CSC584: Assignment 2
Submitted by: Malika Hafiza Pasha

Chapter 22: Project Management

1. **Identify all possible software project management activities. Explain why the best programmers do not always make the best software managers. You may find it helpful to base your answer on the list of management activities in Section 22.1.**

Answer:

Software project management activities include:

a. **Project Planning:**
This entails work assignment, scheduling, and estimation for project development. It entails overseeing the work to make sure it satisfies the necessary requirements and keeping an eye on developments to maintain the development on schedule and within budget.

b. **Reporting:**
It is the duty of project managers to update clients and the managers of the software development business on the status of the project. This involves the capacity to convey information at many levels, from in-depth technical details to executive summaries, and to do so while conducting progress evaluations.

c. **Risk Management:**
This includes identifying risks that could have an impact on the project, keeping an eye on those risks, and acting when issues emerge.

d. **People Management:**
One of the most important duties is leading the group of individuals working on the project. This entails deciding on team members and setting up efficient procedures.

e. **Proposal Writing:**
Writing a proposal to secure a contract, including the project's goals and how it will be executed in addition to cost and schedule estimates, may be the first step in a software project.

These exercises demonstrate the diverse responsibilities of a project manager, who must possess a good mix of managerial, technical, and interpersonal skills. Considering these endeavors, it is possible to comprehend **why the best programmers are not always the best software managers**. Beyond programming's technical proficiency, project management requires a variety of other abilities. They include communication skills, strategic planning, risk management, and team leadership. Understanding the complexities of software development requires a solid technical background, but effective project management also calls for a broader skill set that includes:

1) **Leadership and Team Management:**
Establishing, inspiring, and keeping a harmonious team, settling disputes, and guaranteeing productive cooperation.

2) **Strategic Planning and Risk Management:**
Recognizing possible issues, preparing for unforeseen circumstances, and devising a plan to accomplish project objectives within financial and schedule limitations.

3) **Communication Skills:**
Having effective communication with clients and higher management in addition to technical teams and non-technical stakeholders.

As a result, **even while highly skilled programmers have extensive technical knowledge, they also need to acquire new skills in communication, strategy, and leadership to go into management**. The most successful software managers are those who can successfully apply their technical expertise to these more general administrative responsibilities.

2. Discuss software project risk management process and describe 12 or more possible risks that could arise in software projects.

Answer:

The software project risk management process involves four key stages:

a. **Risk Identification:**
Identifying potential project, product, and business risks.

b. **Risk Analysis:**
Assessing the likelihood and consequences of these risks.

c. **Risk Planning:**
Developing plans to address the risk either by avoiding it or minimizing its effects on the project.

d. **Risk Monitoring:**
Regularly assessing the risk and revising mitigation plans as more information becomes available.

Possible risk that could arise in software projects are:

a. **Staff Turnover:**

The possibility of losing team members in the middle of the project, which could result in knowledge loss, low morale, and higher training and hiring expenses.

b. **Management Change:**

Team dynamics, priorities, and project continuity can all be affected by changes in project or organizational leadership.

c. **Hardware Unavailability:**

The chance that necessary hardware won't be accessible when needed, which could cause the testing and development phases to be delayed.

d. **Requirements Change:**

Project scope creep, higher expenses, and delays can result from frequent or major modifications to the requirements.

e. **Specification Delays:**

Project schedule can be affected, and development deadlines pushed back by delays in establishing specifications.

f. **Size Underestimate:**

If the project's size or complexity is underestimated, insufficient resources may be allocated, which could result in cost and time overruns.

g. **CASE Tool Underperformance:**

Reliance on subpar Computer-Aided Software Engineering (CASE) technologies can reduce software quality and productivity.

h. **Technology Change:**

Rapid advancements in platform or underlying technology can make the original project assumptions dated, requiring expensive redesigns or rework.

i. **Product Competition:**

The project's product's marketability and relevancy may be impacted by newly developed rival goods.

j. **Organizational Financial Problems:**

An organization's financial instability may result in project cancellation, resource shortages, or budget reduction.

k. **Recruitment Problems:**

The inability to find employees with the necessary skills can impede the execution of a project and drive-up expenses because skilled workers may require more training or pay.

l. **Staff Illness:**

Unexpected employee illness can seriously affect project timeframes and job distribution, especially when it affects critical workers.

m. **Defective Components:**

Using faulty software libraries or components might result in more testing and debugging work, which can affect project schedules and quality.

n. **Organizational Restructuring:**

Organizational structure modifications may have a negative impact on project priorities, resources, and support, which could halt development.

o. **Database Performance Issues:**

Bottlenecks can impact the software's overall performance and scalability if the database performs below expectations.

p. **Underestimated Development Time:**

Ignoring how long development tasks take can result in hurried work that is of poor quality and misses deadlines.

