CSC-584 Assignment-2

By Neil Duraiswami

SID: 212620514

**Contents**

[**Chapter 22: Project Management** 2](#_Toc158574458)

[**Chapter 23: Project Planning** 7](#_Toc158574459)

[**References:** 17](#_Toc158574460)

# **Chapter 22: Project Management**

**1.**

The software project management activities are:

1. **Project planning:**

* Project planning, estimates, and timing.
* Placing individuals on projects so they collaborate.
* To monitor the work to ensure it meets original requirements/ specifications.
* Keeping an eye on the progress to ensure it is within budget and schedule.

1. **Reporting:**

* Keep clients and business management informed about the project status.
* This entails communication from summaries of management to in-depth technical information.
* Prepare clear and well-organized documents that extract vital details from extensive project reports providing data when appraising progress.

1. **Risk Management:**

* Recognizing the potential for risks to impact software quality or the project schedule.
* Taking actions to reduce identified dangers.
* Assessing business, product, and project risks.
* Observing and updating strategies to mitigate risk.

1. **People management:**

* Leading a group of people.
* Choosing team members.
* Implementing effective teamwork methods that will boost output.

1. **Proposal writing:**

* Applying for software project contracts is done through the submission of an application.
* The goals and procedures used in the project are described in this section.
* This part contains cost and timeline projections.
* Justify why a certain group or team was chosen to get the project contract.

Here's why the most skilled programmers might not always be the most effective software managers:

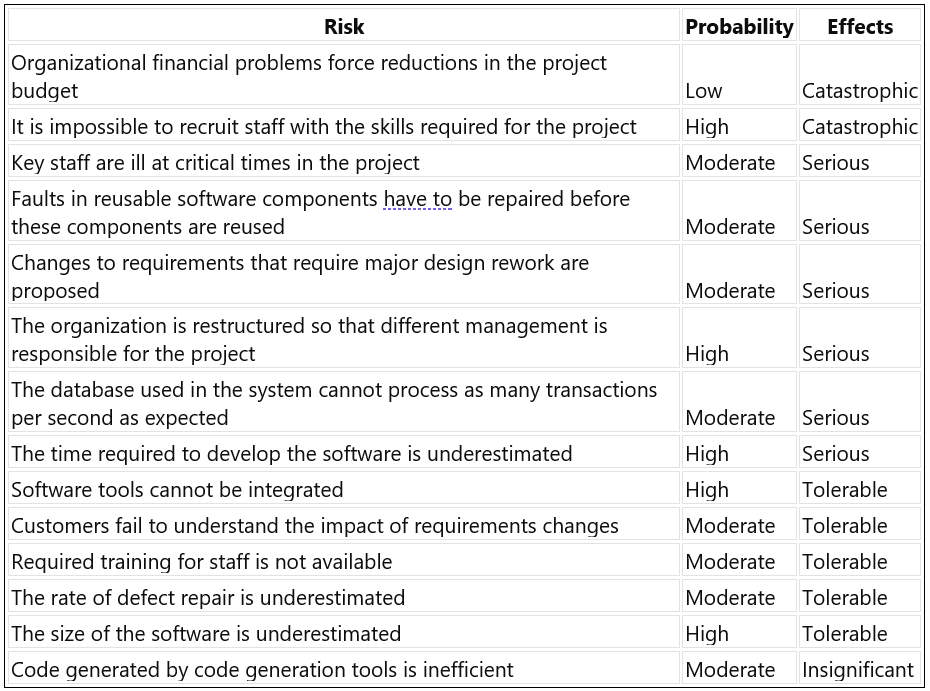
* **Focus Shift:** Programming and management require different skill sets and mindsets. While programming is all about technical problem-solving and coding, management entails project planning, reporting, risk management, personnel management as well as proposal writing. For some programmers, it may be hard to move from practical hands-on technology work to supervising these various activities.
* **Communication Skills:** To make their tasks satisfactory good managers need clear project aims which they should delegate responsibility for and then offer feedback upon the whole process accompanied by conflict resolution. Although gifted programmers can be excellent communicators within their teams, they might not have the broader communication skills necessary for interacting with clients, stakeholders, and other departments.
* **Project Planning and Reporting:** Software managers oversee organization, estimate, schedule, and report on their projects. For this purpose, strategic thinking, attention to detail, and effective resource management are a must. Even though programmers may be good in technical matters, they may struggle with other general management issues of a project.
* **Risk Management:** Risk management is an integral part of software management that entails the identification, evaluation, and minimization of project risks. This involves anticipation of likely barriers, preparing backup plans as well as being judicious in decision-making to minimize risk. Technical problem-solving may be the forte of programmers but they might not have the knowledge required for evaluating and handling significant project risks.

