the resource skill level(s) and quantities are identified for the first task on a path, follow the same process for the successor tasks on the path, noting any resource continuity issues in the task notes.
>
> 6.4. Repeat for all pathways in the project.

Next post: Step 7: Scrutinize the network logic using subject matter and/or skill set experts (third risk avoidance).

Building Executable Project Plans – Part 9

Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 8 detailed Steps 1 - 6 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 19.

Step 7: Scrutinize the network logic using subject matter and/or skill set experts (third risk avoidance)

Until now, we have used a small team of very knowledgeable people to build the project network. We have used people with good overall knowledge of the company, of projects and of the network building approach. We have made many assumptions along the way, both with regard to the tasks that need to be accomplished to meet the stakeholder needs, and in terms of resources who can do the work. Step 7 is intended to find missing pieces of the network, by having experts in specific sections review their relevant parts of the network for deficiencies or missing elements. We want to ensure that we've not missed any tasks, not misstated any of the necessary interdependencies, and appropriately identified the critical resource levels and quantities. In other words, we want to make sure that we have the "work" and "critical workers" of the project correct before we start gathering task estimates.

Identifying and Using Expert Scrutiny

By now it's likely that "themes" have developed in the network. There may be a section that involves the start-up of the project, followed by multiple paths of work branching off from the start (as an example, hardware, software, documentation, integrated testing); likely there is some final section of the project (which may include system test, packaging, shipping to distribution centers, etc.). These themes help indicate whose expertise is needed to do the scrutiny and when.

Consider having smaller groups of experts in for scrutiny of their specific theme area. Recognize there may be overlapping areas of expertise and plan the scrutiny accordingly. Warning! Do not bring in all the experts at the same time and plan to go over the entire network diagram. Bring the appropriate experts in <u>only to review their specific sections</u>! Otherwise, you will waste a lot of the experts' time listening to reviews of areas in which they have no interest nor expertise to share. This will cause them to resist future efforts to enlist their help.

Ensure you provide a broad overview of the project to the experts before going to scrutiny of a particular section. As an example, "This project is to accomplish the upgrade of Absolutely Essential software at GENEXCO." Give them enough depth on the stakeholder needs and deliverables so that they can scrutinize their section within the context of the needed results.

Referencing the clean network diagram plot before them, and starting from the left, show them the different sections before going to their section under scrutiny. This gives them an understanding of what work is done before getting to their part(s) of the project. It's useful to have a computer to display the task notes (and to add any additional notes that the experts recommend).

The experts should have experience in their theme area as well as knowledge about the skill levels of the organization. The experts may be the Resource Managers for the thematic areas or senior experienced resources. It's not unusual for management team members with previous expertise in the thematic area to be called as expert scrutinizers.

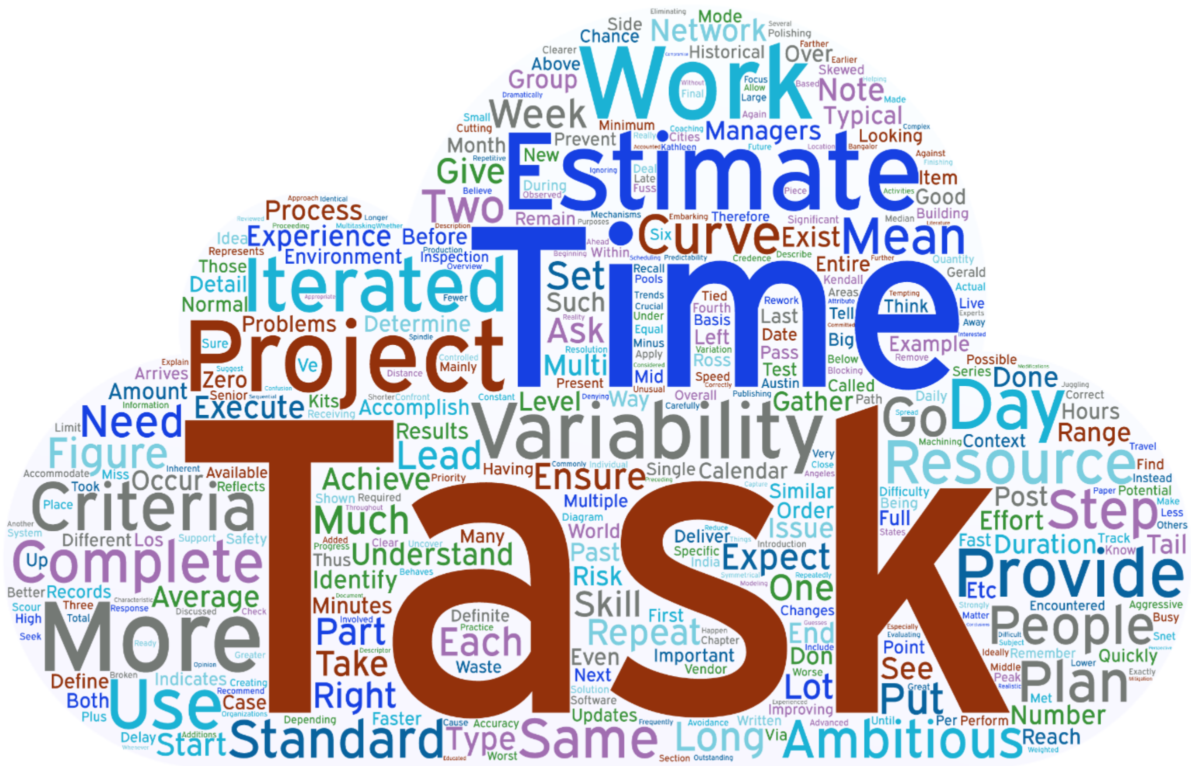
Make any recommended changes at the end of each scrutiny session and reprint a fresh network diagram before starting another scrutiny session. Remember that any additional tasks are subject to the Step 5 scrutiny – make sure that you are not adding tasks that are not required to meet the scope of the project.

