Unit I

1 Introduction to Software Project Management

Syllabus

Why is Software Project Management important ? What is a Project ? Contract Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some Ways of Categorizing Software Projects, Stakeholders, Setting Objectives, Business Case, Project Success and Failure, what is Management ? Management Control, Traditional versus Modern Project Management Practices.

Case study : Online Shopping System.

Contents

1.1 Software Project Management

Introduction :

The first concern is whether managing software projects truly differs from managing other types of projects :

- We must examine certain fundamental concepts regarding the scheduling, observing and controlling of software projects in order to provide a response. We will see that achieving goals is the main goal of every project. A software project must meet actual needs, just like any other project. We must first identify the project's stakeholders and their goals in order to do this. The goal of project management is to make sure that their goals are achieved. However, until we are aware of a project's current position, we cannot predict if it will achieve its goals in the future.
- **Definition :** A software project is the entire process of software development, from requirement collecting to testing and maintenance, that is carried out in a specific time frame to produce the desired software product, according to the execution techniques.

1.1.1 Need of Software Project Management

1. Software is often referred to as an intangible asset. Software development is a relatively young field in the world of business, with little expertise in the creation of software products. The majority of software solutions are developed to order in order to meet the specific needs of the client.
2. The most crucial is that the underlying technology evolves and changes at such a rapid pace that experience gained from one product may not be applicable to another. All of these commercial and environmental restrictions put software development at risk, thus it's critical to manage software projects effectively.

Fig. 1.1.1 : Software project constraints

3. For software projects, the Fig. 1.1.1 depicts triple limitations. It is an important component of software development to create a high-quality product while staying within the price constraints of the client and completing the project on time. This triple constraint triangle may be influenced by a number of internal and external factors. Any one of three factors can have a significant impact on the other two. As a result, software project management is critical for incorporating user needs as well as budget and time restrictions.

1.1.2 Why is Software Project Management Important ?

- Students who are more technically oriented may become impatient when they have to study material that diverts their attention from writing code. So why is it important ?

1. The first issue is the one involving money. The financial viability of ICT projects is crucial. The federal government of the United Kingdom spent more money on contracts for ICT projects than on contracts for road construction during the fiscal year 2002 - 2003. The department for work and pensions was the department with the highest ICT spending, spending more than £800 million. ICT projects that are poorly managed result in less money being spent on beneficial things like hospitals.
2. Projects unfortunately don't always turn out well. In a 2003 analysis, the Standish Group in the US examined 13,522 projects and came to the conclusion that just 33 % of them were successful, 82 % of them were late and 43 % went over budget.
3. The management of projects is frequently to blame for these project flaws. For instance, the National Audit Office in the UK identified "lack of skills and proven approach to project management and risk management" as one of the causes of project failure.

1.1.3 Project

1.1.3.1 What is Project ?

Definition : A project is a series of distinctive, intricate and interconnected tasks with a single objective that must be finished on schedule, on budget and in accordance with specifications.

- Project is a precise design or plan, a well-planned project, A significant undertaking .
- The focus on planning suggests that we can figure out how to complete a task before we begin. However, this may be challenging in exploratory projects. Even with uncertain tasks, planning entails thinking hard about something before doing it and as long as the resulting plans are viewed as provisional, this is worthwhile.

- Other tasks, such as routine maintenance, will have been completed so many times that everyone is familiar with the procedure. Planning appears to be unnecessary in these situations, while processes may be written to guarantee consistency and to assist newcomers.
- The activities that gain the most from traditional project management are likely to fall somewhere in the middle shown in the Fig. 1.1.2.

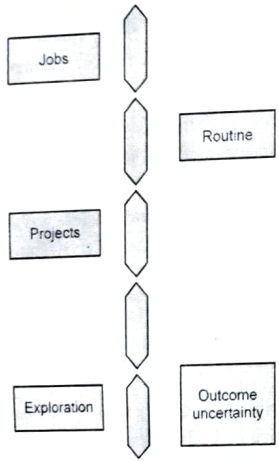

Fig. 1.1.2 : Project management activities that are most likely to benefit

- The line between a non-routine project and a routine job is blurry. It will feel like a project the first time we conduct a routine task. A project to construct a system similar to past ones that we have developed, on the other hand, will include a significant element of routine.
- The characteristics listed below distinguish projects.
  1. Tasks that are not routine are involved
  2. Preparation is required
  3. Certain goals must be accomplished or a specific product must be generated
  4. The project has a deadline
  5. Work is performed for someone other than ourself

6. Work entails multiple specialties
7. To complete the assignment, people are organized into a temporary work group
8. The work is carried out in stages
9. The amount of resources accessible for usage on the project is limited
10. It's a big or complicated project

- The more of these elements that apply to an activity, the more complex it becomes. The magnitude of the project is extremely essential. Because of the requirement for additional coordination, a project with 20 developers is likely to be disproportionately more challenging than one with only ten.

1.1.3.2 Software Projects Versus Other Types of Project

- Any project management principles that apply to general projects also apply to software projects, however Fred Brooks recognized specific peculiarities of software projects that make them particularly challenging.
- **Invisibility :** The progress of a physical item, such as a bridge, can be observed. Progress in software isn't always apparent. The process of making the unseen visible can be viewed as software project management.
- **Complexity :** Software goods are more sophisticated than other constructed artifacts for every dollar, pound or euro spent.
- **Conformity :** Any project management principles that apply to general projects also apply to software projects, however Fred Brooks recognized specific peculiarities of software projects that make them particularly challenging. Individuals aren't the only ones who can be unreliable. Organizations can display significant organizational stupidity as a result of gaps in collective memory, internal communication or effective decision making.
- **Flexibility :** It is considered a strength that software is simple to change. When a software system interacts with a physical or organizational system, the software is expected to adjust to accommodate the other components rather than the other way around. As a result, software systems are especially vulnerable to change.

Review Questions

1. What is project ?
2. What is the need for software project management ?
3. Differentiate between software projects versus other types of projects.

1.2 Contract Management

- In-house projects take place when both the users and the programmers of new software are employed by the same company.
- However, more and more businesses hire outside developers to develop their ICT. In this situation, the client organization will frequently choose a "project manager" to oversee the contract and will hand off many decisions with a technical focus to the contractors. Therefore, as long as the whole project is completed on schedule and under budget, the project manager won't have to worry about calculating the time required to build individual software components. Project managers who deal with the more technical difficulties will be required on the supplier side.

