CS 362 Homework 3, Lyell Read

Q.1: Explain why the process of project planning is iterative and why a plan must be continually reviewed during a software project.

Planning is iterative because in the process of planning, one may realize that there are components that are infeasible, or that the allocated time frame is insufficient. For example, if at the start of a project, one were to allocate time for setup of tools and equipment (and associated costs) to the best of their ability, and include those setup costs in the budget they would be taking a risk. This risk is that if something happens to that estimate, the costs and time can be thrown off, and either more time is available for other parts of the project (if the setup was made shorter), or the performers may be in a position of requesting additional funding, as a result of misestimation in the planning stages. For this reason, development must be iterative, so that the performers can return to their customers, and re-consult about which features or goals are to be emphasized and which are to be pushed off, as a result of changing performance times (how fast or slow the performers have performed on the project).

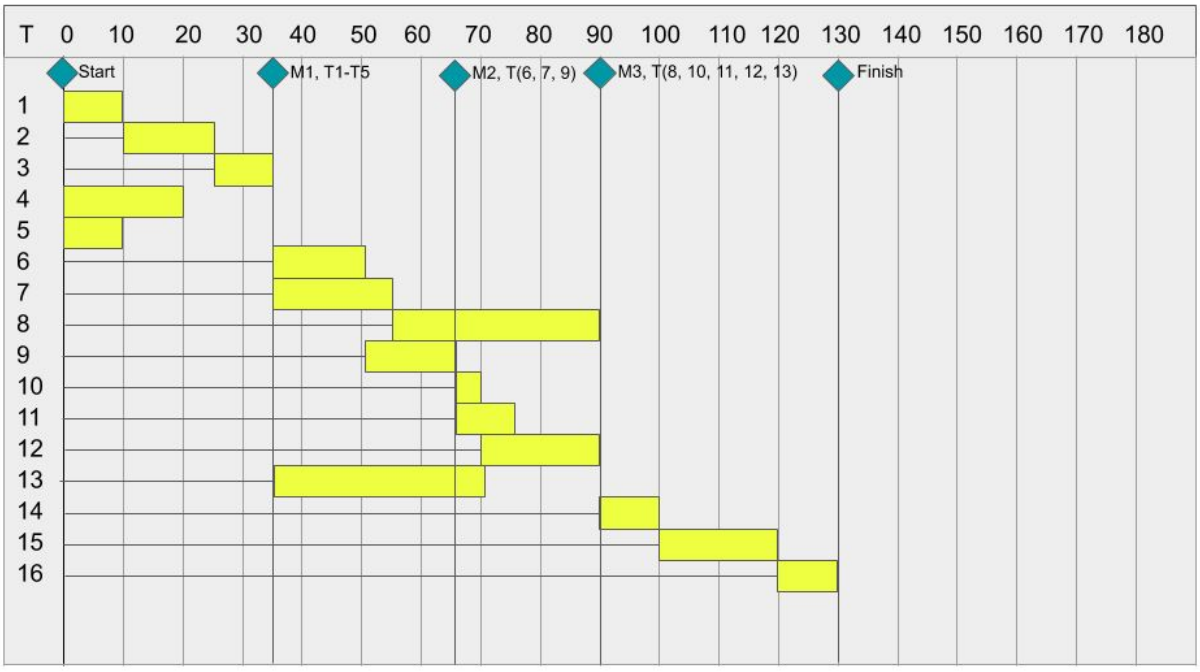
Another example where this iterative, flexible planning method shows how essential it really is is when a customer or customers change. This can happen from small companies (startups, where the user base changes or is reassessed and shown to be different from expected), to large ones (the PM of your project is changed out for another). In these cases, the new customers will almost certainly come into the project with new ideas of what they want done: a feature that used to be considered mission-critical might now be a nice-to-have, as the priorities of certain elements are shuffled (to line up with the customer's). This iterative process allows the performers to reassess with the customer, and change what goals they are aiming towards, and more broadly, to succeed.