3. **Propose a case study in the style used here to illustrate the importance of communications in a project team. Assume that some team members work remotely and that it is not possible to get the whole team together at short notice.**

Answer:

Considering a hypothetical scenario:

Case Study: Amfil's Remote Project Team:

A crucial project involving the construction of a new financial analytics platform was started by the software development business Amfil. The project team was made up of people from France, India, and the United States, among other countries. It was not feasible to have a last-minute physical meeting with the full team due to their geographical dispersion.

Issues with Project Initiation and Communication:

When the project first started, everyone in the team was excited to pitch in. But as the weeks went by, the group ran into serious communication problems. Time zone variations caused responses to be delayed, and the absence of in-person interactions made it challenging for team members to build rapport and trust. Email conversations contained crucial project data that were lost, and problems in the development process resulted from misreading written correspondence.

Putting Communication Strategies into Practice:

The project manager made the following strategic communications enhancements to address these issues:

a. **Planned Frequent Video Conferences:**

In order to guarantee that every team member could attend at least one meeting, the project manager implemented biweekly, all-team video conferences at times that accommodated all time zones. During these meetings, questions were answered, project status was discussed, and problem-solving ideas were generated.

b. Use of Collaborative Tools:

Trello was used by the team for task management, and Slack was used for real-time messaging. With the aid of these tools, the amount of email traffic was decreased and real-time updates on the project's status were provided to all parties.

c. Understanding the diversity of the team:

The project manager set up cultural sensitivity training so that individuals could respect and comprehend one another's communication and cultural backgrounds.

d. Meeting Time Rotation:

The team's meeting times were switched around to equally divide the annoyance of strange meeting hours. This made sure that no one on the team was always made to stay up late or early in the morning for meetings.

d. Establishing a Virtual Water Cooler:

To facilitate conversations unrelated to work, a special channel was made where team members could exchange jokes, interests, and personal news. Team members were able to develop empathy and personal relationships as a result of this program.

Results:

The team's cooperation and communication improved dramatically as a result of these tactics. Project milestones were reached more effectively, and the caliber of the output rose. The team members' increased sense of camaraderie and dedication to the project's accomplishment showed how good communication can overcome the difficulties associated with geographical dispersion and remote work.

Summary:

This case study highlights the importance of communication in distant project teams and the ways in which proactive approaches can lessen the difficulties brought on by geographical dispersion. Amfil's experience demonstrates that remote teams can succeed in projects and operate at high levels of efficiency with the appropriate strategies.

4. **Your manager asks you to deliver software to a schedule that you know can only be met by asking your project team to work unpaid overtime. All team members have young children. Discuss whether you should accept this demand from your manager or whether you should persuade your team to give their time to the organization rather than to their families. Identify 5 significant factors in your decision.**

Answer:

Managing a team of people who have small children and having to work under pressure to produce software on time while also demanding paid overtime presents difficult managerial, social, and ethical decisions. Here are five important things to think about while considering whether to comply with this request or convince the group to put work demands ahead of family time:

a. **Team Morale and Well-Being:**
Requiring unpaid overtime has a big impact on team morale and members' mental health. Considering their obligations to their families, particularly for those with small children, can cause stress and burnout, which may eventually lower output and lower the caliber of work.

b. **Legal and Ethical Considerations:**
Since many jurisdictions have labor laws that protect employees' rights regarding work hours and overtime pay, it is important to think about the legal ramifications of forcing unpaid overtime. Respecting employees' work-life balance, appreciating their efforts, and paying them appropriately for overtime are all imperative from an ethical standpoint.