Conclusions

Expert scrutiny gives the team constructing the project network a chance to have the most knowledgeable, experienced people in the organization review the tasks defined within their subject area. The project plan is broken down into themes according to subject matter categories. Subject matter experts are brought in, separately, for each theme area. With an overview of the project objectives, scope and sections of work that precede their subject area, they can quickly review the tasks defined and advise if changes are necessary. The result of Step 7 is an almost-finished project network. The work definition should not change extensively in the following three steps.

Next post: Step 8: Define time estimates, with range of variability (fourth risk avoidance)

Building Executable Project Plans – Part 10



Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 9 detailed Steps 1 - 7 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 20.

Step 8: Define time estimates, with range of variability (fourth risk avoidance)

In the new world you are embarking into, do not put a lot of credence on individual task time estimates. The entire solution provides a multi-project environment where work will be completed faster than ever before. Remember that this effort is not mainly about creating good project networks. We are:

- Dramatically cutting project work in progress, thus giving senior management, support groups and resource managers a much faster response time to resource issues and better coaching of resources.
- Eliminating multitasking of project work
- Providing mechanisms such as a fast track issue resolution process, daily task updates and full kits, to remove blocking issues quickly and prevent rework.
- Much more carefully defining the project plan so that work that is needed before a task is started is more likely to be available, with a much clearer definition of the work to be done.

- Changing to a single priority system, preventing the constant juggling of resources.

You cannot expect people who are providing the time estimates to understand or even believe that all of the above changes will be in place when their tasks execute. Therefore, this process needs to be accomplished in a way that does not confront the people providing the estimates, but at the same time arrives at much more aggressive task estimates than typical of the past.

Approach to Gathering Time Estimates for Each Task

In Step 8, it's important to have two time estimates for each task: an aggressive, achievable time (if not too much difficulty is encountered accomplishing the task completion criteria) and an understanding of the variability one might experience if there is more difficulty than expected achieving the task completion criteria.

Variability exists in almost every task of a project. We don't expect the same type of variability in every project or even every task. Some projects have more variability in the beginning, some at the end, others in the middle, and some experience different levels of variability throughout the entire project. Having an understanding of the potential variability in a task gives a context for understanding task updates during execution; i.e., is the task proceeding as expected or are problems occurring which need more time to achieve the task completion criteria? This is crucial information for project and resource managers to know.

In project organizations, the same "types" of tasks are performed repeatedly by similar resource pools. It is tempting to scour the historical records of similar tasks to identify actual task durations and use those (or averages or weighted averages, etc.) for task estimates. We strongly recommend <u>against</u> that practice. Having the historical records are important for evaluating trends to uncover potential improvement areas; however, using those historical records for task times is not a good idea.

Most people, when asked to describe variability, think about a "normal" (symmetrical) curve, meaning an equal chance of finishing the task below the average (mean) or above the average, with the same level of variation possible in either case. In this type of curve, the mid-point (called the median) of the curve is at the mean, and the most frequently occurring point (called the mode) on the curve is also the mean. Such a curve is most commonly observed in repetitive, controlled environments, such as production, where the same task is repeated many times. See Figure 20-1.

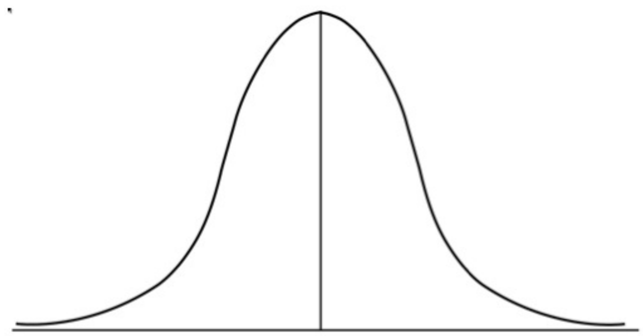


Figure 20-1: Normal Distribution

However, when thinking about variability in project task times, the normal curve does not apply. A project task time typically behaves more like estimating how long it will take to get to work in Los Angeles or worse, Bangalor, India. People who live in big, busy cities may find that, on average, it takes them 45 minutes to get to work. They can recall times when they arrived in just under 30 minutes (minus 15 minutes from the average). But is the worst case plus 15 minutes from the average? No! These people can tell you about the time it took them three hours to travel exactly the same distance.

