Numerical left (probably will be done in copy)

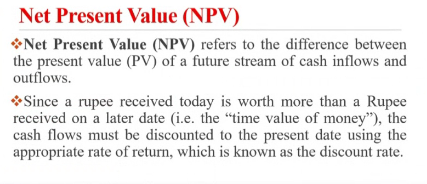
# What are different methods for comparing projects on the basis of their cash flow forecasts? Explain why discounted cash flow techniques provide better criteria for project selection than net profit or return on investment.

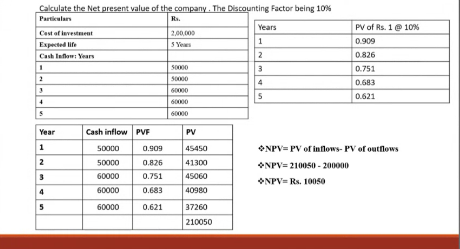
There are several methods for comparing projects based on their cash flow forecasts, including:

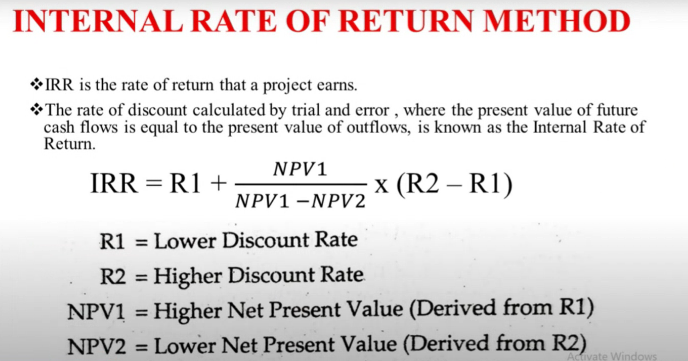
1. **Net Profit:** Net profit, also known as the bottom line, is the amount by which the total revenue of a company exceeds its total expenses. It is a simple and widely used measure of a project's financial performance, but it does not consider the time value of money or provide insights into the overall profitability of a project.
2. **Payback Period:** The payback period is the length of time required to recover the cost of an investment. It is a basic measure that is easy to understand and calculate. However, it does not account for the time value of money, and it ignores cash flows beyond the payback period, potentially overlooking the long-term profitability of a project.
3. **Return on Investment (ROI):** ROI is a financial metric that calculates the return generated on an investment relative to its cost. While widely used, ROI does not fully capture the time value of money or the variability of future cash flows, which are essential for making informed investment decisions.

ROI=Cost of Investment Net Profit Loss ​×100For example, if an investment costs $1,000 and generates a net profit of $200, the ROI would be:ROI=(200/1000)×100=20%ROI=(200/1000)×100=20%

ROI is a straightforward and intuitive measurement, but it has some limitations. First, it does not account for the time value of money; second, it does not necessarily indicate the likelihood of achieving the projected returns; third, it assumes that all costs are included in the calculation; fourth, it does not account for risk; fifth, it does not consider the effect of inflation on investment returns.

1. **Net Present Value (NPV): **

****

1. **Internal Rate of Return (IRR):** 

**Discounted Cash Flow (DCF)**

This method converts future cash flows into present values by applying a discount rate that factors in the time value of money. DCF provides a more complete picture of a project's economic merits since it accounts for the time value of money and the uncertainty surrounding future cash flows.

DCF is preferred over net profit and ROI due to its ability to:

* + Account for the time value of money, ensuring that future cash flows are valued appropriately according to their distance from the present.
  + Provide a clearer indication of a project's true economic benefits once the initial investment is recouped.
  + Allow for sensitivity analysis, enabling decision makers to explore the impacts of varying assumptions regarding future cash flows and discount rates.
  + Offer a standardized approach to comparing projects across sectors, irrespective of their size or complexity.

However, DCF is dependent upon accurate predictions of future cash flows and discount rates, and it may not always reflect the unique circumstances of certain innovative or growth-oriented projects. In such cases, alternative valuation approaches may be more suitable.