c. **Project Management and Scheduling:**
This example illustrates possible problems with project management and planning. The project's budget, scope, and schedule may need to be reviewed. Meeting deadlines without frequently turning to unpaid overtime is a sign of inefficient project management, which may also point to impractical scheduling or resource allocation.

d. **Long-term Organizational Impact:**
While achieving the deadline may result in short-term benefits, it's important to think about how it will affect the team's cohesiveness, loyalty, and reputation over the long run. Prolonged employee attrition, a decline in job satisfaction, and possible bad press from these actions may offset the short-term advantages.

e. **Alternative Solutions:**
Examine several approaches to completing the project by the deadline before deciding. This can entail setting project job priorities, contacting temporary outside help, or working with stakeholders to agree on a more flexible timeline. In addition to fostering creative solutions, involving the team in these conversations shows respect for their personal and professional life.

The urgent needs of the project and the organization must be balanced with the rights and well-being of team members to choose whether to agree with the manager's demand. A more ethical and long-lasting solution may result from a careful strategy that considers the aspects.

Chapter 23: Project Planning

5. **Explain why the process of project planning is iterative and why a plan must be continually reviewed during a software project.**

Answer:

The dynamic character of software projects, where changes are the norm rather than the exception, is highlighted by the requirement for ongoing review and iteration of the project plan. Throughout a project's lifecycle, new information, opportunities, and problems must be taken into account and plans must be adjusted accordingly for effective project management. Iterative project planning in software development is necessary for a number of reasons, all of which call for ongoing evaluation during the project:

a. **Planning is iterative:**
It starts at the beginning of the project and is updated on a regular basis in response to new knowledge and expertise. This iterative method enables the plan to be modified in response to modifications in the capabilities of the development team, software needs, and other project-related considerations.

b. **Modifications and Uncertainties:**
Unexpected difficulties, shifting corporate objectives, and requirements modifications are commonplace in software projects. Since no plan can predict every scenario that could arise, the original one created at the beginning of the project must be flexible enough to account for these shifts.

c. **Learning and Adaptation:**
The project team gains knowledge about the software being produced, the technical difficulties, and the efficacy of their planning and execution techniques as the project moves along. The strategy must be modified in light of this knowledge to properly account for the time, money, and resources needed.

d. **Risk management:**
Identifying possible risks and creating plans to reduce them are important aspects of project planning. In order to address new risks or modifications in the evaluation of current risks, the strategy must be reviewed and adjusted on a regular basis.

e. **Stakeholder Engagement:**
Continuous review and iteration of the project plan ensure that stakeholders are kept informed and engaged, allowing for their input and adjustments to the plan based on their feedback and changing priorities.

f. **Optimization and Efficiency:**

Iterative planning allows for the optimization of resource allocation, prioritization of tasks, and adjustment of project schedules to ensure the most efficient use of resources and timely delivery of the project.

6. **Cost estimates are inherently risky, irrespective of the estimation technique used. Suggest fourways in which the risk in a cost estimate can be reduced.**

Answer:

Adopting techniques that improve accuracy, permit flexibility, and incorporate a thorough grasp of the project and its possible uncertainties are all necessary to lower the risk associated with cost estimating. The following four strategies will lower the risk in cost estimates:

a. **Employ Benchmarking and Historical Data:**

When estimating costs, take into account past performance on related projects. This method can be useful in determining probable cost sources and learning how comparable problems have been handled in the past. A reasonable starting point for expenses can also be obtained by benchmarking against projects completed by rivals and industry norms. By depending on real project outcomes rather than theoretical assumptions, this strategy aids in the creation of estimates that are more accurate.

b. **Include Contingency Planning:**

Funds designated for unanticipated expenses are known as contingencies. Your cost estimate might act as a financial safety net to help you manage the risks brought on by project uncertainties by including a contingency allowance. Based on a risk analysis of the project, which takes into account variables such project complexity, duration, and the accuracy of the cost data, the size of the contingency should be determined. This strategy lessens the financial burden of unanticipated events but does not eliminate uncertainty in and of itself.

c. **Conduct Sensitivity Analysis:**

Sensitivity analysis is the process of adjusting one or more cost factors and observing the impact of those adjustments on the estimate as a whole. Project managers can use this method to determine which costs have the biggest influence on the overall estimate, which helps them determine where the estimation risk is largest and where to concentrate their risk mitigation efforts. You can more effectively control and reduce risks associated with those cost components by knowing which factors are most sensitive.