A third example of flexibility in planning and performance can be found in unexpected events, such as (like we are seeing now) pandemics, or other disasters like earthquakes, fires, or protests. These can drastically alter the performance timeline of a project. For example, take COVID-19: in the intermission stage between work as usual and quarantine, people are able to perform especially little work, due to many factors like stress, changes in living style. This is facilitated by the flexible process as a project can flexibly adapt their performance deadlines and communicate those with the customer, to keep expectations aligned. Again, in the long run, this keeps the project and the customer informed and synchronized (and each has input) and helps the project and customer succeed.

Q.2: The table below sets out a number of tasks, their durations, and their dependencies. Draw a bar chart showing the project schedule.

Task	Duration	Dependencies
T1	10	
T2	15	T1
T3	10	T1, T2
T4	20	
T5	10	
T6	15	T3, T4
T7	20	T3
T8	35	T7
T9	15	T6
T10	5	T5, T9
T11	10	T9
T12	20	T10
T13	35	T3, T4
T14	10	T8, T9
T15	20	T12, T14
T16	10	T15

Figure 1



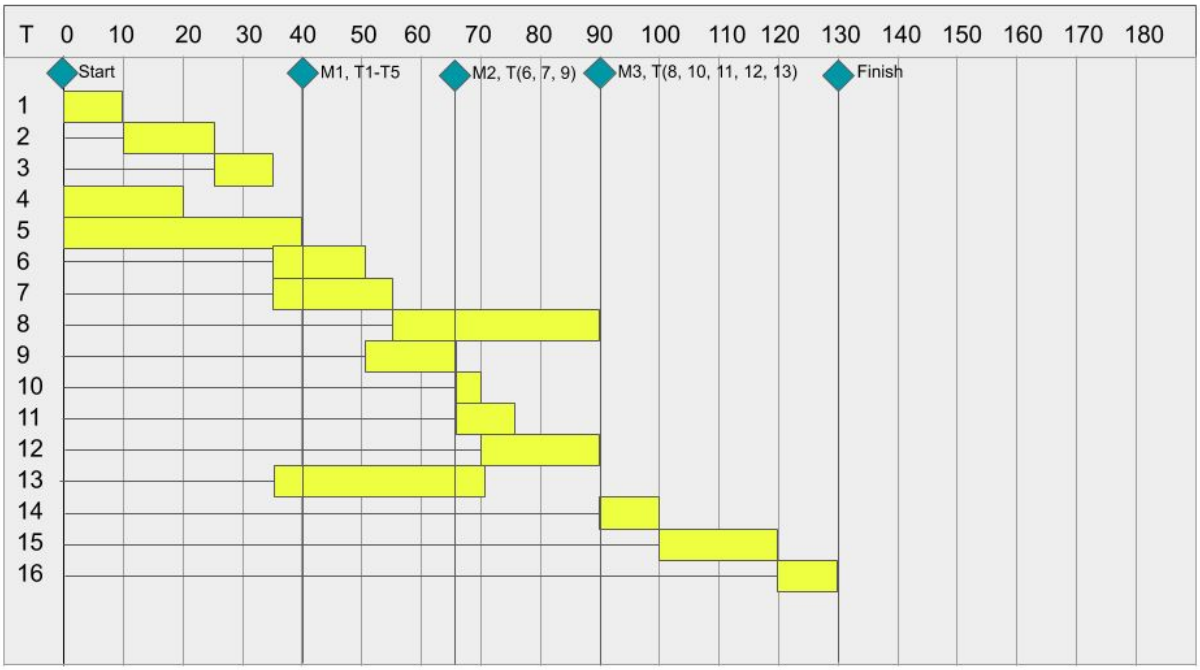
Graph Description

In this graph, the duration of tasks is represented by the yellow-green box, while milestones are determined by the cyan diamond. Each task begins exactly where it's dependants (i.e. at the termination of the last dependant task), and last for their listed duration in days. The lines

leading off the left edges of the boxes are for ease of viewing, to give the viewer an idea of which box corresponds to which task number.