# What are the characteristics of waterfall, prototyping, incremental software process model? Explain the situation and factors of selecting an appropriate process models for a software development project.

Waterfall, prototyping, and incremental software process models are different approaches to software development. The characteristics of each model are as follows:

1. **Waterfall Model**: The waterfall model is a linear, sequential approach to software development. It involves a series of phases, including conception, initiation, analysis, design, construction, testing, implementation, and maintenance. Each phase must be completed before the next phase can begin, and once a phase is completed, developers cannot go back to a previous phase without starting the project from the beginning. The waterfall model is best suited for projects with fixed requirements, ample resources, an established timeline, well-understood technology, and unlikely to require significant changes.
2. **Prototyping Model**: The prototyping model involves creating a working model of the software before developing the final product. The prototype is used to gather feedback from users and stakeholders, which is then used to refine the final product. The prototyping model is best suited for projects with unclear or changing requirements, where user feedback is essential, and where the development team needs to experiment with different design options.
3. **Incremental Model**: The incremental model involves dividing the software development process into smaller, incremental pieces that are developed and tested individually. Each increment builds upon the previously created functionality until the software is completed. The incremental model is best suited for projects with evolving requirements, where early market entry is essential, where the deployment of new technology is underway, where the development timeline of a project is extended, and where some features and goals are high-risk.

**Factors for selecting an appropriate process model for a software development project**:

* **Requirements Stability**: If requirements are stable and well-understood, a waterfall model may be suitable. For evolving requirements, iterative models like prototyping or incremental may be more appropriate.
* **Project Size and Duration**: Larger projects may benefit from incremental development to manage complexity, while smaller projects with fixed requirements may opt for the waterfall model.
* **Risk Tolerance**: Projects with high uncertainty and risk may benefit from iterative models like prototyping or incremental, which allow for early feedback and risk mitigation.
* **Resource Availability**: The availability of resources, including time, budget, and skilled personnel, can influence the choice of process model. Some models may require more upfront planning and documentation, while others may prioritize rapid development and frequent iterations.

Ultimately, the selection of a software process model depends on the specific characteristics of the project, including its requirements, constraints, and objectives. It's important to carefully evaluate these factors and choose a model that best aligns with the project's needs and goals. Additionally, hybrid approaches that combine elements of different models can also be considered to tailor the development process to the unique requirements of the project.

# Explain the differences between modern project management practices and traditional practices.

| **Aspect** | **Traditional Project Management** | **Modern Project Management** |
| --- | --- | --- |
| **Planning** | Follows a step-by-step plan from start to finish | Plans as you go, adapting along the way |
| **Delivery** | Aiming for one big result at the end | Delivering bits of work regularly throughout |
| **Communication** | Formal and top-down | Open and everyone talks to each other |
| **Risk Management** | Addressed occasionally, separately | Constantly considering and handling risks |
| **Flexibility** | Once the plan is set, it's hard to change | Ready to adjust and change whenever needed |
| **Client Involvement** | Clients involved at major milestones | Clients involved throughout the process |
| **Team Structure** | Work is often divided by department or expertise | Teams are cross-functional, working closely together |
| **Feedback and Iteration** | Feedback is gathered at specific milestones | Continuous feedback and improvements are encouraged |
| **Documentation** | Emphasis on extensive documentation upfront | Focus on delivering working products over documentation |

# Explain the objective- or product-driven Project Characteristics with example.

**Objective-driven Project Characteristics**

Objective-driven projects are temporary endeavors aimed at reaching specific goals and ending upon achievement of predetermined objectives and milestones. Some characteristic examples include:

* Clear, specific goal and plan-driven approach
* Resourcing adjustments based on project needs
* Focus on time, budgets, and deliverables
* Performance measurement against predefined objectives and milestones
* Ends after achieving predetermined objectives

For instance, constructing a bridge would be an objective-driven project, aiming to span a river and support specific load capacities. Upon completion and verification of these objectives, the project would conclude.