1.3 Activities Covered by Software Project Management

- The creation of software is simply one aspect of a software project. In reality, even when a software programme is purchased "off the shelf," even while there may not be any software writing specifically involved, a software project is still being undertaken because so many of the other activities related to software development will still be taking place.
- Typically, a new system is created through three successive processes, shown in the Fig. 1.3.1 :

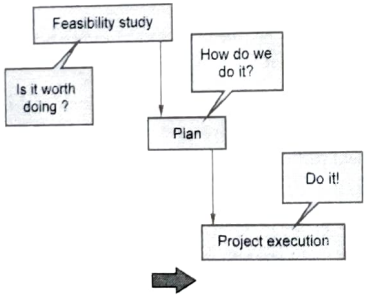

Fig. 1.3.1 : Feasibility, study / plan / execution cycle

1. The **feasibility study** determines whether a project has a strong business case and is worthwhile to pursue. Information is acquired about the application's suggested requirements. At least initially, requirements elicitation can be complex and challenging. The stakeholders could be clear on the goals they want to pursue but uncertain on the ways to get there. It will also be necessary to assess the price of the new system's

benefits as well as its operations and development costs. The feasibility study might be a separate project with its own concept for a big system. The research might be used in a strategic planning exercise that looks at several possible software improvements. An organization may occasionally evaluate a development programme composed of several projects.

2. **Project execution** Now the job can be carried out. A project's execution process frequently includes separate design and implementation phases. The distinction between design and planning can frequently be blurry for students who are new to project planning. Making choices concerning the shape of the things that will be produced is called design. This may have to do with the user interface, which is the software's visible look or the internal design. The plan outlines the tasks that must be completed to produce these goods. Because design decisions are influenced by planning decisions at the most granular level, the terms planning and design are sometimes used interchangeably. Therefore, it is likely that five sets of operations will be needed to produce a software product having five primary components.
3. **Planning Project** planning can begin if the feasibility analysis concludes that the potential project appears viable. We wouldn't start all of our in-depth planning for bigger projects at the beginning. We drew a broad project plan and a more specific one for the initial phase. Planning for the later stages of the project is deferred until closer to its beginning because we won't have access to more comprehensive and accurate project information until the earlier stages have been finished.

1.3.1 Sequence of Software Development Activities

- The international standard ISO 12207 recommends the typical order of software development operations, which is depicted in Fig. 1.3.2. While some operations relate to software, others are system-related. Software development will only be a small portion of the whole project. For instance, software could be created for a project that includes calls for the setup of an ICT infrastructure, the creation of user jobs and user training.

1. **Architecture design,** It is necessary to determine which parts of the new system satisfy each requirement. Some needs may be met by already-existing components. Other times, a brand-new component will need to be created. These elements could be brand-new hardware or operational procedures as well as software. Although software components are the main focus of software engineers, it is extremely uncommon for these to be created independently. They will, for instance, need to consider the legacy

systems that are already in place with which they will communicate. Thus, the requirements for the software are influenced by the system architectural design. The software requirements are subsequently translated into software components during a subsequent architecture design phase.

2. **Requirement analysis**, also known as requirements collecting, is the first step in the requirements analysis process. It specifies what the future users and their managers expect from the new system. It might have to do with a task that the system is supposed to perform. It might refer to how well the functions must operate as a quality criteria. Sending an ambulance in response to an emergency phone call is an illustration of this. In this scenario, both the efficiency of the hardware and software and the speed of user input would have an impact on transaction time. A system requirement for the project, as opposed to a software-specific requirement, is training to guarantee that operators use the computer system effectively. Additionally, there would be resource needs related to the price of application development.
3. **Detailed design**. Numerous software pieces that can be independently created and tested make up each software component. These unit's intricate designs are created independently.
4. **Writing code** for each software unit is referred to as "code and test." At this point, first testing would be done to debug certain software units.
5. **The integration,** To determine whether they satisfy the standards as a whole, the components are tested together. Combining several software components might constitute integration, as could integrating and testing the software component of the system with the hardware platforms and user interfaces.
6. **Qualification testing**. To make sure that all the requirements have been met, the system, including the software components, must be thoroughly tested.
7. **Installation,** This is the procedure for turning on the new system. Setting up standing data (for instance, employee information in a payroll system), establishing system parameters, installing the software on the hardware platforms and user training would all fall within this category.
8. **Acceptance support,** This entails fixing any flaws in the freshly installed system, as well as putting agreed-upon expansions and upgrades into effect. We might think of software maintenance as a collection of little software initiatives. In many situations, maintenance constitutes the majority of software development.

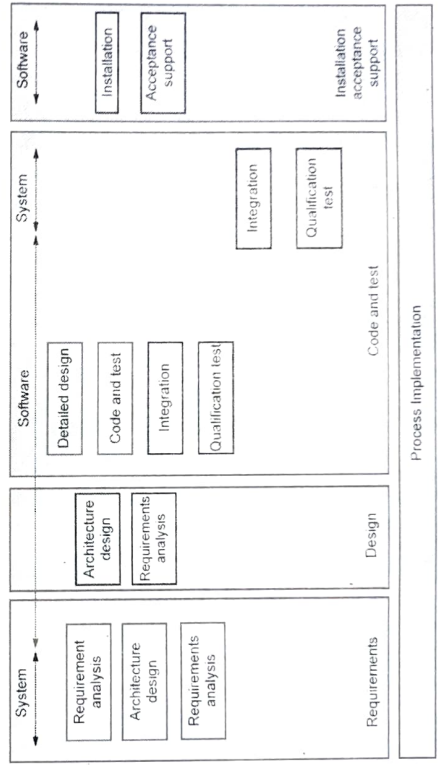

Fig. 1.3.2 : The ISO 12207 software development life cycle

Review Questions

1. Write a short note on contract management.
2. What are the activities covered by software project management ?
3. Explain sequence of software development activities in detail.

1.4 Plans, Methods and Methodologies

- An idea of a working approach must be the foundation of any plan for an activity. For instance, even if we had no prior knowledge of the software that was to be evaluated, we might assume that we would need to :
  - Examine the software's requirements
  - Devise and write test cases to ensure that every requirement has been met
  - Develop test scripts and anticipated outcomes for every test case
  - Compare the disparities between the actual and anticipated results
- A plan translates a method (and possibly others) into actual actions while a method pertains to a type of activity generally. It does this by identifying the following for each activity :
  - its beginning and ending times;
  - who will execute it;
  - what equipment, supplies and knowledge will be required.
- One method's output could serve as the input for another. Techniques or method groups are frequently categorized into methodologies, such object-oriented design.

Review Question

1. Write a note on : Plan, Methods and methodologies.

1.5 Some Ways of Categorizing Software Projects

- Due to the diverse technical items that will be produced, projects may vary. Therefore, we must determine the qualities of a project that might influence how it should be designed and handled. Below is a discussion of additional factors.

1. Compulsory versus voluntary users

- Staff must use certain technologies in the workplace in order to perform tasks, such as registering a transaction. However, as with computer games, system use is becoming more and more optional. As we could with a commercial system, it is challenging to extract specific criteria from potential users in this situation. Thus, the success of the game will be greatly influenced by the creative acumen of the designers as well as tools like market research, focus groups and prototype review.