A project task has a minimum amount of time that it takes to accomplish the steps required to achieve the task completion criteria – that means there is a definite lower limit to the curve, which is greater than zero. Depending on the amount of variability that is estimated to accomplish the task completion criteria – the inherent task variability only – the right tail of the curve can go on for a long time. This results in project tasks with curves that are skewed to the right, with the mean farther to the right than the mode (peak). A significant characteristic of these skewed variabilities is a long right tail. See Figure 20-2.

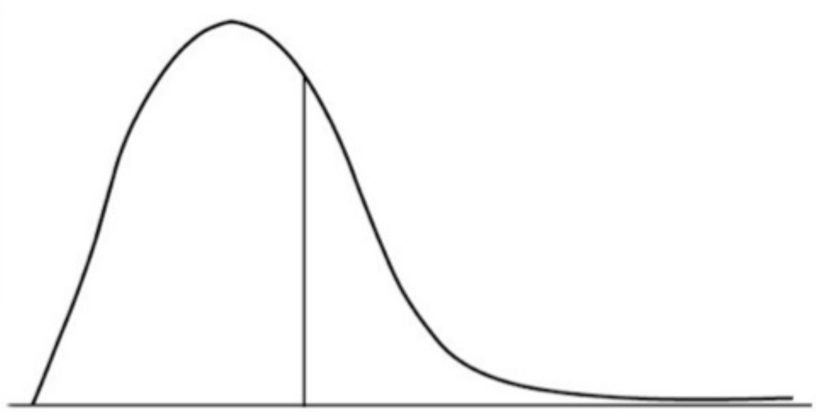


Figure 20-2: Skewed Distribution

What we need for the time estimate is an understanding of both the more ambitious time and the range of variability expected for the task. In gathering the task times from people with experience both in the amount and type of work required to reach the task completion criteria and also the skills of the resource(s) identified, start at the left of the project network for that skill set (ideally, start with a resource skill set that is a path start).

8.1. Work with only the experienced people for that resource skill set or sets and that type of work for a specific section of the project (again we are not looking for a large group tied-up for a significant amount of time; multiple small groups should be used so as to not waste people's time).

8.2. Ensure you have presented an overview of the project work so they have a context.

8.3. Present the task description and the resource skills and quantities that are identified for the task.

8.4. Go over all the task notes for the task being estimated to ensure there is an understanding of the work involved and the task completion criteria; if they suggest additions or modifications – especially to the steps or activities to ensure the resources are clear what work to perform, be sure to include them in the task notes.

8.5. Explain that we are looking for the task variability so will be asking for two task estimates.

8.5.1. Ask first for the time that is typically estimated for this task – this is considered the standard or right side of the variability range – remember people are not likely to give a task estimate that they have a high chance to miss.

8.5.2. Next ask for an ambitious time -- how long it would take to accomplish the task completion criteria if no unusual problems occur while accomplishing the task completion criteria. This time gives us the left side of our range for planning and scheduling purposes. We are not looking for the minimum time estimate! This time does have safety and variability in it and is expected to be to the right of the mean. This is a time with *less* safety in it because fewer problems (variability) are expected.

8.5.3. The two task times are written as (ambitious, standard) with the time descriptor (d for days or h for hours) added. For example, if the standard time is 8 days and the ambitious time is 6 days, it would be written as (6d, 8d).

8.6. Repeat for all tasks with the appropriate subject matter experts.

Calendars

When gathering the task time estimates, we are interested in the working calendar only. If the resources work 8 hours a day, 5 days a week, the example (6d, 8d) represents a week and a day for the ambitious time, and a week and 3 days for the standard time.

When multiple resources work on a task and work different calendars, you must accommodate this via resource calendars. This does make it more complex to determine overall estimated task duration times; it is not only the resource(s) that are estimated to need the most time, it is also how the resource(s) spread their task work over their available calendars.

Long Lead (Delay) Variability

When ordering long-lead items, the vendor provides the lead-time estimate. Use that estimate as the ambitious time. Use your experience with the accuracy of the estimate to determine the standard time. For example, if the vendor states a lead time of one month (typically 20 working days) and your experience is that they deliver in that time, use (20d, 20d) as the lead-time task estimate. However, if they estimate one month, and your experience with them is that they are more likely to deliver within six weeks (30 days), use (20d, 30d) for the lead-time task estimate.

Receiving Task Variability

Receiving tasks have a SNET (start no earlier than) date attribute which represents the date the provider has committed to providing the item to you (at your location). There is a chance they will deliver on time and the ambitious time reflects that – use 0.1 h (typically that is as close to zero time as software will allow*). For the standard time, use your experience with the provider. If they are typically two weeks late, the task would be estimated at (0.1h, 10d).

* It's important to understand your software to ensure modeling this correctly!

Iteration Variability

Iteration variability occurs when a series of tasks have to be repeated in order to reach the last task's completion criteria. This can happen when a test is at the end of group of tasks; if the test is not passed, the work of preceding tasks must be repeated.