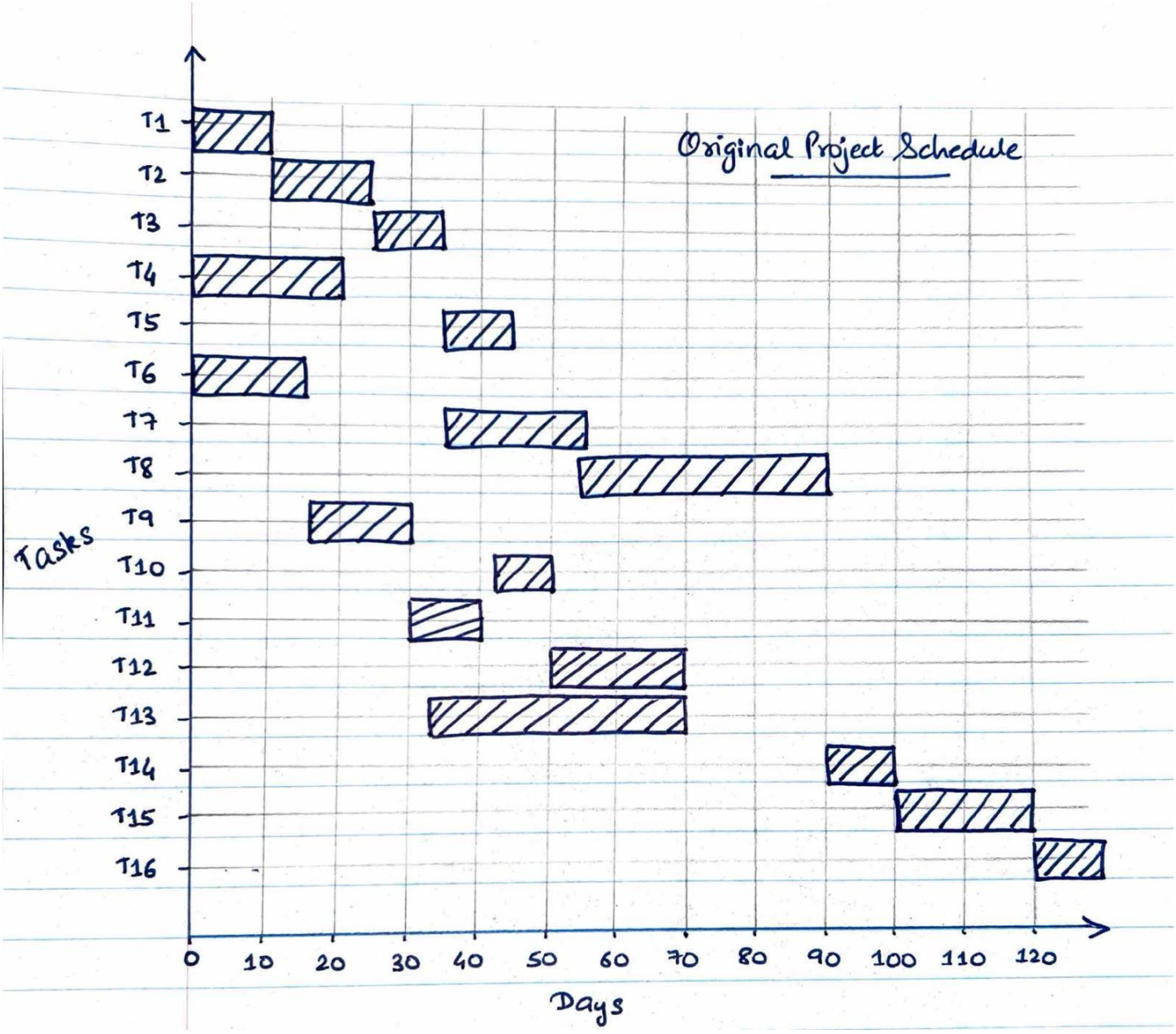
d. **Employ a Phased Estimation Method:**

As the project develops, more precise and in-depth cost estimates can be made by segmenting it into smaller, more manageable segments. Though estimates can be more generic in the beginning, they can be more precise as the project's scope becomes clearer and more data becomes available. This method lowers the risk related to uncertainty in preliminary estimations by enabling modifications as new information becomes available.

