PROJECT SCHEDULE MANAGEMENT

One of the most difficult constraint to handle in every project is **time**, thus, making the management of time one of the most challenging task as a project manager. The reason for this might be because time is the least flexible among the other constraints. Whether we like it or not, time passes by as it consistently and constantly does.

Let's take for example a software development project that outsourced human resources. Difference in individual work styles of the project team members is to be expected by default but adding to this is the timezone difference, the cultural difference, and varying perspectives on schedules.

Time is a project constraint applicable to all kinds of project whether it has a flexible or strict due date. By managing the project schedule, we aim to ensure the timely completion of the project.

Project Schedule Management is a group of processes focused on making sure the project delivers the product on time and handling other constraints at the same time such as scope and resources.

It was previously called "time management" but I'm not sure why it was changed to 'schedule'. I believe, though, that technically no one can manage and manipulate time itself. What we can do is to ensure good use of time to complete a quality proejct. We can do this by scheduling and managing it well.

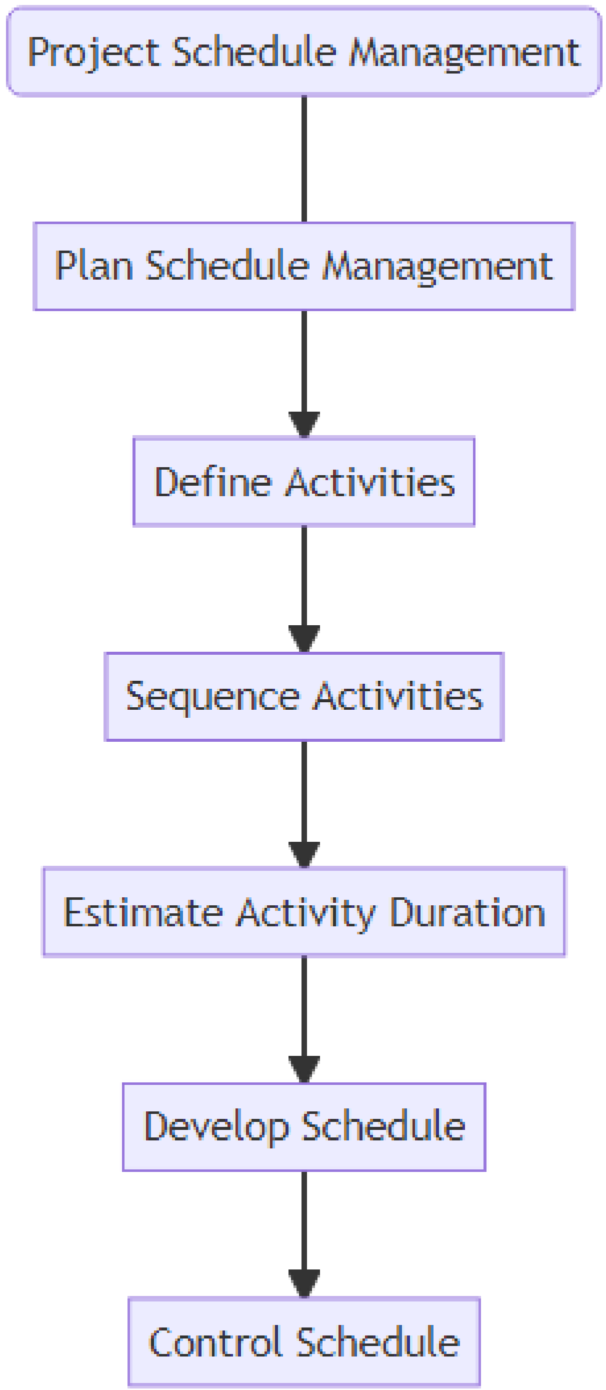
Project Schedule Management
Plan Schedule Management
Define Activities
Sequence Activities
Estimate Activity Duration
Develop Schedule
Control Schedule

Figure 1. An Overview of the Project Scope Management Processes

For projects with smaller scope, some of the processes, namely, defining activities, sequencing, estimating duration, and developing the schedule are considered as one big process and so they are executed immediately one after the other.