**Product-driven Project Characteristics**

Product-driven projects focus on ongoing value delivery, maintaining customer satisfaction, and continuous improvement. Some characteristic examples include:

* Long-lasting teams centered around a specific product or product family
* Repetitive, incremental improvements in short cycles (e.g., two-week sprints or release cycles)
* Prioritization based on delivering the most valuable features to users
* Frequent feedback collection and implementation
* Focus on business objectives like increased adoption, revenue, or productivity gains
* Continuous learning and adaptation to market conditions

Consider a software development team working on a customer relationship management (CRM) platform. Their main focus is on enhancing the usability, scalability, and feature set of the CRM system to increase customer satisfaction and improve business performance. The team collects frequent feedback and implements changes accordingly, ensuring the CRM remains relevant and competitive in the market

| **Characteristic** | **Objective-Driven Projects** | **Product-Driven Projects** |
| --- | --- | --- |
| **Focus** | Achieving specific objectives or outcomes. | Delivering a tangible product or service. |
| **Goal Setting** | Goals defined in terms of desired results or achievements. | Goals defined in terms of delivering specific products or features. |
| **Flexibility** | May prioritize adaptability to changing objectives. | Adaptability may be secondary to delivering the product as specified. |
| **Measurement of Success** | Success measured by the achievement of objectives. | Success measured by the completion and quality of the product. |
| **Examples** | Launching a marketing campaign to increase brand awareness. Implementing a new training program to improve employee skills. | Developing a mobile application with specific features. Constructing a building according to architectural plans. |

# What do you understand by software configuration? What is meant by software configuration management? Why is software configuration management crucial to the success of large software product development projects?

Software configuration refers to the process of identifying, organizing, and controlling changes to the software system throughout its lifecycle. Software Configuration Management (SCM) is the discipline of managing and controlling changes to software configurations effectively. It involves establishing and maintaining a consistent and stable software baseline, as well as managing changes, versions, releases, and documentation associated with the software product. SCM ensures that the software development process is systematic, controlled, and traceable.

SCM is crucial to the success of large software product development projects because it helps to ensure that the software is developed and delivered on time, within budget, and with the desired quality. SCM helps to manage the software development process by providing a framework for managing changes to the software system, including source code, documentation, and other artifacts. SCM also helps to ensure that the software is developed in a consistent manner, which helps to improve the quality of the software and reduce the risk of errors and defects.

In summary, software configuration management is a process that helps to ensure that software is developed and delivered on time, within budget, and with the desired quality. SCM provides a structured approach to software development, which helps to manage changes to the software system and ensure that the software is developed in a consistent and predictable manner. SCM is crucial to the success of large software product development projects because it helps to reduce the risk of errors and defects and improve the quality of the software.

# As a project manager, identify the characteristics that you would look for in a software developer while trying to select personnel for your team.

As a project manager, there are several characteristics that I would look for in a software developer while trying to select personnel for my team. Some of these characteristics include:

1. Technical expertise: A software developer should have a strong technical background and be proficient in programming languages, software development tools, and technologies.
2. Problem-solving skills: A software developer should be able to identify and solve problems quickly and efficiently. They should be able to analyze complex issues and develop creative solutions.
3. Attention to detail: A software developer should have a keen eye for detail and be able to identify and correct errors in code.
4. Communication skills: A software developer should be able to communicate effectively with team members, stakeholders, and clients. They should be able to explain technical concepts in a clear and concise manner.
5. Team player: A software developer should be able to work collaboratively with other team members and be willing to share knowledge and expertise.
6. Adaptability: A software developer should be able to adapt to changing project requirements and be willing to learn new technologies and tools.
7. Positive attitude: A software developer should have a positive attitude towards work and be willing to take on new challenges.
8. Time management skills: A software developer should be able to manage their time effectively and meet project deadlines.
9. Passion for software development: A software developer should have a passion for software development and be committed to delivering high-quality software products.
10. Continuous learning: A software developer should be willing to continuously learn and improve their skills to stay up-to-date with the latest technologies and trends in software development.