**2.**

To begin with, the uncertainties confronting software projects such as poorly described requirements, changes in customer needs and lack of time and resource estimation needs make the risk management process crucial to them. It is also significant for Software Project Managers to identify risks, analyze them, plan for them, and monitor them so that they can anticipate potential problems and take necessary actions in advance. In this way project completion delays, budget overruns, or cost escalation risks as well as issues of quality are avoided which guarantees successful outcomes of the projects; thus, minimizing their negative impacts on the project, product line, and firm at large. In addition, by documenting a risk management plan that shows the risk management process stakeholders would be well informed and have transparency throughout the lifecycle of the project according to the Project Manager’s perspective.

The processes in software project risk management are as follows:

* **Risk identification:** Risk identification is the process of determining potential hazards to projects, products, and businesses. The identified risks are put down in a table with their probability and possible outcomes.
* **Risk Analysis:** Analysis of risk is when an assessment is done on the probability and effects of a risk that has been found. Each specified risk’s likelihood and consequences are displayed in the table below.
* **Risk Planning:** Having examined the risks, it is beneficial to develop ways of reducing or eliminating their impact on the project. This may involve creating contingency plans.
* **Risk Monitoring:** Risk management is an ongoing process that requires frequent assessment of risks and analysis of the strategies put in place to minimize them. As new information becomes available, plans ought to be changed and risks should be reviewed.

Determining the risks listed in the table, as well as the likelihoods and consequences of each:

**3.**

**Case Study: Communication Challenges in a Remote Project Team**

At a tech startup that specializes in creating mobile apps for healthcare, Tullika is the project manager leading a team of software developers. The team consists of ten members, comprising developers, designers, and testers. However, when it comes to the nature of the project and the availability of resources, some of them work from different parts of the world.

Tullika initially experiences a hitch-free run with periodical online team meetings to discuss progress made, address setbacks, and planning. Nevertheless, with the progression in this project’s timeline, Tullika discerns communication challenges taking place within her team.

The absence of real-time communication between office-based employees and remote staff members is another major problem. While those working in-house can easily interact face-to-face, Tullika feels disjointed from her remote colleagues who take a long before responding to queries on time.

In addition, scheduling meetings and assigning tasks becomes difficult due to varying time zones. It means that Tullika struggles to find an appropriate time for all members’ presence to make decisions on time or do duties faster. This leads to misunderstandings, replication of work, and missing deadlines. As a result, team morale drops because members think they are isolated and unsupported. Tullika realizes the significance of communication when carrying out any project and hence decides to address this problem by taking proactive measures. This plan will include:

**Establish Clear Communication Channels:** Tullika then sets up specific channels for different tasks and talks concerning the task on hand such as Slack or email threads. Subsequently, all colleagues have access to the necessary materials and can discuss them at convenient times.

**Regular Check-ins:** In doing so, she takes care of getting connected with her remote team members at scheduled check-in meetings with updates being provided by Tullika herself while concerns are addressed. This way, it becomes easier for remote team members to feel involved in the project.

**Documenting Decisions:** Lastly, Tullika always insists on writing decisions made during meetings as well as follow-up actions. What this means is that no member is left uninformed, and everyone has something they can refer to if need be.