7. **Figure 23.14 in the textbook sets out a number of tasks, their durations and their dependencies. Draw a bar chart showing the project schedule. Assume that a serious, unanticipated setback occurs, and instead of taking 10 days, task T5 takes 40 days. Draw up new bar charts showing the project might be reorganized.**

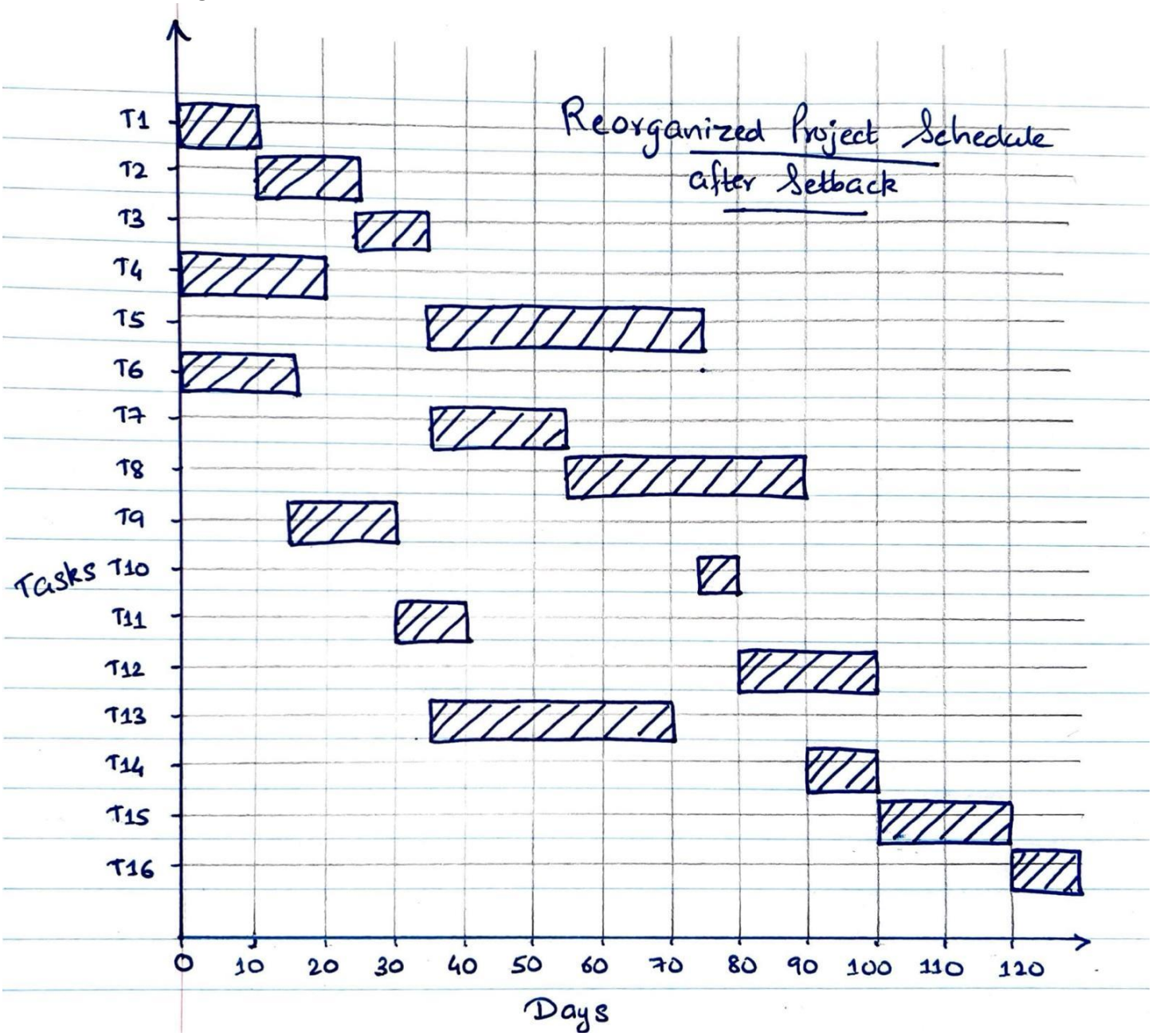
Answer:

Based on the dependencies and durations of each activity, we must determine the start and end dates in order to produce a bar chart that illustrates the project schedule. Following that, we may modify the itinerary to reflect T5's delay. Let's compute the initial schedule first, and then make the bar graph. After that's finished, I'll make a new bar chart that displays the rearranged project and update the timetable to account for the delay in T5.



This is an illustration of the initial project schedule in bar chart form. Every bar represents a task; the position denotes the start and end days, and the length indicates the time.

Let's now proceed to modify this timetable to take into account the fact that T5 is taking 40 days to complete instead of 10 days. All jobs that directly or indirectly depend on T5 will be impacted by this. After recalculating these tasks' start and end times, we'll create a fresh bar chart.



8. **Discuss the Wideband Delphi estimation process and identify its steps. Consider the following case: There are n estimators E_1, E_2, …, E_n in an estimation team that is using Wideband Delphi process to estimate the cost of a software project. Suppose that, in estimation round i, the estimator E_j has estimation $C_j(i)$, and the average estimation is C(i). In the next round i+1, E_j update his/her estimation $C_j(i+1) = C_j(i) + (C(i)-C_j(i))*p_i$, where p_i is a percentage. Show, if all p_i's are the same, then the average of the estimation will never be changed.**

Answer:

A technique used in software engineering to estimate project effort and expense is the Wideband Delphi estimation process. It is an iterative process meant to bring together the ideas of experts in order to come to an estimate consensus. The procedure is based on the Delphi technique, but it encourages greater participation and dialogue between the participants. The main steps in the **Wideband Delphi estimate procedure** are as follows:

a. **Preparation:**
The coordinator sets up the estimating assignment, providing a thorough project description or a list of the particular elements that need to be estimated. This entails obtaining all relevant records and data to support the estimators in making defensible choices.

b. **Kick-off Meeting:**
To discuss the estimate problem, the team of estimators (E1, E2,..., En) gets together. The coordinator explains the assignment, answers any questions, and makes sure that everyone is aware of the estimation's parameters.

c. **Individual Estimation:**
Without consulting the others, each estimator individually calculates their own estimate (Cj(i) for estimator Ej at round i). By doing this, bias and outside influence from other team members are reduced.

d. **Estimate Sharing:**
After the estimators get back together, everyone shares their estimates with the group. If the coordinator gathers the estimations and provides them to the group, then the sharing can be done in an anonymous manner.

e. **Discussion:**
Following the sharing of all estimates, the group talks about the rationale behind the various estimates. Here is an important opportunity for estimators to challenge any underlying presumptions, methods, or other variables that may have contributed to the disparity in estimates.

f. **Revision:**
Each estimator makes any required revisions to their estimate in light of the conversation. Here's when your updated estimate formula comes into play:

$$Cj(i+1) = Cj(i) + (C(i) - Cj(i)) \times pi,$$

Where pi is a percentage that represents the amount of the difference that each person wants to make in their estimate for the following round between their estimate and the average of the group.

g. **Repeat Rounds:**
Until the group comes to an agreement or the estimates converge within a predetermined acceptable range, steps 4 through 6 are repeated multiple times.

h. **Final Estimate:**
The coordinator gathers the last round of estimates and, frequently, uses the average or median of the final estimates to ascertain the consensus estimate.

Addressing the mathematical assertion that the estimation's average will never change if all pi's are the same.

Given:

Cj(i) -> estimation of Ej at round i

C(i) -> average estimation at round i

For the next round i+1, the updated formula is Cj(i+1) = Cj(i) +(C(i) – Cj(i)) x p, where p is constant across all estimators.

Let's **prove** that the average C(i+1) will be w=equal to C(i) under these conditions.