2. Information systems versus embedded systems

- A traditional distinction has been made between embedded systems, which control machinery and information systems, which allow people to do office processes. An information system would be a stock control system. A building's air conditioning

system might be managed by an embedded or process control, system. Some systems may combine both features; for instance, a stock control system may also operate an automated warehouse.

3. Outsourced projects

- Sometimes it makes good commercial sense for a business to contract out some of its work to other companies while developing a large project. This choice may have been made for a number of reasons. For instance, a business may think outsourcing is a good idea if it believes it has the necessary skills to develop some particular product components or if it finds that other businesses can develop some components more affordably. An outsourced project is typically small in size and has a short turnaround time because it is only a small portion of a larger project. Given these distinctions between an outsourced project and a regular project, managing an outsourcing project presents unique difficulties.
- Indian software companies have established a solid reputation in the world for successfully completing outsourced software projects. Recently, the Indian businesses have gradually started to concentrate on product development as well. A company's profitability may be impacted by the kind of development work it handles. For instance, a business that creates generic software typically receives an ongoing stream of income over a number of years. Outsourced projects, however, only generate one-time money for any given organization.

4. Objective - driven development

- Whether a project's goal is to create a product or to accomplish a specific goal can help to distinguish it from other projects.
- A project could involve developing a product based on client-specified requirements. It is the client's responsibility to defend the purchase.
- On the other hand, the project need to be to achieve a set of goals that can be accomplished in a variety of methods. A company could request a solution from a professional when they have an issue.
- Two stages are common in software projects. An objectively driven initiative that produces recommendations comes first.
- This could point out the requirement for fresh software. The effort to actually produce the software is the next stage.
- When the technical work is being done by an external group and the first user demands are not evident, this is helpful. For a set price, the outside company can provide a

preliminary design. The price for the second step, implementation, can then be quoted by the developers if the design is deemed acceptable based on an established requirement.

Review Question

1. How to categorize software projects ? Explain in detail.

1.6 Stakeholders

- Stakeholders are those with a stake or an interest in the undertaking. It's crucial to identify them as soon as possible because we will need to open up effective lines of communication with them. Stakeholders can be divided into the following categories :
- This implies that the project manager will have full managerial control over them :
  1. **Internal to the project team :** This indicates that the project manager will have full managerial control over them.
  2. **External to the project team,** however, within the same company To test the system, for instance, the project manager might need user assistance. In this case, the parties concerned must negotiate their level of commitment.
  3. **External to both the project team and the organization Customers (or users)** who will benefit from the system the project deploys could be considered external stakeholders. They might be independent contractors who will work on the project. Here, the partnership is frequently established via a contract.
- One of the responsibilities of the project leader is to identify these various interests and be able to reconcile them. Different stakeholder types may have different aims. For instance, end users might be worried about the new application's usability while their managers might be more interested in personal savings. Therefore, the project manager must be a skilled communicator and negotiator. According to the "Theory W" of software project management put out by Boehm and Ross, the manager should focus on setting up scenarios in which all parties stand to gain from the project and hence have a stake in its success. (The "W" represents "win-win").
- Project managers occasionally fail to identify a crucial stakeholder group, particularly in unfamiliar business environments. These can be divisions that provide crucial services that are taken for granted.
- It is advised practice for a communication strategy to be prepared at the beginning of a project due to the significance of coordinating the efforts of stakeholders.

Review Questions

1. Write a short note on stakeholders.
2. What are different types of stakeholders ? Explain in detail.

1.7 Setting Objectives

- The objectives should specify what the project team needs to accomplish to be successful. The project objectives define the agreed aims for the project even while different stakeholders have different reasons.
- Instead than concentrating on the tasks involved in a project, objectives concentrate on the desired outcomes of the project; they are the project's "post-conditions." The project will be a success if... might be followed by a series of statements to form the objectives informally. In place of " to establish an e-commerce website, " one of a set of objectives can read, " Customers can order our products online." There are frequently multiple ways to accomplish a goal, therefore having as many as feasible is beneficial.
- There may be a number of stakeholders with interests in the project, including users from various business sectors. In such a scenario, a project authority that has overall control over the project must be identified expressly.
- This authority frequently consists of a project steering committee, project board or project management board, which is in charge of creating, monitoring and changing goals on a broad scale. Daily project management is handled by the project manager, who also regularly updates the steering committee.

1.7.1 Sub-objectives and Goals

- An individual's controllable goal must be something they can do themselves in order to be effective. One goal might be to have the software application generated reduce staff costs to the point where it pays for itself. This may make sense as a general corporate goal. It would not be realistic for software developers because any decrease in operational staff costs depends not only on them but also on the operational management of the finished system. The software developers' more appropriate goal or sub-objective should be to maintain development expenses within a predetermined limit.
- We can say that in order to accomplish the main aim, we must first accomplish some secondary goals. Similar to how goals scored in a football game are stages toward the aim of winning the game, these are steps toward reaching a target. This can be said informally

as a series of clauses that are followed by the phrase "To reach aim..., the following must be in place.

- Well-defined goals are also referred to by the acronym SMART :

1. **Specific effective** goals are specific and clearly outlined. Ambitions that are too general, such as "to improve customer relations," are unacceptable. The project's success should be made clear to everyone by clearly defining the objectives.
2. **Measurable measures** of efficacy that let us know how successful the initiative has been are ideal. An objective like "to reduce customer complaints" would be preferable to "to improve customer relations," for instance. In certain circumstances, the measure can be a simple yes/no response, such as "Did we deploy the new software by 1 June ?"
3. **Achievable** The ability of the person or organization to accomplish the goal must be considered.
4. **Relevant** The goal must be pertinent to the project's actual goal.
5. **Time** constrained There should be a set deadline by which the goal must have been accomplished.

1.7.2 Measures of Effectiveness

- Effectiveness metrics offer doable ways to verify whether a goal has been attained. For example, the "mean time between failures" (mtbf) metric may be used to gauge reliability. This can only be done once the system is up and running because it is a performance measurement. The performance of the finished system should be estimated by project managers as the system is being built. Therefore, project managers will look for prognostic indicators. For instance, a high volume of mistakes discovered during code inspections may be a sign of future reliability issues.

Review Questions

1. Write a short note on SMART: objective.
2. What do you mean by Sub-objectives and goals ?
3. How to set objectives of the project ?
4. Define performance measurement.

1.8 The Business Case

- The feasibility assessment for the project should establish the business case.