Not every project has iteration variability. It must be accommodated and accounted for when it is part of the project environment. Note: Don't go looking for iteration variability that does not exist. But, ignoring or denying that it exists on paper (when it is in the reality of the projects' environment) does not cause it to go away!

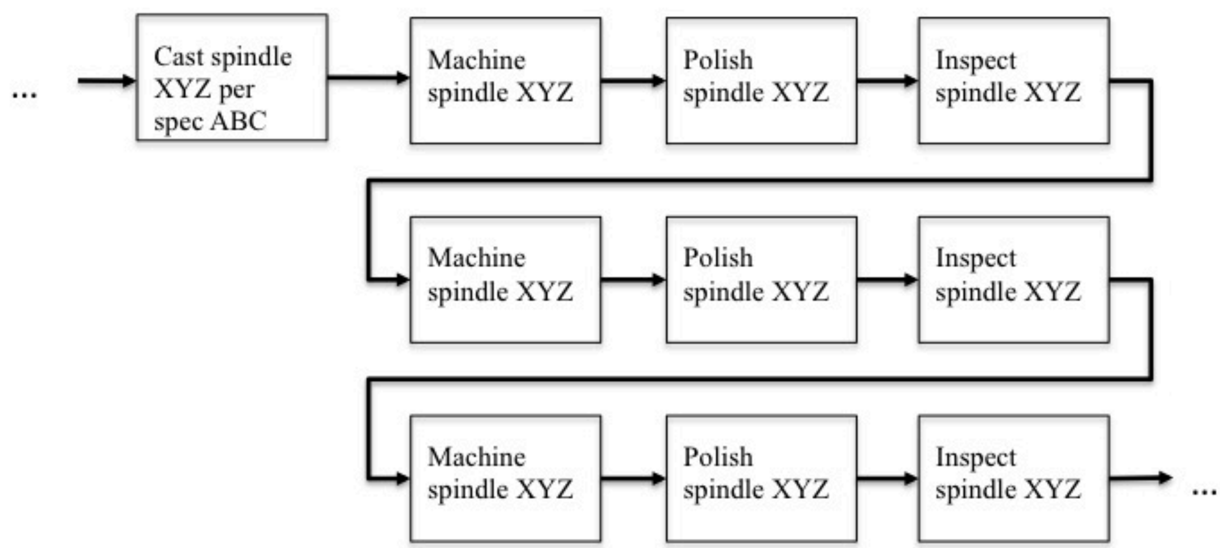


Figure 20-3: Iteration Variability (Notional)

In Figure 20-3, if the spindle does not pass inspection the first time, the machining, polishing, and inspection must be repeated. There can be variability in the number of times the tasks must be repeated before the task completion criteria are met. Therefore, the process for identifying the quantity of iterations is similar to the process of identifying task times:

1. Identify the tasks likely to be iterated (in Figure 20-3, they are the machining, polishing, and inspecting tasks).

2. Determine the standard number of iterations (in Figure 20-3 the standard is 3).

3. Determine the ambitious number of iterations (in Figure 20-3 the ambitious number of iterations is 1).

4. Determine whether the task times for iterations 2 and 3 are the same as for iteration 1 or not. Depending on the work being done, the task times can remain the same, be shorter, or be longer. Ensure you understand and document the estimated task times for EACH iteration in the task notes.

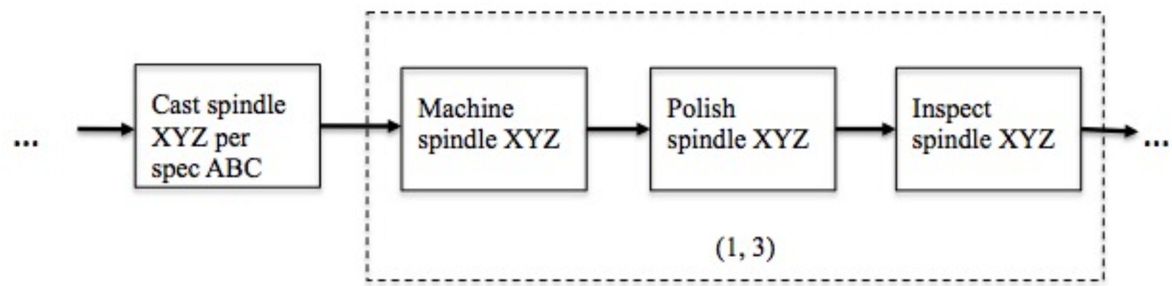


Figure 20-4: Identifying Iteration Variability

Figure 20-4 indicates how the network diagram indicates ambitious and standard iterations when the task times are estimated to be identical for each iteration. The tasks put into the network are the ambitious number of iterations. For example, if the ambitious and standard for our example were (2, 3) the tasks would be shown in the network as in Figure 20-5.