**Use Collaboration Tools:** Tullika encourages the use of collaboration tools, including Google Docs and Trello boards, to enhance real-time collaboration as well as the management of tasks. They make it possible for a team to collaborate effectively wherever they are.

**Encourage Transparency:** In this regard, Tullika creates an atmosphere of transparency and openness in her team where members can openly share their thoughts, worries, and criticism. This improves trust between team members thereby resulting in better teamwork.

**4.**

To deal with your manager's demand that your project team deliver software on time while considering the members who have children, there are several things that you should think about. Here are five important items to consider:

**Work-Life Balance and Team Well-Being:** The physical and mental health of your team members must be given priority. For workers, unpaid overtime can affect work-life balance, increase stress levels, and lead to burnout, particularly for those with young children requiring nurture and care. To prioritize their well-being, it may be necessary to advocate reasonable working hours for maintaining a healthy work-life balance.

**Impact on Morale and Motivation:** When employees are asked to work extra hours without pay, it harms their motivation and morale. Having to forgo personal time with loved ones to get some work done may lead to resentment as well as a decrease in job satisfaction. Low morale can affect project success, teamwork and productivity.

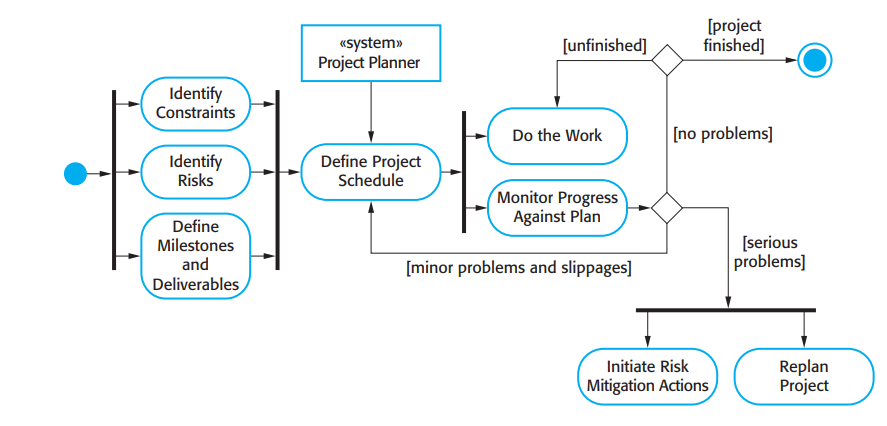
**Ethical Considerations:** Asking team members to work unpaid overtime is fraught with ethical issues, especially when those team members have family obligations. It could be considered exploitative and unfair to put pressure on workers to work longer than the agreed-upon hours without getting paid. Keeping a positive work culture requires holding onto moral values that dictate fairness between mates.

**Legal and Administrative Compliance:** Working hours, overtime pay, and employee rights are all governed by laws and regulations in some places. Requesting unpaid overtime from workers could be against labor laws and put the company at risk of fines and other legal repercussions. Ensuring adherence to pertinent labor laws and regulations is crucial for safeguarding the interests of both the organization and its workforce.

**Alternative Solutions and Mitigation Strategies:** Instead of depending solely on unpaid overtime to meet deadlines, it might be more sustainable and effective to investigate alternative solutions and mitigation strategies. Such a move could involve a re-evaluation of project schedules, resource redistribution, setting work priorities, or negotiating expectations with stakeholders. To achieve long-term success, cooperative solutions that strike a balance between the team’s needs and project requirements must be found.