1. Plan Schedule Management

Like any other project management knowledge area, we start establishing guidelines, policies, and procedures in coming up with the project schedule and on how to monitor and control it.

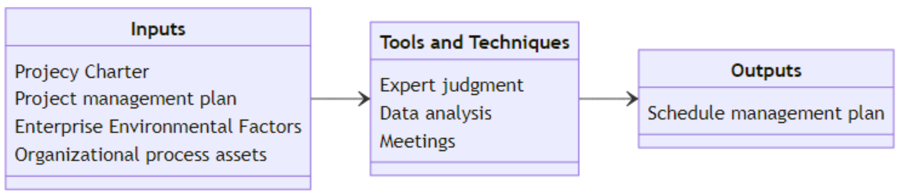

Figure 2. The process of planning Schedule Management

The planning phase provides direction for all stakeholders most especially that especially for the projects that has time as the primary constraint.

2. Define Activities

Recall that one of the main processes in scope management is the creation of the Work Breakdown Structure (WBS). In WBS, we decompose major work deliverables into smaller components of the project such as work packages.

These work packages will be then decomposed further into specific activities that can now be assigned to human resources and can be allocated with the necessary resources.

We try to identify the specific actions that must be done to finish the project within the given time.

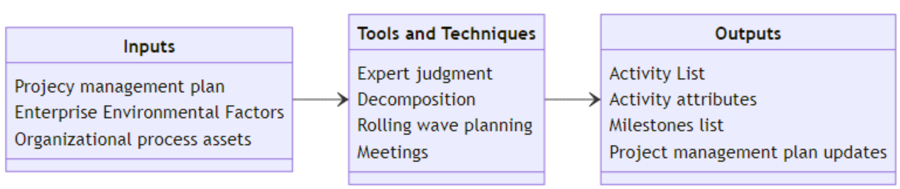

Figure 3. The process of defning activities

Activity list is different from the WBS, aside from the fact that WBS is our scope baseline while activity list is our basis in developing the schedule. The Activity list is the actionable version of the WBS which means the list contains specific tasks and to dos. WBS, on the other hand, may only contain the deliverables and components of the project. However, you may decide to have a uniformity of the two, meaning, the last level of the WBS can serve as your activity list provided that the WBS are elaborated enough. In fact, the WBS and activity list can be created sequentially or concurrently.

The decomposed work packaged or the schedule activities will then be our basis in the estimation and development of our project schedule model.

Activty ID	Activity Name	Estimated Duration (days)	Preceding Activities
A	Create project charter	10 days	
B	Develop preliminary scope statement	5 days	A
C	Create WBS and WBS dictionary	4 days	A, B

Table 1. Example of an Activity list with attributes

3. Sequence Activities

Now that we have identified the specific activities needed to be accomplished, the next step is to determine which activities should be done first and which should be done later. This means that we have to put the activities in order by determining the sequential and logical relationships between them.

In SCRUM, when the product owner has already ranked the product backlog list by priority, the predetermined dependencies would expedite the sprint planning process especially in adding tasks to the kanban.

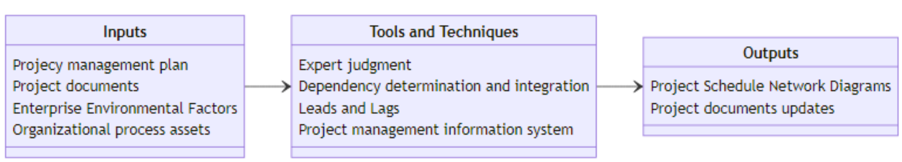

Figure 4. The process of sequencing activities

To visualize the sequential and logical relationship between activities, it is a good practice to use a project schedule network diagram. This way, stakeholders understand the relationships between activities better instead of a tabular format.