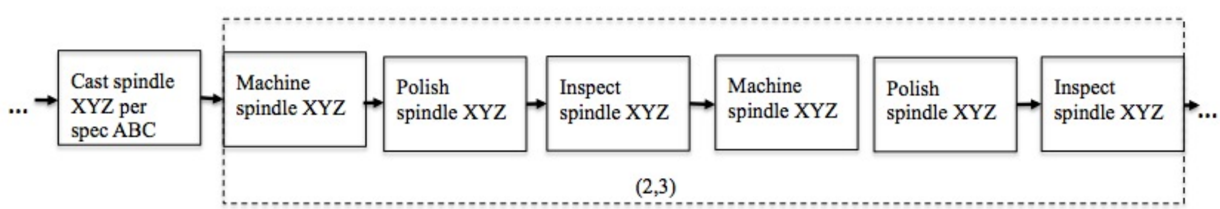


Figure 20-5: Iteration Variability

When task times are different for each iteration, the network diagram would be shown as in Figure 20-3 with the task times indicated per task.

Important note: There is often confusion about what iteration variability means. It is NOT the number of times work must be repeated within a single task to reach the task completion criteria; that would be reflected in the task's ambitious and standard times. Iteration variability is specifically when multiple sequential tasks must be repeated in order to achieve the completion criteria of the last task in the series.

Task Time Estimate Final Check

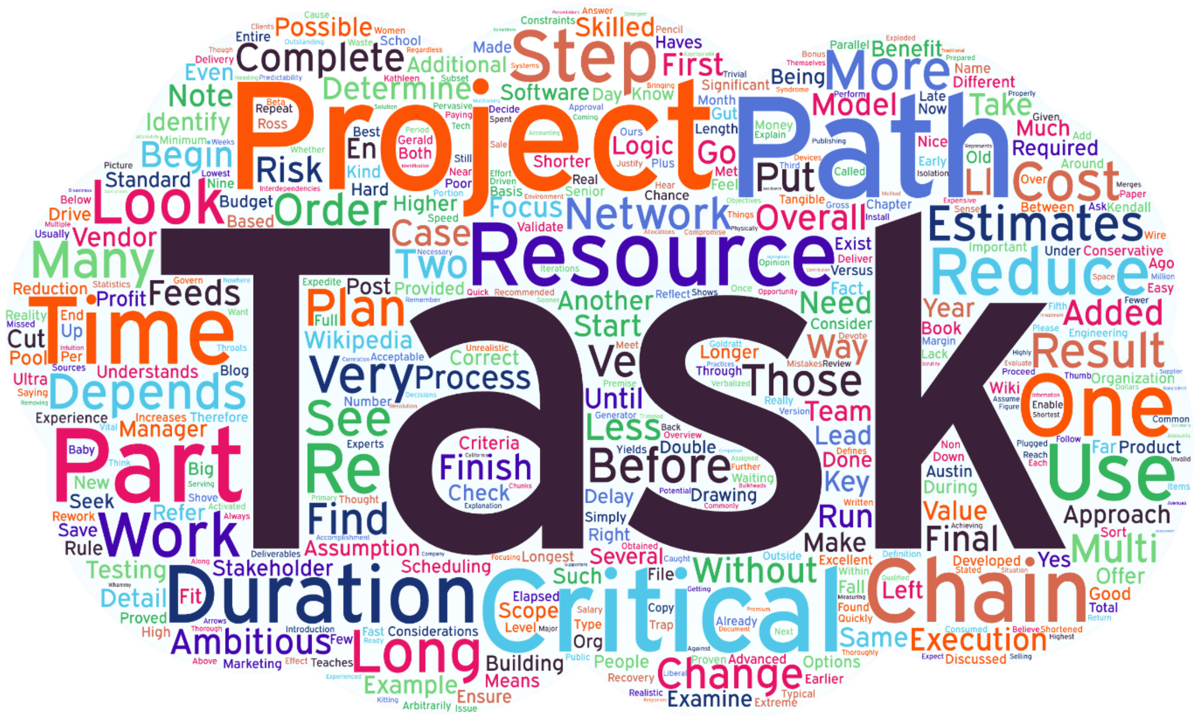
Step 8 is not complete until you've reviewed all of the task times. Whenever there is an ambitious estimate of more than two weeks, that task should be broken into multiple tasks. Why? It's another perspective on risk mitigation, this time in planning for execution. During execution, task updates will be done in the context of remaining duration. It is very difficult to really provide a realistic estimate of remaining duration when a task is planned to be more than 10 days long.

Conclusions

In a lot of the project management literature and in many project management improvement efforts, a great deal of fuss is made over task time estimates. The world would have been much further ahead if the same effort had instead been put into helping people get tasks done more quickly. Estimates are just that – educated guesses, based on past experience, of how long it will take us to do a piece of work several weeks or months in the future. In our opinion, it is a waste of time to focus on accuracy of estimates as a means to improving multi-project results. Instead, in Step 8, we quickly gather two estimates for each task to capture the variability of task time possibilities – one estimate is if things go as normal (a standard estimate) and the other estimate is if almost no problems and delays are encountered (an ambitious estimate). With this range of variability, we are ready to put these estimates into the plan and see what the total project duration is likely to be.

Next post: Step 9: Seek ways to reduce overall project duration without compromise.