While at first glance, one may notice that there are relatively few milestones, I optimized for milestones that completed approximately when quite a few tasks were set to complete, and that roughly delineated the job to be done into 4 sections or “quarters”. I did so, because humans are already familiar with the 4 quarters system, as we use it to break up the year (as students or as businesses), and these milestones would be more meaningful as they encompass a larger work load. Further, they allow for more reshuffling of included / preceding tasks, as opposed to many smaller milestones.

Q.3: Figure1 shows the task durations for software project activities. Assume that a serious, unanticipated setback occurs, and instead of taking 10 days, task T5 takes 40 days. Draw up new bar charts showing how the project might be reorganized.



Graph Description

In this graph, the duration of tasks is represented by the yellow-green box, while milestones are determined by the cyan diamond. Each task begins exactly where it’s dependants (i.e. at the termination of the last dependant task), and last for their listed duration in days. The lines leading off the left edges of the boxes are for ease of viewing, to give the viewer an idea of which box corresponds to which task number.

While at first glance, one may notice that there are relatively few milestones, I optimized for milestones that completed approximately when quite a few tasks were set to complete, and that roughly delineated the job to be done into 4 sections or “quarters”. I did so, because humans are already familiar with the 4 quarters system, as we use it to break up the year (as students or

as businesses), and these milestones would be more meaningful as they encompass a larger work load. Further, they allow for more reshuffling of included / preceding tasks, as opposed to many smaller milestones.

Question
As a result of the issue that hindered completion of Task 5 (it's length was increased from 10 to 40 days), the first milestone had to move five days further so that it's member task T5 could be part of that milestone. Surprisingly (and thankfully to the team that would be effectuating this plan), none of the other tasks had to shift because of this change: the only task that might have had to change is Task 10 (which relies on Tasks 9 and 5), but it started much later than 40-day-T5 ran for, so there was no need to move it.

It is plausible to consider that the same issue that affected completion of tas T5 might also affect other tasks - if this were the case, the same process of (1) updating dependent tasks, and (2) updating milestones would need to be undertaken for each impacted Task.

Q.4: Cost estimates are inherently risky, irrespective of the estimation technique used. Suggest four ways in which the risk in a cost estimate can be reduced.

Hire an Expert
I would estimate that the best way to get an accurate budget estimate for a project is to hire or contract with someone who has either (1) been successful or had experience budgeting a similar type, scope, and field project, or (2) has been shown to be an excellent budget regardless of project field. This contractor or new hire will have experience budgeting a similar project, and therefore has a track record that demonstrates success. This can be paired with the below tactics to increase success in budgeting and reduce risk.

Over Budget
The best way to tackle the risk issue of budgeting is to over budget for each task, to account for fluctuations and unforeseen changes. This does not work as well in all kinds of institutions, as some do not have the capital to make such additional investments, and some are not willing to accept the tying up of financial resources. That second point leads to the second issue with this: it ties up capital. While it does decrease risk with relation to budget, it is inefficient, especially if a company invests their free capital in a trust, where it could be earning money while stagnant. However, when it comes to risk, almost all the risk in budgeting comes from under budgeting, and ending up (a) not being able to perform / finish the project as a result of lacking funds or (b) having to ask for more funding at a later time.

Consult Performers
It is often the case that those performing on a project will know better what the project will need to be successful. Consider the case where the goal is a website, and the budget estimates that the backend will cost 100 man hours. However, the backend developer, who has many years experience, informs the budget or that in fact, this will take 250 hours, as a result of there being

a recently decommissioned SQL API that they will have to rewrite or maintain. This would only be known to someone who does backend, and would have created under budget risk if not for talking with the performers.

Headroom

Headroom is a method of reducing risk that might be considered a subcomponent of over budgeting. However, where over budgeting deals with the risk of running out of money by budgeting a bit extra for each item or performance needed, headroom ensures that there is a chunk of free capital to account for a serious change, i.e. a pandemic or natural disaster, health emergency. This creates a bit of buffer or headroom between the project and the risk of running out of money.