- **Most projects need to have a justification or business case :** The benefits that will eventually be felt must be recognized as making the work and price of advancing the project worthwhile. A cost-benefit analysis will frequently be included in the feasibility assessment for the project. The expenses and advantages of the project will then be itemized and quantified. The benefits will depend on when the project is finished; the sooner the project is finished, the sooner the benefits will be felt. A business model that shows how the new application can produce the promised benefits must frequently be developed in order to quantify the benefits.
- A simple example of a business model is the possibility that a new web-based application will enable clients from all over the world to order a company's goods online, increasing sales and, consequently, revenue and profits.
- The business case must be maintained according to any project plan. For instance :
  - That development expenses are not permitted to increase to a point where they might threaten to outweigh advantages;
  - That the system's features are not scaled back to the point where the anticipated advantages cannot be realized;
  - That there is no unacceptable loss of benefits due to a delay in delivery.

Review Questions

1. Write a note on business case.
2. How to justify projects ?

1.9 Project Success and Failure

- By maintaining the project's business case, the project plan should be created to guarantee project success. However, every non-trivial project will encounter issues; at what point can we declare a project to be a failure ? Some stakeholders in a project may view it as a success while others do not because they have different interests.
- The distinction between project objectives and business objectives can be made generally. The targets that the project team is required to meet are the project objectives.
- Software initiatives typically fall under the heading of delivering the following :
  1. the agreed functionality
  2. to the required level of quality
  3. on time
  4. within budget

- A project might achieve these goals, but after the application is delivered, it might not support the business case. Even if a computer game is produced on schedule and under budget, it might not be commercially successful. Although a successful commercial website for online sales could be developed, clients might not use it to make purchases as they could do it elsewhere at a lower price.
- As we have seen, a project is considered successful in commercial terms if the value of its benefits outweighs its expenses. We have also shown that, despite the fact that project managers have a great deal of control over development expenses, the value of the project's deliverables depends on outside variables like the quantity of clients. Project goals still have an impact on future business success. The duration of time during which benefits can be realized is shortened and the project's value is reduced by a completion delay.
- A project may be successful after it is completed but fails commercially. The deliverables of a project, however, could nevertheless produce benefits over time that outweigh the initial investment even if it is late and over budget.
- Some argue that taking a broader view of projects that incorporates business considerations might help close any potential gap between project and business concerns. An e-commerce website implementation project, for instance, could schedule activities like market research, competitor analysis, focus groups, prototyping and evaluation by typical potential users, all of which are intended to lower business risks.
- It may not be obvious that a project is actually one of a sequence because project management naturally focuses on the current project. The technological expertise acquired on past projects helps later projects. Early initiatives will incur higher expenses as a result of technical learning, while later projects will profit from the faster, less expensive and more accurate deployment of the learned technology. Additional software assets, such reusable code, are frequently included with this expertise. Although there may be short-term cost reductions when software development is outsourced, these longer-term advantages of increasing knowledge will be lost. Smart managers can determine which technological specialties should be developed.
- Additionally, customer relationships might develop across a number of initiatives. A client is more inclined to use a supplier they trust if they have performed quality work in the past, especially if the new request expands on previously given capability. Compared to keeping current clients, gaining new ones is far more expensive.

Review Questions

1. What is project success and failure ?
2. Distinguish between project objectives and business objectives.

1.10 Management

What is Management ?

- According to certain theories, management entails the following tasks :
  - **Planning** - deciding what has to be done through planning;
  - **Organizing** - establishing plans;
  - **Directing** - the act of offering directions;
  - **Monitoring** - keeping track of developments;
  - **Controlling** - taking initiative to address delays;
  - **Innovating** - which involves creating fresh approaches;
  - **Representing** - liaising with customers, users, the developer, suppliers and other stakeholders.
- Only three of the eight listed activities - project planning, monitoring and control - take up the majority of the project manager's time. Fig. 1.10.1 shows the time frame in which these tasks are conducted. It demonstrates that, regardless of the technique employed, project management is conducted through three clearly defined stages or processes. An initial plan is created during the project initiation stage. When a project begins, it is watched over and managed to ensure that everything goes according to plan. However, as new information and restrictions concerning the project become known, the initial plan is periodically updated to take these into account. The project is finally finished. All operations are logically finished during the project ending stage and all contracts are properly closed.

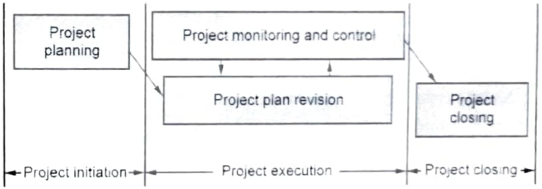

Fig. 1.10.1 : Principal project management processes

- Immediately following the feasibility study stage and before beginning the requirements analysis and specification process is the initial project planning step. The beginning of the project is depicted in Fig. 1.10.1. Estimating various project attributes is a necessary step in the initial planning process. All further project actions are scheduled based on these estimates. As the project advances and additional project data becomes available, the initial

project plans are periodically altered. As soon as the project execution begins, monitoring and control tasks are undertaken to guarantee that everything goes according to schedule. The monitoring activity entails keeping track of the project's advancement. To reduce any considerable deviation from the plan, control activities are started.

- The project manager has a crucial duty: project planning. The project manager must do a few clearly defined tasks during project planning, which are listed below.
- For the planning activities of software projects, a number of best practices have been suggested.
  - **Estimation -** The ensuing project characteristics are projected.
  - **Cost** - How much will it cost to finish the project ?
  - **Duration** - How long will it take to finish the project ?
  - **Effort -** How much work would be required to finish the project ?
- The effectiveness of all operations, even those that are planned later and involve scheduling and personnel, depends on how precisely the aforementioned three project criteria have been estimated.
  - **Scheduling -** Schedules for personnel and other resources are created based on estimates of effort and time.
  - **Staffing -** Plans are established for staffing and staff organization.
  - **Risk management** - Identification, analysis and planning for risk mitigation are all included in this process.
  - **Miscellaneous plans -** Making several other plans, including quality assurance and configuration management plans, is part of this.
- After the start of development activities, project monitoring and control activities are carried out. Making ensuring that software development continues according to plan is the goal of project monitoring and control operations. A project manager may occasionally discover that it is required to modify the plan while carrying out project monitoring and control tasks in order to deal with particular scenarios and make the plan more accurate as new project data becomes available.
- The project manager does not have complete knowledge of the project's specifics at the beginning of the project. The project manager's knowledge base continually expands as it moves through various development phases. Different project activities' complexity is revealed, some expected risks are mitigated and other hazards emerge. The project parameters are periodically re-estimated to account for new knowledge and parameter

changes. The project manager can plan subsequent tasks more confidently and correctly by taking these developments into consideration. This feature is depicted in Fig. 1.10.1 as iterations between the monitoring, control and plan revision processes.