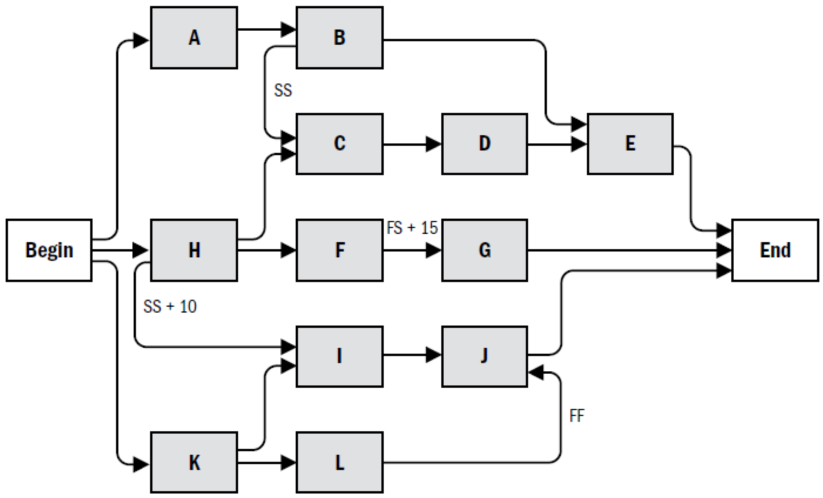

Figure 5. Example of a project schedule network diagram

The arrows pointing from one activity to another indicate the dependencies of one activity to the other. Another technique used in here is the Precedence Diagramming Method (PDM), also known as activity on node (AON), which links activities graphically indicating the type of relationship they have. The PDM relationship types pertaining to the logical relationship between activities are the following:

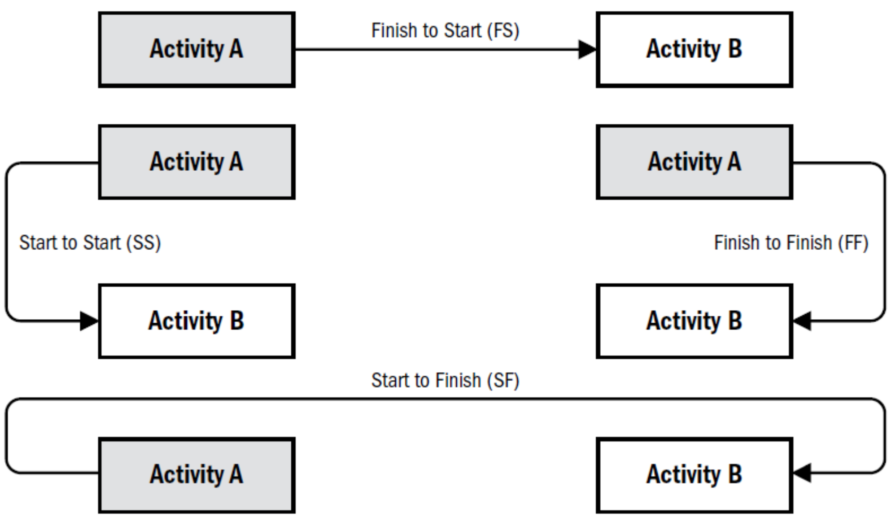

Figure 6. PDM relationship types

- Finish-to-start (FS): The successor activity (Activity B) cannot start until the predecessor activity (Activity A) has finished.
- Finish-to-finish (FF): The successor activity (B) cannot finish until the predecessor activity (A) has finished.
- Start-to-start (SS): The successor activity (B) cannot start until the predecessor activity (A) has started.
- Start-to-finish (SF): The successor activity (B) cannot finish until the predecessor activity (A) has started.

The dependencies of the activities, then, can be distinguished through a set of 4 attributes wherein the two can be applicable at the same time.

- Mandatory dependencies: Also called 'hard logic' or hard dependencies, mandatory dependencies are those that are naturally connected. An example would be creating a prototype or an MVP (most viable product) before it can be tested.
- Discretionary dependencies: Also called as 'soft logic', discretionary dependencies are based on the team's decision and known best practices. For example, one of the best practices in software development is the TDD (test-driven) approach which encourages us to build test cases before the actual development. With enough manpower and time, this is something you would want to do, however, when given a limited time and unclear set of requirements, it may be a good choice to create the test cases during the development phase. This still depends on the constraints and priority of the project.
- External dependencies. This involves non-project and project activities wherein an activity is dependent on another activity that is outside the project teams' control. An example of this is when the testing phase of a software development project is dependent on a procurement and delivery of an equipment or a hardware from another company.
- Internal dependencies. These dependencies are what's iniside the project team's control. Activities can be done by the project team themselves but must be done sequentially as one is dependent on the other.