# **Chapter 23: Project Planning**

**5.**



The reasons for iterating project planning are as follows:

* **Dynamicity of projects:** Software projects themselves are dynamic, with changing priorities, requirements, and unexpected problems. Therefore, project planning should adjust to these changes over time, and it cannot be done at once.
* **Development in Increments:** Incremental development is supported by various software development approaches such as Agile where software is created within short iterative cycles. The approach necessitates continuous planning and adaptation based on suggestions and amendments.
* **Uncertainty and Risk:** Every software progress carries inherent uncertainty and risk. Variables like resource availability, market dynamics, or technological breakthroughs can affect the results of a project. Early identification and mitigation of risks is possible through iterative planning processes.
* **Feedback Loop:** Project teams can obtain input on their work, identify problems, and make the required corrections by using continuous review and iteration. By using a feedback loop, the project is kept in line with its aims and objectives.
* **Resource Allocation Optimization:** Project managers can optimize resource allocation, modify timelines, and reallocate priorities in response to evolving conditions by routinely reviewing the plan. By doing this, resource efficiency is guaranteed, and project effectiveness is increased.
* **Feedback loops:** Continuous review and iteration enable project teams to seek input into their work, identify mistakes, and make the necessary adjustments. The use of a feedback loop ensures that the project is kept on track with its goals.
* **Optimization of Resource Allocation:** Regularly reviewing the plan allows project managers to optimize resource allocation, change timelines, and reallocate priorities in response to changing circumstances. This guarantees resource efficiency and boosts project effectiveness.

**6.**

A cost estimate’s risk can be reduced in four different ways:

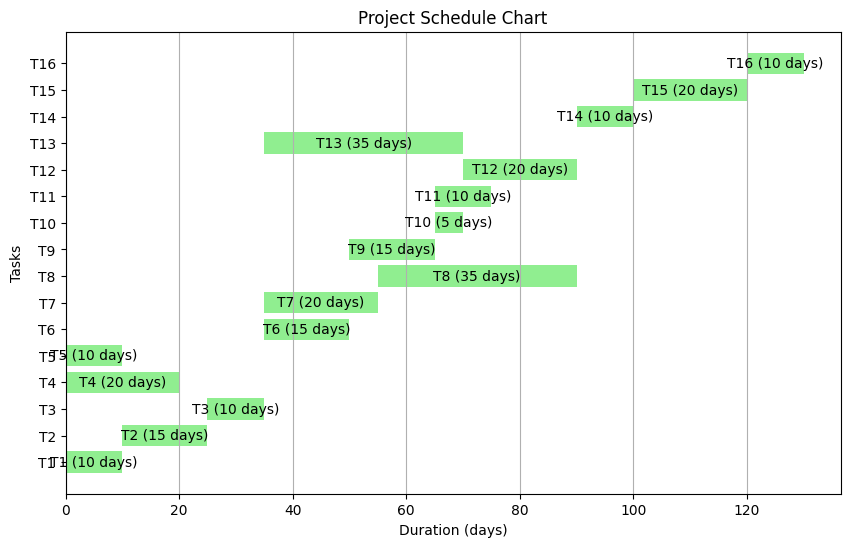
* **Market Opportunity:** A software development company could offer below-average prices for a new market niche of software to win a project and gain experience there. By accepting less profit from one job, the firm may make more money later. This approach reduces the risk by gaining critical market insights and diversifying portfolios.
* **Cost Estimate Uncertainty:** Companies with uncertainties about their cost projections might add a contingency that is greater than their normal profit margin to raise the price. This contingency accounts for unpredicted risks or changes in project scope that would otherwise result in an underestimation of costs.
* **Contractual Terms:** You can reduce your risk by negotiating useful contractual terms with your customer. For example, if developers retain ownership of source code, they may be able to charge clients less. Ensuring that code reuse brings other developers profits later, mitigates the risks involved with investment into development.
* **Volatility in Requirements:** Changes in project requirements can be included as part of cost plans that firms may develop. Businesses aiming to get a contract will lower their bid if they predict fluctuations in the specifications. The need to prevent declines in income from scope creep sometimes leads to more expensive modifications after the awarding of the tender.