1.11 Management Control

- Setting goals for a system and then tracking its success are two general aspects of management. The "actual world" is depicted in Fig. 1.11.1 as being largely formless. There will be a lot going on, especially in the case of huge projects, of which management should be aware.
- The local managers will be involved in data collecting in this way. Simple facts like "location X has processed 2000 papers" won't be particularly helpful to higher management; data processing is required to turn this raw data into information that is valuable. A "% of records handled," "average documents processed per day per person," or "estimated completion date" are just a few examples of how this information might be presented.

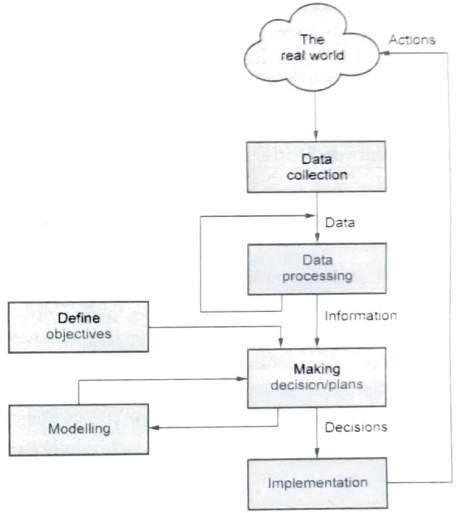

Fig. 1.11.1 : The project control cycle

- **Example :** The project management may look at each branch's "estimated completion date" for the data transfer process. These can be compared to the overall deadline for finishing this project phase. In essence, they are contrasting actual performance with a particular goal of the overall project. They might discover that one or two branches won't be able to deliver information on time. After that, they would have to decide what to do (Fig. 1.11.1 shows this as the box making decisions / plans).
- Moving personnel temporarily from one branch to another is one option. If this is done, there is always a chance that the date for the branch from which workers are being transferred will be pushed forward past the overall target date even while the completion date for the one branch will be pushed back to before that date. The project manager would have to carefully calculate the effects of transferring employees from specific branches. This involves simulating the effects of a prospective remedy. This method might be used to simulate a number of various proposals before one was selected for implementation.
- It is clear that a project plan is dynamic and will require ongoing modification throughout project execution. Project planning frequently receives a lot of attention in project management. While this is to be expected, almost always significantly more time is spent on the project itself than on its preparation. A good plan lays the groundwork for a good project, but it is useless without skillful execution. The initial strategy won't be adhered to rigidly, but rather revised to accommodate for developing circumstances.

1.12 Traditional Versus Modern Project Management Practices

- The fundamental strategy used by the software industry to create software has drastically changed during the past 20 years. Nowadays, very little new software is created from scratch.
- More and more software development initiatives are based on modifying pre-existing products or utilizing pre-built libraries.
- In either case, increasing code reuse and shortening project length are two key objectives of modern life cycle models. Other objectives include encouraging and accommodating customer input and participation in project development activity, as well as delivering the product incrementally with developing functionality.
- Customer change requests are welcomed rather than ignored. On the other side, customers are calling for more savings in both product pricing and delivery times. Project management procedures have evolved significantly as a result of these recent innovations.

- The following list of key distinctions between modern project management techniques and traditional techniques :

1. Planning incremental delivery :

- Projects were considerably simpler and more predictable a few decades ago than they are today. Back then, projects were meticulously planned out before beginning their real execution. After the project was started, monitoring and control procedures were used to make sure that it was carried out according to schedule. Today, projects must be done in a lot less time and quick application development and deployment are important tactics. Adaptive short-term planning has replaced traditional long-term planning. The project manager is no longer creating a long-term strategy for the project's completion; instead, the project manager now plan all incremental deliveries with developing functionalities.
- Extreme project management is another name for this style of project management. Extreme project management is a highly adaptable method of managing projects that emphasizes managing people rather than using formal and intricate planning and monitoring procedures (such as managing project stakeholders).

2. Quality management :

- Consumer awareness of product quality has significantly increased recently. Project managers now have a significant obligation to perform tasks related to quality management. Monitoring the quality of all intermediate artifacts and evaluating project progress are now among a project manager's primary duties.

3. Change management :

- Any changes to the requirements were previously infrequently accepted after the customer had approved them. Currently, customer feedback is actively sought out and incorporated into the development process. The usage of incremental delivery models is becoming more and more common to promote client feedback. Product development is done by introducing successive product iterations with progressively more advanced functionalities. Additionally, each release receives requests for customer comments to be incorporated. An organization now needs to keep track of the many versions and revisions that the product goes through as a result. The following is another justification for why tracking versions and revisions is becoming more and more crucial. Custom application development has gained popularity as a business strategy.
- As a result, it is now usual for a development company to need to support a significant number of product versions. In this situation, version control and product base lining are crucial functions of the project manager. As a result, the project manager now has a significant role for change management.

1.13 Case Study : Online Shopping System

- Everyone is so busy these days with their jobs that they rarely have time to buy in the (crowded) huge supermarkets. It takes longer and there is no assurance that every product will be available in every store when we shop because we have to look through each shelf to find what we're looking for. Furthermore, the weather is too severe for us to get the supplies we need. Because we must wait in line to pay, payment is another major issue. Not all stores are open constantly (24/7).
- We can use the method of online shopping to resolve the issues listed above.
- Customers benefit from this not only due to the high level of ease, but also due to the wider choices, affordable prices and improved information accessibility. Along with higher revenues, it improves an organization's value to its customers and fosters the development of sustainable skills.

  Advantages of online shopping system :

  - conserves time and energy.
  - The ease of doing our shopping at home.
  - There are many different goods available.
  - Good price reductions and discounts.
  - Learn all there is to know about the product.
  - We can contrast different models and brands.
- The majority of shoppers prefer internet buying since it is quicker and more effective. Numerous studies warn that the web's originality has faded and that user-centered design is now more necessary than ever. Businesses should always keep in mind that some things, including understanding the customer's requirements and wishes and keeping commitments, are timeless since they provide customers a reason to return. While physical retail locations will have the real goods and the manufacturer's packaging on display for direct examination, online stores are required to describe the products they are selling with text, images and multimedia materials (which might involve a test drive, fitting or other experimentation).
- Some internet retailers offer additional product information, such as directions, safety precautions, demonstrations or manufacturer specifications or they link to it. Some offer background details, suggestions or how-to manuals to aid consumers in making purchasing decisions. In certain establishments, customers are even permitted to remark or review the products.

A : Project requirement :

- For our client Metro Super Stores, it is our responsibility to establish an online shopping software system. The core concept of online shopping is not merely having a visually appealing website that appears in numerous search engines or the creative design of the site. It involves more than just sharing knowledge because it also involves developing relationships and earning money. Businesses should strive to eliminate all errors and improve their appeal to become more appealing to online shoppers if they want to keep their current clients or attract new ones.