When adjusting activities and estimating the duration, it would be good to know the possible leads and lags of the activities with dependencies.

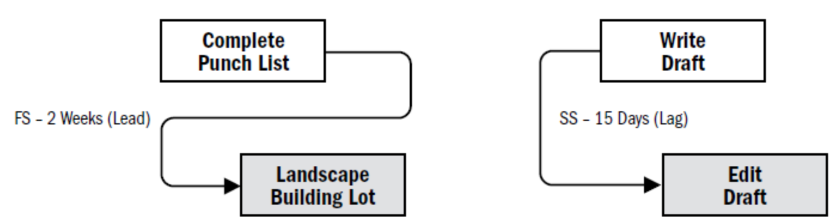

Figure 7. Lead and Lag

A lead is the estimated amount of time an activity can possibly be done early even with the existing dependency of an activity to its predecessor. This also gives us the time when two activities can overlap and be done in parallel.

A lag is the estimated amount of time an activity can be delayed without affecting its successor and without making a significant impact to the entire project schedule.

4. Estimate Schedule Duration

After we have sequenced the activities based on logical and sequential relationships, we can then estimate the duration of each activity. This will give us an idea how long the project should be and which parts must expedited. We will also know the risks of changes and delays.

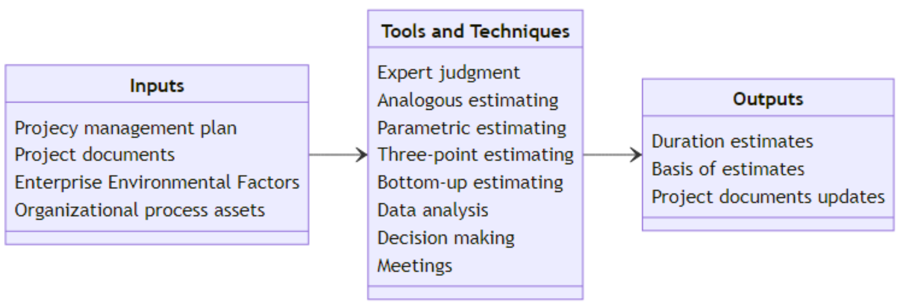

Figure 8. The process of estimating activity duration

The primary goal in this process is to have a close-to-reality estimate of the total expected duration in completing the project by estimating each activity.

There are a number of techniques in estimating. - Analogous estimating: - Parametric estimating - Three-point estimating - Bottom-up estimating

Three-point estimating uses the average of three variables: tM (most likely time duration), tO (optimistic estimate of the duration), and tP (pessimistic estimate of the duration). There are two ways to do this:

- Triangular distribution

$$tE = \frac{(tO + tM + tP)}{3}$$

- Beta distribution

$$tE = \frac{(tO + 4tM + tP)}{6}$$

You may use the triangular distribution if there is not enough historical data to backup your estimates.

The beta distribution is usually used in Program Evaluation and Review Technique (PERT) which is a schedule network analysis technique that estimates project duration as well. PERT can aid in creating schedules that are more realistic.

5. Develop Schedule

Overview

This is the process of drafting and creating the project schedule based on activity sequences, estimated durations, resources, and project constraints. Developing project schedule also covers the time and duration of execution, monitoring, and controlling activities.

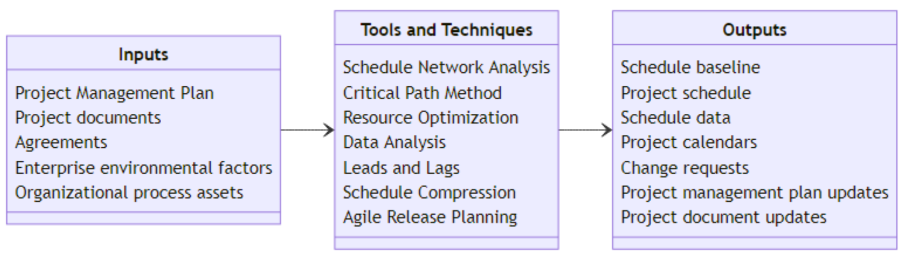

Figure 9. The process of developing a schedule

The outputs of the this process serve as the schedule baseline which will be our basis in monitoring and controlling the schedule. The project schedule can be presented in tabular form but it's more common to present it through diagrams and charts. The two most common representation of the project schedule are:

1. Gantt Chart

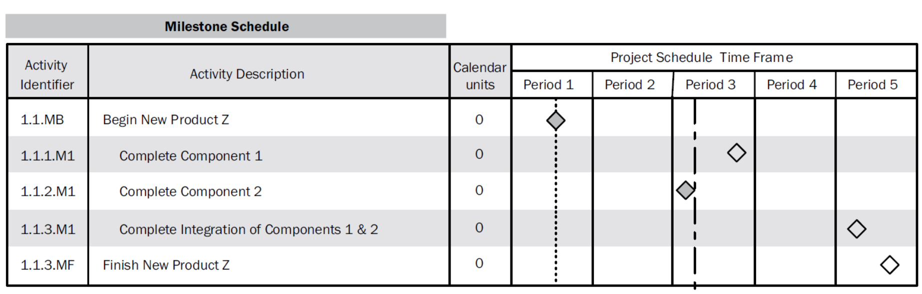

Figure 1: Gantt Chart - Milestone

Figure 9. An example of a Gantt Chart

2. Project Schedule Network Diagrams

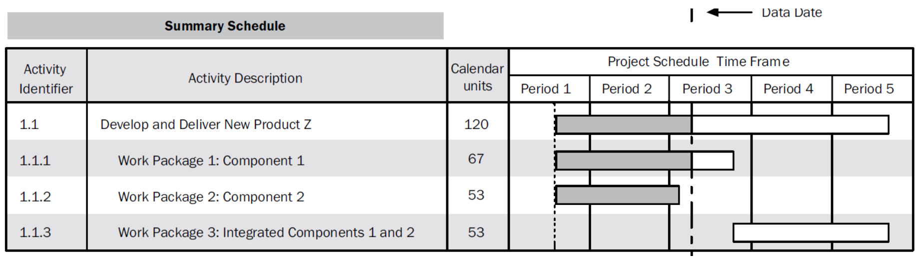

Figure 2: Gantt Chart

Critical Path Method

The Critical Path (CP) is the longest full path of the project based on the sequence of activities and duration estimates from start to end. This gives us an idea of the Critical path method (CPM). CPM is basically a schedule network analysis technique that tries to estimate the total project duration.

The critical path in the example below is A-C-D which has the longest amount of time traversing from start to finish.

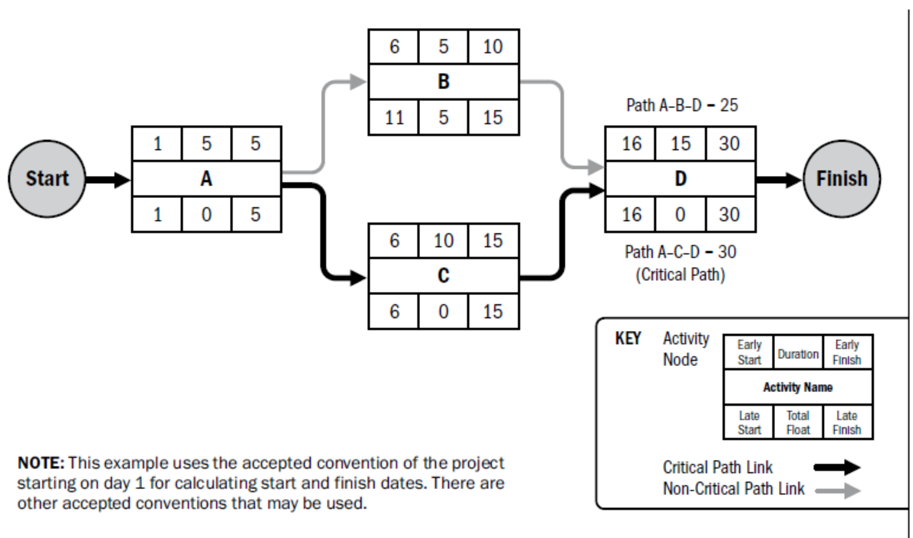

Figure 10. An example of using the critical path method

To compute for the path, we just need to get the sum of the estimated duration of all activities in the path.