**7.**

**A. Original Schedule:**

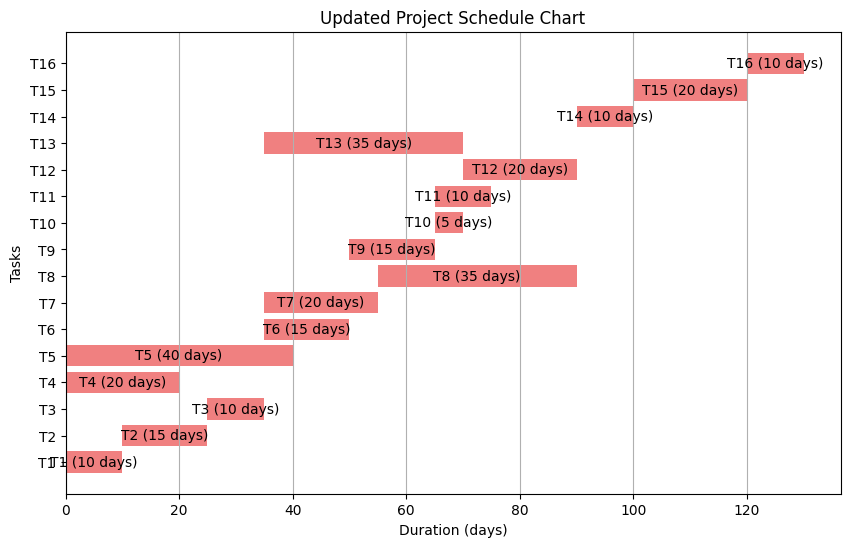
| **Task** | **Duration (days)** | **Dependencies** |
| --- | --- | --- |
| T1 | 10 |  |
| T2 | 15 | T1 |
| T3 | 10 | T1, T2 |
| T4 | 20 |  |
| T5 | 10 |  |
| T6 | 15 | T3, T4 |
| T7 | 20 | T3 |
| T8 | 35 | T7 |
| T9 | 15 | T6 |
| T10 | 5 | T5, T9 |
| T11 | 10 | T9 |
| T12 | 20 | T10 |
| T13 | 35 | T3,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

**Original Project Schedule Chart:**

Total days to complete all tasks = 155 days.

**B. Updated Schedule (with extended duration for Task T5):**

| **Task** | **Duration (days)** | **Dependencies** |
| --- | --- | --- |
| T1 | 10 |  |
| T2 | 15 | T1 |
| T3 | 10 | T1, T2 |
| T4 | 20 |  |
| T5 | 40 |  |
| 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 | T2, T14 |
| T16 | 10 | T15 |

**Updated Project Schedule Chart:**

Total days to complete all tasks = 175 days.

**8.**

**Wideband Delphi Estimation Process:**

Wideband Delphi is a technique that a team could employ to come up with an estimate. The project manager selects a group of estimators, and the estimated values are agreed upon through them. Wideband Delphi is an iterative estimation process because it consists of simple steps that can be done in the same manner every time.

1. **Choose the Team:**

* The project manager selects a mixed team of estimators containing three to seven members and appoints someone to moderate.
* The moderator should be knowledgeable about the Delphi process but not have any preference for the result.

1. **Kickoff Meeting:**

* Before they start, the team meets for the first time to appreciate the steps and review project documentation.
* During that time, assumptions are generated by the group as a whole and a work breakdown structure (WBS) is developed hence agreeing on an estimation unit.

1. **Individual Preparation:**

* In this case, all estimators on their own will calculate estimates on how much work is involved in each task of the WBS.
* Estimators also make notes about any assumptions they have made during estimation.

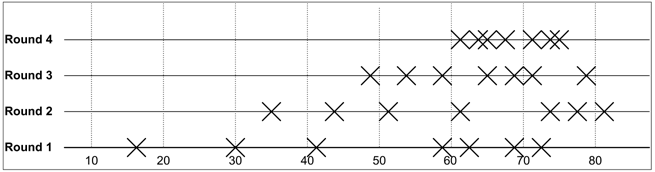
1. **Estimation Session:**

* It is during this stage that members meet to have one opinion regarding how long it would take for every task to be accomplished.
* Through rounds of discussions, estimators modify their estimates until there is agreement among them all about what it entails.