B : Contact management :

- Keeping our clients' contact information in an address book or a Google spreadsheet is still an option, but it takes a lot of work. Our company may store vital customer data more effectively and efficiently with the use of a contact management system. Utilizing this digital strategy can help our company expand and diversify its clientele, which is one of its key advantages.

C : Plans :

- Here are the detailed directions.
  - Create a business strategy after selecting a specialty.
  - Choose a domain name.
  - Choose an online store builder.
  - Create the appearance of our online store.
  - Product-wise, expand our internet store.
  - Organize shipping.
  - Promote our online shop.
  - Start up our online shop.

D : Methods :

- Seven common online shopping techniques used by consumers,
  - Order online and pick up in store
  - Online shopping with curbside pickup (ship to store).
  - Buy online,
  - Pick up in store

- Webrooming,
- Digital wallets,
- Showrooming, etc.

E : Ways of Categorization :

- Business-to-Business (B2B)
- Business-to-Consumer (B2C)
- Consumer-to-Consumer (C2C)
- Consumer-to-Business (C2B)
- Business-to-Administration (B2A)
- Consumer-to-Administration (C2A)

F : Stakeholders for online shopping system :

- Any person, entity or party with an interest in a business and the results of its operations is considered a stakeholder. Employees, clients, shareholders, suppliers, communities and governments are typical examples of stakeholders. Companies frequently have to make trade-offs while attempting to satisfy a variety of stakeholders, each of whose interests are diverse.
- Fig. 1.13.1 shows types of stakeholders used for online shopping system.

Fig. 1.13.1 : Types of stakeholders

1. **Customers :** Many would contend that companies are there to serve their clients. Customers are stakeholders in a company because they are affected by the value and

quality of the goods and services offered. For instance, when boarding a flight, a passenger practically places their life in the hands of the airline.

2. **Employees :** Because they receive benefits and an income to maintain themselves, employees have a direct financial interest in the business (both monetary and non-monetary). The nature of the firm will determine whether employees' interests in health and safety exist (for example, in the industries of transportation, mining, oil and gas, construction, etc.).
3. **Investors :** Both debt holders and shareholders are considered investors. Investors put money into the company with the expectation of receiving a specific rate of return. The idea of shareholder value concerns investors frequently. All additional capital providers, such as lenders and possible buyers, are grouped together with this one. Although stakeholders are not always shareholders, all shareholders are by definition stakeholders.
4. **Suppliers and vendors :** Suppliers and vendors depend on a firm for ongoing income production through selling goods and/or services to it. Suppliers may be intimately involved in the business' activities in various industries, putting their health and safety at risk.
5. **Communities :** Communities play a significant role in the success of large firms based there. They are impacted by a variety of factors, such as the production of jobs, economic growth, health and safety. There is an immediate and dramatic impact on employment, incomes and spending when a large corporation moves into or out of a small community. There may be a health impact on various industries as well because businesses may change the environment.
6. **Governments :** Governments can also be seen as a significant stakeholder in a business since they collect taxes from the enterprise (corporate income taxes), from every employee (payroll taxes) and from other expenditures the enterprise makes (sales taxes). The total Gross Domestic Product (GDP) that businesses contribute to is beneficial to governments.

G : Project objectives :

- The project's major goal is to give Metro Superstore the ability to grow their business by allowing customers to purchase their products online using a secure and safe online payment mechanism. This will enable Metro Superstore to expand their market and draw more customers. Additionally, Metro Superstore will be able to make it quick and simple for customers to buy products and leave reviews. Customers can click a button to pay for things and take advantage of a variety of incentives online. Customers are finding it harder and harder to find the time to visit a mall and browse a vast selection of goods as times change.

- The customer will find it simpler to navigate through this large selection of products and compare them to similar products on the market thanks to the online shopping cart system. The project's other objective is to gather and then analyze consumer input in order to maintain or improve product quality and create better customer service. With this market domination, Metro Superstore hopes to draw in a greater variety of customers. Additionally, it will make it possible for Metro Superstore to establish connections with clients from all over the world.
- The idea tries to overcome the communication gap between the Metro supermarket and its patrons. Instead of making the effort to physically visit the business, customers will be able to order the things they want online from the convenience of their homes. Customers will have the option to register, peruse the products and add them to virtual shopping carts. After that, they can decide whether to pay with cash when the things are delivered or pay online using a secure online transaction.

H : Successful project :

- Seven promotional techniques for our online store,
  - Utilize our email list and continue to expand it.
  - Increase our natural social presence.
  - Make our website SEO-friendly.
  - Make informative, entertaining material.
  - Try Google Ads out.
  - Use social media to promote.
  - Partner with brands that are complementary.

I : Management :

- Another trend that is starting to emerge is the creation of explicit policies for the administration of online product returns, which directly addresses a major worry or query from some customers. One of the main factors preventing some of the expansion of online shopping has been the ambiguity surrounding how returns of damaged or unwanted goods would affect the overall cost and inconvenience to the customer. Companies who can clearly articulate their return policies for online purchases and offer a simple method for customers to do so have an edge over competitors who are perceived as "difficult" to work with when processing returns for online purchases.

❑❑❑

Unit II

2 Project Design and Evaluation

Syllabus

Project Design : Overview of UML diagrams : Use case, Class, Activity, State, Sequence, Deployment.

Project Evaluation : What is Project Evaluation ? Importance of Project Evaluation, Cost Benefit Evaluation Techniques.

Process Evaluation and Improvement : The Process Improvement Process : The Process Improvement Cycle, Process Measurement : The GQM Paradigm, Process Analysis : Techniques of Process Analysis, Process change : The Process Change Process.

Case study : Online Shopping System, Perform Cost-Benefit Analysis using Microsoft Excel.

Contents

2.1 Overview of UML Diagrams

- UML diagrams can be used as a way to visualize a project before it takes place or as documentation for a project. But the overall goal of UML diagrams is to **allow teams to visualize how a project is or will be working** and they can be used in any field, not just software engineering.
- UML stands for Unified Modeling Language. UML **is not a programming language** but it is a visual language. It describes the behavior and structure of the system. It is used to improve the quality of your systems analysis and design. With the help of UML diagrams, it becomes easy to describe how a project will work.
- UML is linked with **object oriented** design and analysis. UML makes the use of elements and forms associations between them to form diagrams.
- UML diagrams represent two different views of a system model :
  1. **Structural diagrams -** This view describes the static structure of the system using objects, attributes, operations and relationships. Ex : Class diagram, composite structure diagram, component diagrams, object diagrams, class diagrams and deployment diagrams.
  2. **Dynamic (or behavioral) view -** This view describes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. Behavior diagrams include : Use case diagrams, state diagrams, activity diagrams and interaction diagrams.