Let D be the estimated duration and n be the number of activities in the specified path.

$$path = \sum_{i=1}^{n} D$$

By determining the critical path, we would know which path would cause greater impact to the total project time when changes and delays happen. The critical path is basically the bottleneck route. We can only shorten the total time of the project if we can compress activities in the critical path.

What about the other numbers found in each activity box? We'll dissect each one.

- Total Float (TF) is predetermined by the project team. TF is basically the amount of time the activity can be delayed based on importance of the activity to the project.
- Early Start (ES) gives the earliest possible time when an activity can start.

$$ES_i = D_{i-1} + ES_{i-1}$$

- Early Finish (EF) gives the earliest possible time when an activity can be finished.

$$EF_i = D_i + EF_{i-1}$$

Determining the early start and early finish time of the activities is like a *forward pass* throught the schedule network diagram.

- Late Start (LS) is the latest possible time when an activity can start without affecting the project schedule significantly as to not delay the entire project.

$$LS_i = D_{i-1} + LS_{i-1} + TF_i$$

- Late Finish (ES) is the latest possible time an activity can be finished without affecting the total project time.

$$LF_i = D_i + LF_{i-1} + TF_i$$

Determining the latest start and latest finish time is the *backward pass*.

Critical Chain Scheduling

This is an application of the Theory of Constraints (TOC) which is about the weakest link of a chain wherein it emphasizes the importance of identifying the primary constraint that threatens the project's completion. This constraint should then be something to consider always when managing the project schedule.

Basically, critical chain scheduling considers the limited availability of scarce resources when developing the project schedule. Buffers are also included here.

In critical chain scheduling, we change how we see a “buffer” which is an allowance of time given in case anything goes wrong. Based on Murphy’s Law, if something can go wrong, it will, thus, we should expect interruptions along the way. However, instead of adding buffers or safety nets for each activity, we put the buffer to the whole project which means we have added time for the whole project. So instead of consuming all given time before the due date, we try to create a schedule that would attempt to finish the project before the due date making the gap our new buffer. This is in connection with the Parkinson’s Law which tells us that the work stretches to consume the time given. So, having less buffer in activities and instead putting it to the whole project could make things fast and lessen risk of delays.

Resource Optimization

Resource optimization is about maxamizing the efficient use of resource such that it would aid in completing the project.

Resource Leveling is a technique used when there is a lack of supply and availability of resources required to complete two or more parallel activities. It tries to solve the problem on limited resources by adjusting the start and and finish time.