Building Executable Project Plans – Part 11



Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 10 detailed Steps 1 - 8 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 21.

Step 9: Seek ways to reduce overall project duration without compromise

The project planning team now thoroughly understands the total network logic (tasks, interdependencies, resources, and time estimates that enable the accomplishment of the project scope, objectives, and tangible deliverables). It's time to determine the duration of the project to see where to focus to reduce overall project lead time. For most projects, when the time estimates that you've obtained in Step 8 are plugged into the network, the software provides a duration that is longer (often much longer) than the executives and other stakeholders find acceptable. Therefore, the premise of this blog is that the network building team must be prepared to go through some iterations to reduce project duration, without having to shove unrealistic estimates down people's throats and without adding additional risk to the project.

Finding the Tasks That Will Govern Duration

There are two commonly used approaches for determining how long a project is likely to take, in elapsed time. The Critical Path process was developed in the late 1950s, and defines the longest chain of task dependencies within a project. There are several assumptions about

Critical Path, based on statistics, that proved useful until around 20 years ago. There are many good books written about Critical Path, and Wikipedia,[1] among other sources, has a very good explanation of the approach.

Beginning in the 1990s, as the number of projects activated in organizations exploded, many tasks found themselves waiting not on another task, but on a resource that was already working on another part of the same project. This syndrome of the project duration depending not just on task dependencies, but also on resource dependencies, was verbalized in a book called *Critical Chain* (Goldratt). Wikipedia also provides an excellent overview of this approach[2].

In our opinion, Critical Chain is a more conservative and realistic approach, since resource constraints in the multi-project environment are pervasive. Therefore, we will devote a portion of this chapter to explain it further. From our experience, this focus yields excellent results in reducing project duration. The project's Critical Chain is the longest chain of dependent tasks through the project, where the dependency could be based on either:

- One task depending on another task finishing before it can start because it needs the result of that task in order to start or
- One task depending on a resource that must finish another task on another path of the same project

Software makes identifying the Critical Chain an easy task. Regardless of whether you decide to use Critical Chain or Critical Path, you will have the best chance of reducing a project's elapsed time duration by focusing on that subset of all the project tasks that most likely will drive it.

Overall Project Duration Reduction

When looking to reduce overall project duration, it is necessary to do it without adding any risk to the project. That means we cannot arbitrarily begin reducing either ambitious or standard times. The steps below offer a useful process for reducing duration, without increasing the risk to the task completion criteria. There's an old saying that changing the estimates on paper does not change reality. Do not fall into that trap. Remember during this process that it is possible to go too far – additional resources and more experienced resources are key recovery options during execution; the more you've used those to reduce durations, the fewer options you'll have if/when recovery is required.

9.1. Focus on the Critical Chain or Critical Path tasks first, since they are the primary determiners of project length. Since we do not know which of these approaches you are using, we will simply refer to these tasks as "Critical". Examine the critical tasks looking first at the longest ambitious times. Note: Being able to sort the tasks by ambitious times (highest to lowest) is very useful.

9.1.1 Take a hard look at the ambitious and standard times. Given the team's thorough understanding of the project, do they have a gut feel that any of the task estimates are over-stated? This can result from a lack of understanding of the tasks when they are estimated. Often things are caught when looking at the big picture which are missed when looking at tasks or paths in isolation. Perform a detailed review of those tasks and pencil in the new estimates.

In the software, make the changes and save a copy of the project file under a different name (e.g., project_name_reduction_version 1). Document the changes and validate the new assumptions with the appropriate experts. Re-run the software to identify the Critical Chain or Critical Path; it may be different!

9.1.2 Examine the resulting Critical Chain or Critical Path tasks. Can any significant ambitious and/or standard task estimates be shortened by changing from the minimum skilled resource to a higher skilled resource? Does such a resource exist? Looking at the potential added cost of a higher skilled resource versus the benefit to the organization of bringing in the project earlier, does the benefit justify the cost (if any)? Note that in many cases, there is no added real money cost, when the higher skilled resource already works for the organization. I.e., you are not paying them a higher salary to work on this project! However, cost allocations by project accounting systems can drive some extremely poor decisions.

9.1.3 Would the ambitious and/or standard task estimates be shortened if additional resources were added to the task? Note that this does not always reduce task estimates – nine women cannot have a baby in one month! If additional resources do make a difference, do those resources exist? See the discussion above on project cost versus benefit considerations.

9.1.4 Repeat the process on all significant ambitious Critical Chain or Critical Path task estimates.

9.1.5 When finished, re-run the process of identifying the Critical Chain or Critical Path. Check to see if steps 9.1.1 – 9.1.5 should be repeated.

9.2. Examine long feeding paths. All tasks that are not on the Critical Chain or Critical Path are considered to be on a feeding path. Every project has only one Critical Chain or Critical Path; it has multiple feeding paths. Any path that feeds into or merges with the Critical Chain or Critical Path is considered a feeding path. For very long (a rule of thumb is 2/3 the length of the Critical Chain/Critical Path) feeding paths, follow steps 9.1.1 – 9.1.3 on these paths.