A graph with lines and numbers

Description automatically generated with medium confidence

* The moderator takes part in facilitating discussion without leaving anyone with the ability to talk about individual estimates openly.



1. **Assemble Tasks:**

* Finally, with evaluations from every estimator, the project manager comes up with a final list that contains tasks together with corresponding estimates and assumptions. Wideband Delphi is a technique that a team could employ to come up with an estimate.
* The project manager selects a group of estimators, and the estimated values are agreed upon through them.
* Wideband Delphi is an iterative estimation process because it consists of simple steps that can be done in the same manner every time.

1. **Review Results:**

* The project manager goes over the final task list with the estimation team for clarity and alignment.
* Any differences or concerns are investigated, and changes are made based on feedback.

Now, given the case:

In this process, every estimator updates his/her estimate consequently in each round by applying an average estimate and a certain percentage factor. This implies that if all percentage factors (pi) are equal for all estimators in each round, then the mean value of the estimates will not change over rounds because each estimator modifies her/his estimate by the same ratio resulting in continued maintenance of the mean.

Consider,

Let C(i) be the average estimation in round i.

Let Cj(i) be the estimation of estimator Ej in round i.

Let pi denote a percentage adjustment factor for estimator Ej.

This means that the current estimator for each estimator at round (i + 1) is just the average estimate from round i multiplied by a similar percentage p. This implies that there is constant average estimation across rounds. A white sheet with black text

Description automatically generated

**9.**

When making a revised estimate project, personnel, product, and organizational factors should be considered. The source also estimated it to take 26 person months using the COCOMO estimation model for the development of a safety-critical software system that controlled a radiotherapy machine. There are four factors described below including recommended values and explanations; they can have a significant impact on the initial estimate:

1. **Critical Safety Factor (Product Factor):**

* Factor: How important software is in managing a radiotherapy machine and ensuring that patient safety can be guaranteed.
* Value: High because when there are errors in software, it causes a disaster.
* Justification: Reasoning It takes more time to develop safety-critical systems since they require extensive testing, validation, and compliance activities.

1. **Technical Difficulties (Product Factor):**

* Factor: It becomes technically challenging to develop software for special processors with fixed memory limits.
* Value: High considering the memory and processor limitations.
* Justification: Time The programming team might need a longer duration to improve the performance of the software on specific platforms and ensure addressing memory limits are addressed.

1. **Regulatory Compliance (Organizational Factor):**

* Factor: Adherence to regulatory standards including medical devices regulation is one of the conditions that must be met for market approval.
* Value: High and in compliance with the strict legal regulations governing medical software.
* Justification: To comply with regulatory requirements such as those set by the FDA on medical devices calls for documentation, testing, and validation activities that help in the overall development effort.

1. **Personnel Capability (Personnel Factor):**

* Factor: The knowledge and experience of a development team affects project results.
* Value: Depending on the years of experience of team members, it can range from moderate to high.
* Justification: A team that has already developed safety-critical software before and knows about regulatory requirements may be able to cut down on development time as well as tackle obstacles better than other teams.

A more accurate estimate of the effort required to develop a safety-critical software system can only be reached if these aspects are considered while amending the initial COCOMO estimate. Failing to do this might result in an underestimation of resources available or even expose such projects to risks like; the possibility of delays or failure, thereby exposing patients’ lives or endangering their health due to noncompliance with statutory directives.

# **References:**

1. Sommerville, Ian. Software Engineering. Pearson, 2011.
2. Han, J. (2024). Software Project Planning and Management [PowerPoint slides, Slides 29-36]. CSC 584 - Software Project Planning and Management, California State University, Dominguez Hills. Retrieved from Canvas website. (Accessed: February 7, 2024).