David Luong
NSSA-370
HW#8

8) What are the reasons for creating dependencies of project activities? Explain each one of those reasons with an example.

Project activities have dependencies, including three common dependencies. First, [mandatory dependencies] are required by such factors as legal requirements (e.g., contract agreements); and essentially they represent the relationships between any two core tasks for each activity; these core tasks must be completed before other same-level tasks (and smaller, secondary tasks) can be performed. Mandatory dependencies also ensure that tasks are done properly in specific order, and not randomly nor simultaneously (both occurrences may delay at least one preceding and/or subsequent task, causing an overall delay to the entire project). For example, the project team needs to have at least the product's basic, functional design created – if the product's prototype has not been fully implemented yet – before end users can test the product (or in this example, the product's functional design); this is a [finish-start] relationship. Though rapid design/development, combined with frequent testing at the same time, is effective for identifying missing requirements and/or defects early on, this approach could extend the time to complete the design/development phase because of the project team's need to produce a full-fledged, high-fidelity solution before the project team implements and delivers the product, respectively.

Another group of dependencies, [discretionary dependencies], are usually defined internally "by the project team," and should be included in project activities because they allow the project team to plan necessary project tasks (and their order of completion), easily, to help better monitor, control, and understand the flow of project activities and/or the entire project flow; and to use appropriate, best strategies for getting these tasks completed on time. Unlike mandatory dependencies, which were already established and are bound to happen, discretionary dependencies may be subjected to change as the project progresses, so discretionary dependencies need to be documented regularly, and audited in order to meet organizational standards, and the processes & procedures defined in the project. Discretionary dependencies can be seen directly in the project lifecycle such that project planning may still be done, but not done properly nor effectively without project initiation first; this would be a [finish-start] relationship, but can also be a [start-start] relationship. To be clear, the project needs to start off with a good understanding of the project's high-level requirements and defined scope before plans can be established for the remaining paths to project completion.

Aside from mandatory and discretionary dependencies, there's also [external dependencies], which represent the relationships between both project-related activities and those that are routine-based. External dependencies can also be thought of as prerequisites that need to be met before certain activities can start. For example, the procurement of resources and supplies is an external dependency. The project team cannot carry out the activities nor perform relevant tasks without having the necessary,

tangible tools at hand; this is another [finish-start] relationship. Though the project team may have ordered or requested for resources and supplies ahead of time, a late delivery, an *equipment risk* due to such external risks as slow traffic/poor road conditions and poor weather, would delay the project tasks that require those resources and supplies, postponing the current schedule for the project activities associated with those tasks, and thus, delaying the entire project. In addition, there are two other straightforward dependency relationships. [Finish-finish] indicates that activities and/or tasks are happening at the same time, so all of them must finish in order to be fully complete. [Start-finish] indicates that activities and/or tasks that already started must finish in order to be fully complete. On the whole, the inclusion of dependencies allows the project team to better manage its schedule in order to keep the project on track, and to establish a logical flow of project activities and their associated tasks to ensure the project is delivered on time.

13) Describe forward and backward passes.

Both forward and backward passes are methods used to set reasonable start and end periods for each project activity and its associated tasks, and are based on the project network: a simple, physical model of the full project schedule. The two methods are different in that a forward pass helps to realize how soon, or how early each project activity can start and finish by determining its *duration* (the total actual time for its completion) and the project's *critical path* (the minimal time to project completion upon the consideration of each activity's duration), while a backward pass helps to realize each activity's deadline (or how late it can start and finish) by identifying the total *slack* (the maximum idle time that's allowed without affecting the scheduling of project activities and tasks).

A [forward pass] begins at the project's initial start date (can be set by the project team) – usually at zero for every new project – and moves along the project network from left to right, successively. As the project progresses, the project network shows whether certain project activities are completed early, on time, or late. For the first case, an early completion produces some extra time (a.k.a., reserve time, or buffer time) to be allocated for other important tasks such as documenting and reviewing all the activities done so far. On the other hand, project activities with durations that pass the activities' deadlines produce *slack*, which again, is the maximum time a project activity can be pushed back without affecting the statuses of the remaining activities and tasks. Due to such risk factors as unbalanced schedules and lack of effective resources, negative *slack* is produced; the project schedule needs to be modified in order to reflect on the upcoming plans for preventing additional delays to upcoming activities and the entire project itself. Regarding the updates to project schedule, the activities' earliest start dates and finish dates have to be determined. According to the project network, the {earliest start date} *of the current activity* is equal to the {earliest finish date} *of the preceding activity*; and the {earliest finish date} *of the current activity* combines both the {earliest start date} and the {actual duration} *of the current activity*. Given these basic formulas, the earliest start and finish dates for every subsequent activity can be calculated. The final results will help identify the longest path of the project network

(critical path), which sums up the durations of all project activities shown on the project network, and allows for project completion in the least amount of time.

On the contrary, a [backward pass] begins at the customer's desired date of project completion and moves through the project network in reverse order. Given the earliest start and finish dates of all activities shown in the project network, resulting from a forward pass, the latest start and finish dates can be calculated using a formula that's similar to those used to calculate the earliest times. After a reversal of the order of operations, the {latest start date} *of the current activity* is equal to the {latest finish date} *of the current activity* minus the {actual duration} *of the current activity*. With this formula, the latest start and finish dates of every preceding activity can be calculated. The final results will help determine the total slack for each project activity. On the whole, the ability to identify both time constraints that prevent the project from progressing (by backward passing), and the best direction to take in order complete the project faster (by forward passing), is essential to creating an effective project schedule.

Grade: 86 / 100

Professor's Feedback:

8. Good coverage.

13. Backward and Forward passes are not used to determine or set start/finish dates. They give you a visual representation of the activities. In fact, FP/BP do not even have to show dates in the network diagram.

FP always begin at zero.

A key component missing from the answer was the establishment of the Critical Path.

One can only tell if an activity will be completed early by doing a backward pass.

Backward Pass does not start at customers desired date. In fact, there is no date required (as mentioned above).

You have used the word "constraints" but I don't feel you have a good handle on it. Doing a network diagram does not really shed light on constraints, those have to be assessed individually by activity.