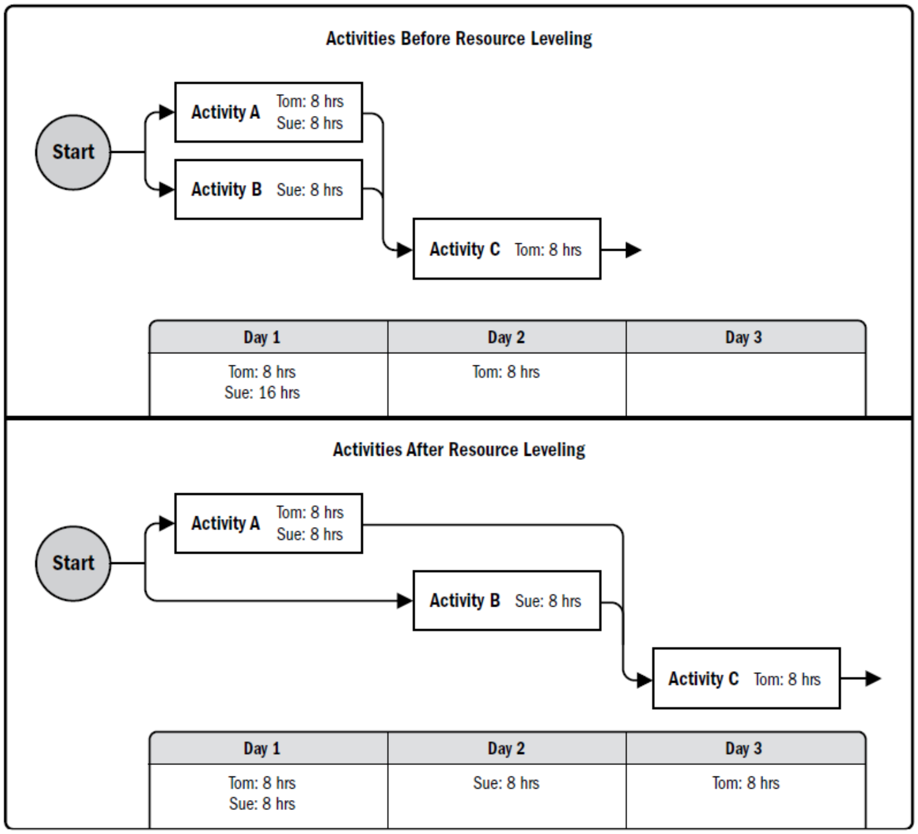

Figure 10. Resource leveling

Schedule Compression

When keeping up with trends and market demand, we might want to accelerate the project and complete it early. Schedule compression is a technique used to try and shorten the schedule duration.

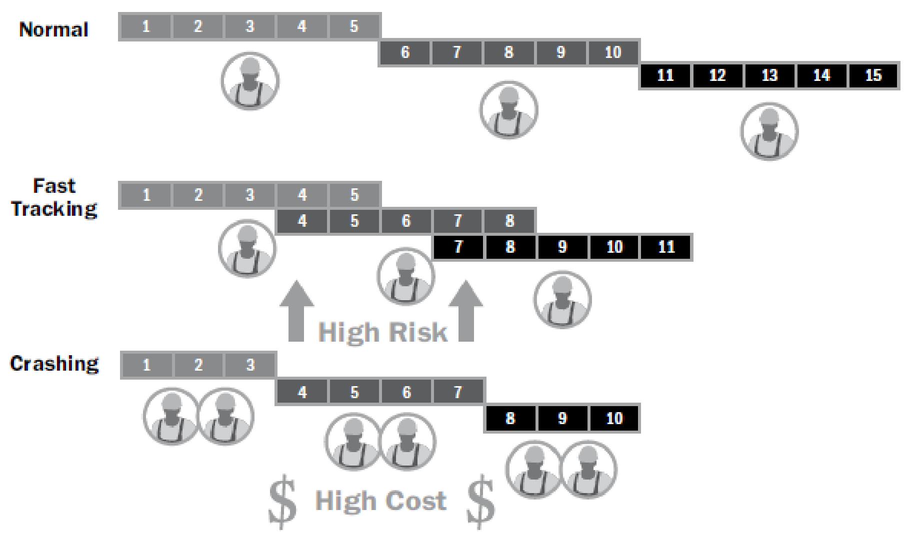

Figure 11. Schedule compress

Crashing. This technique adds resources when shortening the duration which means additional resources will shorten the activities' duration.

Fast tracking. Project is shortened by doing activities in parallel for a limited and certain amount of time. This only works for activities with lead time and can be overlapped with the other activities.

Agile Release Planning

Aside from the number of releases, we also determine the number of sprints or iterations in each release. This gives the product owner and the project team decide how long each deliverable must be developed before we can have a releasable product.