Once all estimators have updated their estimates in accordance with the formula, we will compute the new average C(i+1) and demonstrate that it equals C(i).
The computation verifies the assertion by demonstrating that the new average estimation following the update, C(i+1), simplifies to C(i). This indicates that the group's overall average guess will remain unchanged in the following round if each estimator modifies their estimate according to the same percentage of the difference between their personal estimate and the group average.

9. **A software manager is in charge of the development of a safety-critical software system, which is designed to control a radiotherapy machine to treat patients suffering from cancer. This system is embedded in the machine and must run on a special-purpose processor with a fixed amount f memory (256 Mbytes). The machine communicates with a patient database system to obtain the details of the patient and, after treatment, automatically records the radiation dose delivered and other treatment details in the database.**

   The COCOMO method is used to estimate the effort required to develop this system and an estimate of 26 person-months is computed. All cost driver multipliers were set to 1 when making this estimate.

   Explain why this estimate should be adjusted to take project, personnel, product, and organizational factors into account. Suggest four factors that might have significant effects on the initial COCOMO estimate and propose possible values for these factors. Justify why you have included each factor.

Answer:

The baseline COCOMO (Constructive Cost Model) estimate of 26 person-months is based on average software development conditions. However, given the particulars and limitations of the project, the

team, the software product, and the organizational environment, this estimate might not fully capture the actual effort needed. It is imperative to modify this estimate for multiple reasons:

a. Project factors:
It includes the software system's size, complexity, and needed dependability. The requirement for high dependability and adherence to strict safety requirements (e.g., FDA regulations for medical equipment) can greatly increase the development work for a safety-critical system, such as a controller for a radiotherapy machine.

b. Personnel Factors:
Productivity is significantly impacted by the development team's experience and skill, particularly when it comes to creating safety-critical systems. A team with a lot of experience in this field might finish the job faster than one with less experience.

c. Product Factors:
Development may be more difficult due to the software's unique requirements, which may include real-time performance, fault tolerance, and the requirement to run on a special-purpose CPU with fixed memory. These limitations frequently call for more advanced testing and design techniques.

d. Organizational Factors:
Project effort may also be impacted by the organization's development procedures, resources, and culture. Productivity may be higher in companies with established software development procedures (CMMI Level 4 or 5 are two examples).

Proposed Elements and Potential Values

a. **Software Reliability Requirements (RELY):**
   Reliability is crucial for a radiotherapy machine to guarantee patient safety. An elevated RELY value signifies the heightened endeavor to attain the requisite dependability. One potential result could be "Very High" (1.40).

b. **Product Complexity (CPLX):**
   Developing this program probably required more work than usual because of its real-time nature and specific hardware requirements. "High" (1.34) is one possible result that would suggest the need for more thorough testing and design.

c. **Team Capability (TEAM):**
   The effort needed may be lessened if the team has a great deal of expertise creating comparable safety-critical systems. On the other hand, the effort might be greater if the squad is inexperienced. For a team with experience, a potential score might be "Very Good" (0.85), while for a team with less experience, "Low" (1.20).

d. **Process Maturity (PMAT):**
   Productivity and quality can be greatly impacted by how mature an organization's software development process is. Higher levels of maturity might make work easier. A potential value for an organization at CMMI Level 3 may be "Nominal" (1.00), but a potential value for an organization still refining its processes could be "Low" (1.12).

Justification:

a. RELY:
The system's failure could result in improper treatment dosages, putting patient lives in jeopardy, which is why inclusion is justified. Therefore, ensuring high reliability is essential and calls for more work in the design, development, and testing phases.

b. CPLX:
The system's requirement for a dependable interface with external databases and its use of a fixed-memory, special-purpose processor add to the difficulty of development. This means that integration and optimization testing will take longer.

c. TEAM:
The effectiveness and caliber of the development process are directly impacted by the development team's capabilities. Teams with experience can handle complicated project requirements more skillfully, which minimizes effort in the end.

e. **PMAT:**
The predictability and efficiency of a project's completion are impacted by process maturity. Greater project management, quality assurance procedures, and risk management are typically associated with higher maturity levels. This results in more precise effort estimation and project execution.

A more accurate picture of the work needed is provided by adjusting the COCOMO estimate with these variables, which facilitates improved risk management, budgeting, and planning.