9.3. When you are finished reducing durations, re-run identifying the Critical Chain or Critical Path a final time.

9.4. As a final step, ensure the experts you've used for resourcing (Step 6) and time estimates (Step 8) validate the changes made. A double-check by people serving as senior resource manager(s) and senior project manager is also recommended.

Important Note: Scheduling is not complete. Identifying the Critical Chain or Critical Path is part of scheduling but is not all of the scheduling process. Please refer to *Advanced Multi-Project Management* (Kendall & Austin) for more information on project scheduling.

How Many Resources On A Task?

There is one school of thought that teaches all in a resource pool should be put on a task when planning and scheduling in order to complete the task in the shortest possible time, while another school of thought teaches to put the minimum number of resources required to reach a task's completion criteria on a task in order to save project budget costs. Which is correct? We believe there is no hard and fast rule; it depends! What is important is to know what it depends on, so that you can properly evaluate the situation.

For example, assume that a project requires completion of 500 engineering drawings. If that work represents one or several long critical tasks, it will make sense to put as many qualified resources on it as possible in order to reduce time. However, if the project task is to wire and install measuring devices between bulkheads 234 and 246, there may be only two resources that can physically fit in the space. Yes, these are two extreme examples!

Consider also that resources will not be multi-tasking while working a project task: once a resource begins work on a task, he/she/it will work on that task until task completion criteria are met. However, resources are assigned to tasks, not entire projects. Plus, by not multi-tasking, resources are not needed as long for the same task work. These recommended execution practices are examples of key considerations when determining how many of a resource pool should be planned for a task to reduce overall project duration.

For those projects with budget constraints, experience shows (both ours and also coming from public presentations by organizations using this kind of approach who also have to meet budgets) that the shorter the project duration (driven by not multitasking and quick issue identification and resolution and full kitting projects), the less rework, the less resource time consumed and the less waste; i.e., there is both a correlation and cause-effect between shorter duration and less money spent.

Other Considerations for Reducing Task Duration

1. One of the most common mistakes in building networks is the assumption that ALL task dependencies are, for the most part, correct. Is it possible that nowhere near all of task A must be finished before even starting task B, even though we modeled it as a 100% dependency between the two tasks in the network? In almost all cases, we find a few such cases where the model was ultra conservative, and in fact most of task A can be done in parallel with task B. When you change these assumptions, by removing these dependencies, the typical result is a shorter duration.
2. Where significant amounts of time are used up by outside dependencies (e.g., Vendors), determine what the value is of expediting delivery for critical items. For example, a project in Bangladesh was delayed for months waiting for high end generators from an outside supplier. The value of the project was several million dollars per year and was very tangible. The return on investment was less than one year. Each generator was selling around $150,000. If you figure the traditional way that a vendor values the sale, they usually expect a product gross profit contribution of 40-50%. That means that about $75,000 is their profit margin. If you were to offer them a $25,000 bonus for delivering early, that increases their profit margin by a third. This is one way to expedite with vendors. If you simply ask them if they can possibly deliver earlier, the answer is automatically "no". But offer them a significant premium, and the answer can change

very quickly. In this case, the added cost was trivial in comparison to the value of getting early delivery from this vendor.

3. Re-examine the tasks against the scope and stakeholder needs. In many cases, we find liberal assumptions about tasks being required. When checking back with the stakeholder, we often hear responses such as, "Yes, that would be nice to have but what I'm really after is……". You are NOT cutting scope if these kinds of tasks are trimmed before the project is even started. The key stakeholders are often the strongest supporters, if it means that they can get the most important benefits much sooner without the nice to haves.
4. Re-example the major chunks of project logic. Sometimes, we find mistakes in how the project was modeled. For example, in one product development effort with a California high tech company, there was a 15-day testing period during which no other development work could be done. 95% of the time, the result of the testing is that the product solution is proven to work, and is ready for beta testing with clients. However, being ultra-conservative, they modeled the testing under a marketing resource heading, because they did not want marketing to proceed until the product was proven. In fact, there were several time-consuming and critical marketing tasks that could proceed in parallel with the testing. When this was modeled to reflect those changes, 15 days (3 weeks!) were cut from the project duration.
5. Double check the arrows along the Critical Chain/Critical Path. Is the task to the left really required to be complete before the task to the right can begin? In reality most "must have" dependencies are one of two types:
    a. The task to the left *absolutely* has to be complete before the task on the right can begin; this type is the most common.
    b. The task to the left does not HAVE to be completed before the task on the right can begin, but completing it first reduces the chances for rework or delay. Think of an expensive long-lead part. The engineering drawing for that part does not have to have its final approval before ordering the part, but if the part is ordered before final engineering drawing approval and a change is made to the drawing after it's ordered, the project can experience a very expensive delay (a double whammy!)

Conclusions

In seeking ways to reduce project duration, it is vital to not arbitrarily cut people's time estimates. This usually proves disastrous in execution. The first focus should be on those tasks which are critical to the projects (i.e., Critical Chain or Critical Path tasks), since those tasks, more than any other, are likely to determine how long the entire project will take. The typical options to examine are time estimates that do not reflect the intuition of the team about how long they should take, the opportunity to add more resources to a task to get it done more quickly, and the opportunity to put more highly skilled resources on some tasks to reduce duration.