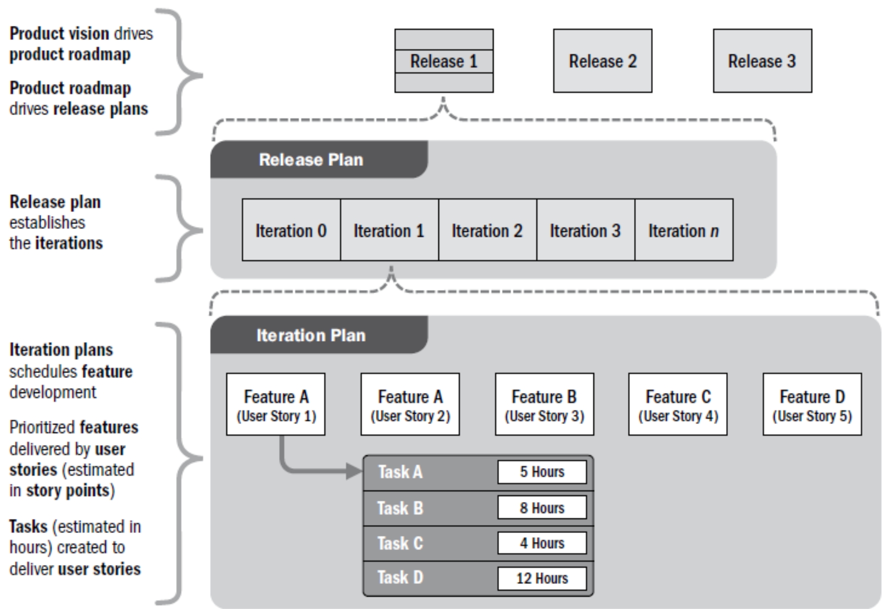

Figure 12. Agile Release Planning process

6. Controlling the Schedule

Overview

This goal of controlling the schedule is to be able to update the schedule baseline throughout the project. This does not mean that the schedule will be fixed as developed because changes will and additional tasks can emerge during the execution of the project. Monitoring and controlling the schedule gives the project manager a hand in keeping the project in line in case of delays.

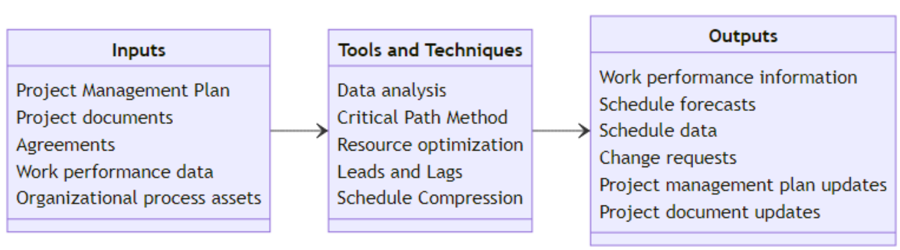

Figure 13. The process of controlling the schedule

In Agile methods, such as SCRUM, we try to create and update an Iteration Burndown Chart for every major deliverable finished or perhaps for every sprint.

This chart shows three aspects of the project. It tracks the remaining work to

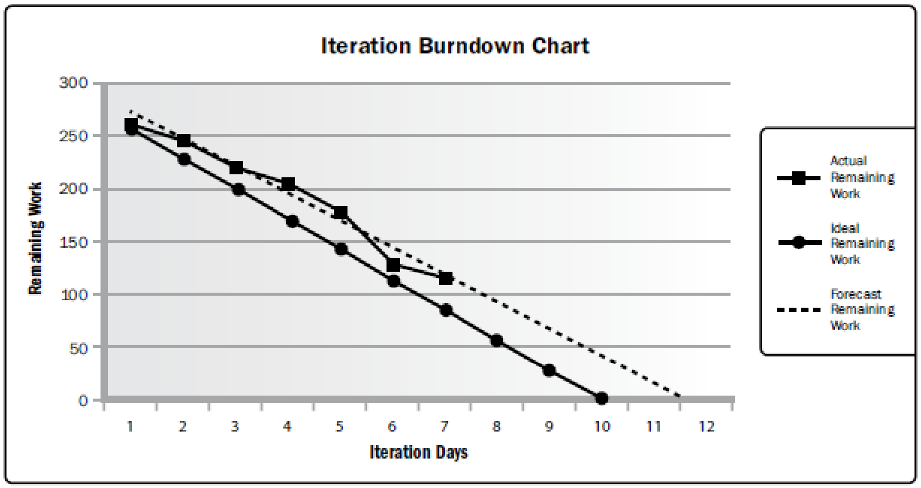

Figure 3: Burndown Chart

be completed. It reminds the team of the ideal burndown of work and then tries to give a forecast the completion of the reamining actual work.