2.1.1 Use Case

- A use case diagram gives graphical representation of your system's users and their interactions with the system. The purpose of use case diagram is to demonstrate the different ways that a user can interact with a system. It consists of actors, set of actions and relationship between them.

2.1.1.1 Use Case Diagram Components

- Actors
- Functionalities to be represented as use case
- Relationships among the use cases and actors.
- **Actors :** An actor can be a person, an organization or an outside system that fulfill certain roles in a given system. The various roles that an actor represents are the actual business roles of the users in that system. An actor in a use case diagram interacts with a use case.

Example 1 : In a banking application, the customer entity represents an actor in an application. Similarly, the person who provides service at the counter is also an actor. The actor is shown as a stick Fig. 2.1.1 in the use case diagram shown "outside" the system boundary as shown in the Fig. 2.1.1 below.

Example 2 : Consider the case where a customer withdraws cash from an ATM. The customer is an actor here.

Actors can be classified as follows

1. **Primary actor :** They are the main users of the system who fulfill their goal by using some service from the system. For example, a customer uses an ATM to withdraw cash when they need it. Here customer is a primary actor.
2. **Supporting actor :** They provide some service to the system. Such an example is "bank agents" who replenish cash reserves. It may be noted that replenishment of cash in an ATM is not the main function of an ATM.

Functionalities to be represented as use case :

- A use case is a function provided by the system.
- Continuing with the ATM example, cash withdrawal is a function that an ATM provides. Therefore, this is a use case. Other possible use cases include checking balance, changing PIN, etc.
- Use cases include both successful and unsuccessful scenarios of user interactions with the system. For example, customer authentication by an ATM would fail if they entered an incorrect PIN. In this case, an error message will appear on the ATM screen.
- A use case is shown as an ellipse in a use case diagram.
- The Fig. 2.1.1 below shows the use-case diagram for the ATM. (Refer Fig. 2.1.1 on next page)

Actors

- **Bank customer :** This actor represents a person with a valid bank card. The bank card is theirs and they know the PIN code.
- **Bank :** This actor represents a financial institution that provides ATM services. Responsible for verifying bank customers, authorizing transactions and recording completed transactions.
- **Maintenance person :** This actor represents the person responsible for ATM maintenance, paper replenishment and cash replenishment.

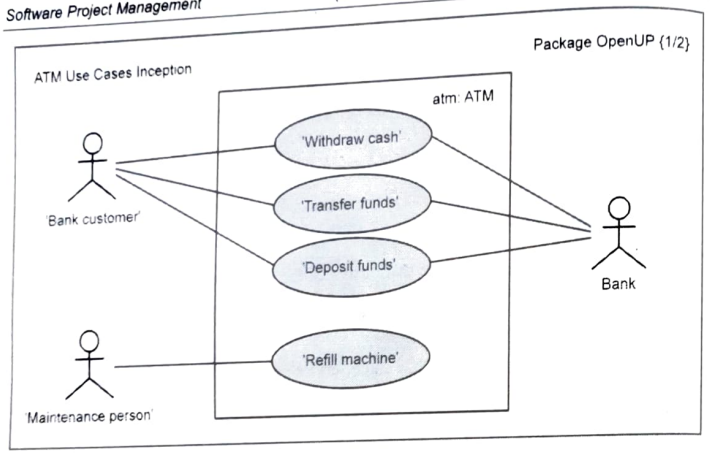

Fig. 2.1.1 : Use-case diagram for the ATM

Use cases

- **Cash withdrawal :** This use case describes how a bank client uses an ATM to withdraw money from their bank account.
- **Fund transfer :** This use case describes how a bank client uses an ATM to transfer money between different bank accounts.
- **Storage of funds :** This use case describes how a bank customer deposits money into an account.
- **Refilling machine :** This use case describes how a maintainer replenishes money, receipt and envelopes.

Use case relationships

- Use cases share different kinds of relationships. A relationship between two use cases is a dependency between the two use cases. Defining a relationship between two use cases is the decision of the modeler of the use case diagram.
- There are three types of relationships between use cases :
  1. Include relationship
  2. Extend relationship
  3. Use case generalization.

1. Include relationship :

- Include relationships are used to show common behavior that is shared by multiple use cases. This could be thought of as analogous to writing functions in a program to avoid writing the same code for multiple times. Such a function would be called from **different points** within the program.
- An include relationship is shown by a directed arrow with a dotted shaft. The arrowhead points to the child use case and the parent use case is attached to the base of the arrow.

Example

- Consider an email application. A user can send a new email, reply to an email they have received or forward an email. However, in each of these three cases, the user must be logged in, to perform these actions. So we could have a login use case that is part of the write mail, reply and forward email use cases.
- The relationship is shown in Fig. 2.1.2.

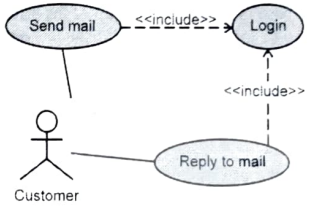

Fig. 2.1.2 : Relationship in an email application

2. Extend relationship :

- In an extended relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case. The extend relationship is represented by a directed arrow with a dotted shaft, similar to the include relationship. The arrowhead points to the parent use case and the child use case is attached to the base of the arrow.

Example

- Consider an online bookstore. The system allows the authenticated user to buy the selected book(s). During order entry, the system also allows you to enter any special shipping instructions, such as calling the customer prior to delivery. This shipping instructions step is optional and not part of the main order use case.

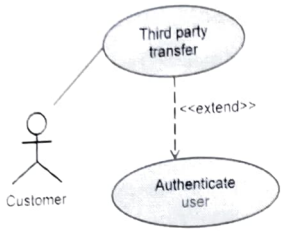

Fig. 2.1.3 : Relationship in online bookstore

3. Use case generalization :

- It is a parent-child relationship between use cases. Generalization relationships are used to represent inheritance between use cases. A derived use case specializes some functionality that it already inherited from the base use case. In a use case diagram, generalization is shown as a directed arrow with a triangle arrowhead. The child use case is connected at the base of the arrow. The tip of the arrow is connected to the parent use case.

Example

- Consider a graphics application of drawing a polygons. We could have a use case to draw a polygon. A rectangle is a specific example of a polygon with four sides at right angles to each other. So the use case draw rectangle inherits the properties of the use case draw polygon and overrides its draw method.