When finished scrutinizing the critical tasks, look at very long non-critical paths, with the same scrutiny.

If the project duration is still much longer than acceptable to stakeholders, then look for invalid assumptions about the network logic and dependencies, and look at long vendor lead times as avenues for reduction.

References

1. See the Wikipedia definition of Critical Path at http://en.wikipedia.org/wiki/Critical_path_method
2. See the Wikipedia definition of Critical Chain at http://en.wikipedia.org/wiki/Critical_chain_project_management

Next post: Step 10: Complete a final, overall project assessment (fifth risk avoidance)

Building Executable Project Plans – Part 12

Part 1 provided an introduction to projects and the need to plan a project. Part 2 discussed how to ensure the correct level of detail in a project network. Parts 3 - 11 detailed Steps 1 - 9 of the 10 steps to building an executable project plan. Basis for this post: *Advanced Multi-Project Management: Achieving Outstanding Speed and Results with Predictability*, by Gerald I. Kendall and Kathleen M Austin, J. Ross Publishing, 2012, Chapter 22.

Step 10: Complete a final, overall project risk assessment (fifth risk avoidance)

You now have the final perspective of all work required to accomplish the project's scope goals, objectives and deliverables, with task notes detailing activities within the tasks, significant assumptions about each task, and specific task completion criteria for each task. You have identified the risks to each task individually through the task notes and task time variability estimates. Durations have been adjusted / corrected where appropriate without compromising on completion criteria, budget, or timeline.

Given the overall understanding of the project stakeholder needs, the goals the project is intended to meet and the work outlined in the project plan, the project team is now prepared to take one final step. Are there other significant risks posed by this project, which would endanger meeting its goals, in spite of all of the preceding work in building a robust plan?

Holistic Risk Mitigation

It is time to look at the project as a whole in terms of risk mitigation. The final project risk mitigation should be done with the original planning team, sponsor(s), key stakeholders, and experts who have provided resource skills and quantities, expert scrutiny, and task time estimates.

This review should go over, in detail, all of the project details and open discussions as to missed risk mitigation. If any significant risks at the task level have been missed, that information should be added appropriately and documented in the task notes.

There are also risks that can occur during the project, but cannot be assigned to a specific task or series of tasks. Examples:

- Some series of tasks are performed by external resources who do not always reliably deliver. You do not know in advance which external resources this will happen with, but experience has shown that in projects like this, it's likely to happen at least twice. When it does happen, delays of at least 10 days occur. Make a note that a project-level risk adjustment (upward or longer in time) needs to be considered.
- Some of the equipment being used on the project has significant scheduled maintenance down time every 500 hours. That scheduled maintenance will happen at some time during the project, but there is no way to know when or the number of times it will occur. Document the number of times and the estimated down time as another project-level risk adjustment (upward or longer in time) that must be considered.

If there is iteration variability present in the project, look at the number of times it is estimated to occur. Typically, if there are one or two opportunities for iteration variability, no project level adjustment needs to be made; however, if there are more than two opportunities for iterations, consider whether the project is being "over-protected". If so, document and note that a project-level risk adjustment (downward or shorter in time) must be considered.

Organizations have typically experienced so much project failure and underachievement, that they are prone to overlook one of the biggest generic risks of a project – the risk of success beyond expectations. In the '90s, when AOL launched their internet service with massive advertising, they lost tens of thousands of customers almost immediately after launch. These customers tried to dial up the free AOL telephone numbers to connect, and experienced busy signals for hours on end. (Yes, we understand we may be describing something that sounds like horse drawn milk trucks and wagon trains to the younger generation!). Similarly, we've seen many cases of product launches where the stock needed to satisfy initial customer requests was grossly underestimated, and the lead time to manufacture more was long. Customers waited months, by which time competitors had caught up and offered their own products. In these situations, it's not just that the organization lost some sales – they made their customers so mad that they lost customers for life.

Another type of risk is what some insurance companies labeled the "front page" risk. This is the risk that their project results in customer complaints that are so severe, that the story ends up as a feature on the news.

The final conclusion of the group should be that all significant task and project-level risks have been addressed: the network as planned is sufficient (but not over-sufficient) to deliver the full scope of the project, at or below the budget, on or before the due date.

Conclusions

The final risk mitigation step is intended as a last, holistic look at the project, to mitigate or prevent any other previously unidentified risks from being realized. With the full team of stakeholder, network builders, and other key players present, the project can be examined to determine if other tasks are needed for final risk avoidance. Most often, the outcome of this step is to proceed with the plan as is or with very slight modifications. However, in the rare case where a major risk requires rework of the entire plan, it is much better to find out before

Final Thoughts on Building Executable Project Plans

This concludes the 12-part series on building executable project plans. Please provide any feedback, comments, questions, concerns, or requests. Did you find this useful? Did it make a difference in how you build networks? Thanks for reading!