In summary, selecting the right personnel for a software development project is crucial to its success. As a project manager, I would look for software developers who possess the above characteristics to ensure that the project is completed on time, within budget, and with the desired quality.

# Explain the advantages of a functional organization over a project organization. Also explain why software development houses prefer to use project organization over functional organization.

Functional organization is a type of organizational structure where employees are grouped based on their areas of expertise and functions. Each department is managed by a functional unit head who reports to the strategic leadership of the organization. In this structure, project coordination typically happens at the functional management level, and staff is allocated from units only as per the requirements of the project.

Project organization, on the other hand, is a type of organizational structure where a temporary team is assembled using team members from different functions once the project begins.

Advantages of a functional organization over a project organization include:

1. Specialization: Functional organizations group employees based on their areas of expertise, allowing for a higher level of skill development and efficiency within each functional area.
2. Clear hierarchy and reporting lines: Reporting lines are typically clear and straightforward in functional organizations, reducing confusion and improving accountability.
3. Efficiency and cost savings: Functional organizations often result in cost efficiencies because resources and personnel are grouped by function, reducing redundancy and eliminating the need for duplicated roles or skills.
4. Focused training and development: Employees within each functional area can receive specialized training and development tailored to their roles, enhancing their skills and capabilities.
5. Streamlined decision-making: Decision-making is often centralized within each functional area, making it faster and more focused on specific expertise.
6. Standardized processes: Functional organizations can lead to standardized processes and procedures within each department, which can improve quality control and consistency.
7. Clear career paths: Employees in functional organizations often have clear career progression paths within their respective departments, which can improve job satisfaction and retention.

Software development houses prefer to use project organization over functional organization because software development is a project-based activity that requires a team to work together to achieve a specific goal. Project organizations are designed to manage projects and provide a framework for managing changes to the software system, including source code, documentation, and other artifacts. Project organizations are more flexible and adaptable to changing project requirements, allowing teams to work collaboratively and respond quickly to changes in the project scope or timeline. Project organizations also provide a clear structure for managing project resources, budgets, and timelines, which is essential for delivering software projects on time, within budget, and with the desired quality.

# What is Contract? Explain bespoke, off-the-shelf package, COTS software. In each of the following cases, discuss whether the type of application package to be adopted would be most likely to be bespoke, off-the-shelf or COTS.

a. A college requires a student fees application. It is suggested that the processes required in the application are similar to those of any billing system, with some requirements that are peculiar to the administration of higher education.

b. A computer-based application is needed to hold personnel details of staff employed.

c. A system is required by a national government that calculates, records and notifies individual taxpayers about income tax charges.

d. An expert system for use in a hospital to diagnose the causes of eye complaints.

A contract is a legally binding agreement between two or more parties that outlines the terms and conditions governing their relationship. Contracts can cover a wide range of transactions, including sales of goods or services, employment agreements, leases, and more. In the context of software development or procurement, contracts define the rights, obligations, and responsibilities of the parties involved in the development, licensing, or purchase of software.

Now, let's explain the terms bespoke, off-the-shelf package, and COTS software:

1. **Bespoke Software**:
   * Bespoke software, also known as custom software, is software that is specifically developed to meet the unique requirements of a single customer or organization.
   * Characteristics:
     + Tailored to the specific needs and preferences of the customer.
     + Developed from scratch based on detailed specifications and requirements provided by the customer.
     + Offers full customization and flexibility to accommodate unique business processes or workflows.
   * Example: A company hires a software development firm to build a custom customer relationship management (CRM) system that integrates seamlessly with its existing sales processes and data infrastructure.
2. **Off-the-Shelf Package**:
   * Off-the-shelf (OTS) software refers to pre-packaged software products that are commercially available and ready to use without the need for customization.
   * Characteristics:
     + Developed by software vendors to address common business needs or industry requirements.
     + Sold as a standard product with predefined features, functionalities, and configurations.
     + Typically comes with documentation, user support, and maintenance services.
   * Example: Microsoft Office Suite, Adobe Photoshop, or QuickBooks accounting software.
3. **COTS Software** (Commercial Off-The-Shelf):
   * COTS software is a subset of off-the-shelf software that is specifically designed for commercial use and is available for purchase or licensing from third-party vendors.
   * Characteristics:
     + Designed to be used by multiple customers across different industries or sectors.
     + Offers standard features and functionalities that meet common business requirements.
     + May require minimal configuration or customization to adapt to specific organizational needs.
   * Example: Enterprise resource planning (ERP) systems like SAP or Oracle, customer relationship management (CRM) software like Salesforce, or project management tools like Microsoft Project.

Based on the given scenarios:

a. Student Fees Application  
Most suitable: Bespoke Software  
Reasons: The application requires unique functionality tailored to the specific administration of higher education. Although there could be generic student enrollment applications, they may not cover all aspects of student fees management.

b. Personnel Details Management System  
Most suitable: Off-the-Shelf (COTS) Software  
Reasons: Managing personnel details is a common requirement across organizations, so there are readily available HR management systems that can fulfill this need.

c. Tax Calculation and Notification System  
Most suitable: Customizable Package  
Reasons: Although there could be generic tax calculation tools, they may not have built-in notification mechanisms. Therefore, implementing a customizable package like a CRM platform with custom modules and integrations would be ideal.

d. An expert system for use in a hospital  
Most suitable: , bespoke software

Reasons: The system needs to be tailored to the specific requirements of the hospital and its patients, and it needs to be able to diagnose a wide range of eye diseases based on the symptoms presented by the patient.

In summary, bespoke software is recommended for highly specific applications, off-the-shelf (COTS) software for common requirements, and customizable packages for meeting specific needs while leveraging existing technology.

# What do you understand by the six sigma quality initiative? Explain the six sigma technique with respect to its goal, the procedure followed, and the outcome expected.

Six Sigma is a quality management methodology aimed at improving processes by reducing defects and variations to achieve near-perfect levels of quality. Developed by Motorola in the 1980s and popularized by companies like General Electric, Six Sigma focuses on data-driven decision-making, process optimization, and continuous improvement..

**Goal:**

The goal of Six Sigma is to identify and eliminate defects or errors in processes to achieve a level of performance where there are only 3.4 defects per million opportunities (DPMO). This equates to achieving a process capability of 6 sigma, which represents an extremely high level of accuracy and precision.

**Procedure Followed:**

1. **Define**: Clearly define the problem or process that needs improvement, along with the project goals and objectives. Identify key stakeholders and customers' requirements.
2. **Measure**: Measure and analyze current process performance using data and statistical tools to identify defects and variations. Determine process capability and establish a baseline for improvement.
3. **Analyze**: Analyze the root causes of defects and variations in the process. Use statistical methods and tools such as Pareto charts, cause-and-effect diagrams, and hypothesis testing to identify contributing factors.
4. **Improve**: Develop and implement solutions to address root causes and improve process performance. Test and validate improvement initiatives using pilot projects or experiments.
5. **Control**: Establish controls and monitoring mechanisms to sustain process improvements over time. Implement measures to prevent regression and ensure ongoing performance.

**Outcome Expected:**

The expected outcome of implementing Six Sigma techniques is to achieve significant improvements in process efficiency, quality, and customer satisfaction. By reducing defects and variations, organizations can enhance product or service quality, reduce costs, increase productivity, and gain a competitive edge in the marketplace. Additionally, Six Sigma fosters a culture of continuous improvement and data-driven decision-making within the organization.

Overall, Six Sigma aims to drive systematic and sustainable improvements in processes, leading to higher levels of customer satisfaction, operational excellence, and business success.