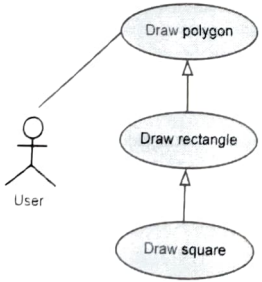

Fig. 2.1.4 : Generalization relationships in a graphics application of drawing a polygons example

2.1.1.2 Use Case Diagram Symbols and Notation

- **Use cases :** Horizontal shaped ovals.
- **Actors :** Stick figures that represent the people who actually use the use cases.
- **Association :** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
- **System boundary box :** A box that sets the scope of the system for use cases. All out-of-the-box use cases would be considered outside the scope of this system. For example, psycho killer is out of range in the chainsaw example below.
- **Packages :** A UML shape that allows you to group different elements into groups. As with component diagrams, these groupings are represented as file folders.

2.1.2 Class

- A class diagram is a static diagram. It represents a static view of the application. A class diagram is not only used to visualize, describe and document various aspects of a system, but also to construct the executable code of a software application.
- A class diagram describes the attributes and operations of a class, as well as the constraints placed on the system. Class diagrams are widely used in modeling object-oriented systems because they are the only UML diagrams that can be directly mapped using object-oriented languages.
- A class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a **structure diagram**.

2.1.2.1 Elements in Class Diagram

- A class diagram contains system classes with their data members, operations and class relationships.
- In UML syntax, a class is identified by a solid outline rectangle with three sections that contain.

1. Class name :

- The class is uniquely identified in the system by its name. A text string is taken as the class name. It is located in the first compartment in the classroom rectangle.
- **Rules for class name :**
  - It should be start with capital letter
  - It should be always at the centre of the first compartment
  - It should be written in bold format
  - If it is UML abstract class, it should be written in italic format.

- **Attributes :** A property shared by all instances of the class. It is located in the second compartment in the classroom rectangle.
- **Operation :** Action execution can be performed on any object of the class. It is located in the last compartment in the classroom rectangle.

Example

- To create a structural model for an educational organization, 'Course' can be treated as a class that contains 'courseName' and 'courseID' attributes with 'addCourse()' and 'removeCourse()' operations that can be performed on any object to that class.

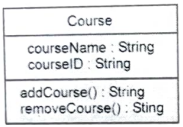

Fig. 2.1.5 : Class diagram

Generalization/Specialization

- It describes how one class is derived from another class. Derived class inherits the properties of its parent class.

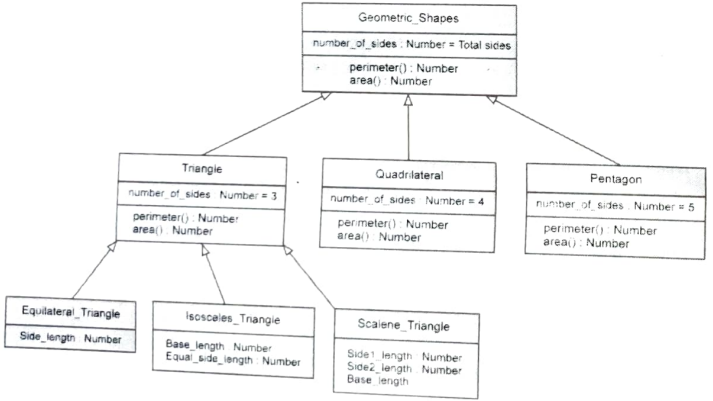

Fig. 2.1.6 : Generalization

- Geometric shapes is a class that describes how many sides a particular shape has. Triangle, quadrilateral and pentagon are classes that inherit from the geometric shapes class property. So the relationships between these classes are generalizations. Now, equilateral triangle, isosceles triangle and scalene triangle, these three classes all inherit the properties of the triangle class because each of them has three sides. So this is a specialization of the triangle class.

2.1.2.2 Relationships

- Existing relationships in a system describe connections between classes in that system.
- Mainly there are three kinds of relationships in UML :

  1. Dependencies 2. Generalizations 3. Associations.

1. Dependencies :

- Dependency means one class depends on another class. Therefore a change in one class forces changes in the other class. So it will create a weaker relationship.
- **Example :** In the following Fig. 2.1.7, it shows that student has a dependency on college.

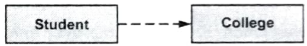

Fig. 2.1.7 : Dependency between student and college

2. Generalization :

- The generalization relationship occurs between two objects, in which one entity is parent and other is child. Child inherits the functionality of its parent and it can access it as well as update it.
- It is represented by a solid line with hollow arrowhead which pointing towards the parent element from the child element.

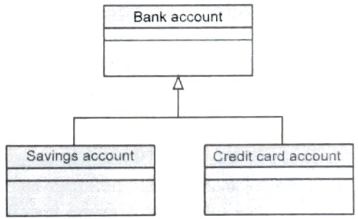

Fig. 2.1.8 : Example of generalization relationship

- From above Fig. 2.1.8, user can have both saving account and credit card account. Both accounts inherits the properties from the bank account, like account number, account balance, etc. (Refer Fig. 2.1.8 on previous page)

3. Association

- It allows messages to be exchanged between objects at both ends of the association. It is a simple straight line connecting two class boxes. We can name the association and we can also list the names of the roles and the multiplicity of adjacent classes on both ends. The association can be unidirectional.
- **Example 1 :** In a structural model for an organization system, an employee (an instance of the 'Employee' class) is always assigned to a specific department (an instance of the 'Department' class) and the association can be represented by a line connecting the respective classes.

Fig. 2.1.9 : Association diagram of employee and department

- **Example 2 :** The relationship between student and college is shown with association studies.

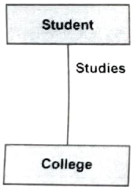

Fig. 2.1.10 : Association of student and college

2.1.3 Activity Diagrams

- An activity diagram is basically a flowchart that shows the flow from one activity to another activity. The activity can be described as the activity of the system. Control flow is transferred from one operation to another. This flow can be sequential, branched or concurrent. An activity diagram shows the flow of control from the start point to the end point and shows the different decision paths that exist during the execution of the activity.

- The purpose of an activity diagram can be described as -
  1. Draw the activity flow of the system.
  2. Describe the sequence from one activity to another.
  3. Describe parallel, branched and concurrent system flow.

2.1.3.1 Components of an Activity Diagram

- **Activity :** An activity refers to a specific action performed in a logical control flow. The activity is represented by a rounded rectangle. The label inside the rectangle indicates the corresponding activity.
- There are two special types of activity nodes : Start and end. They are represented by a filled circle and a filled circle with a border.
  - The start node represents the starting point of the flow in the activity diagram. There can be multiple start nodes, meaning that invoking a particular activity diagram would initiate multiple flows.
  - The end node represents the end point of all activities. As well as a start node, there can be multiple end nodes. Any transition reaching the final node would stop all activity.
- **Flow :** A flow (also called an edge or transition) is represented by a directional arrow. This is used to represent the transfer of control from one activity to another. A flow is often accompanied by a label, called a **guard state**, indicating a necessary condition for the transition to occur.
- **Decision :** A decision node is represented by a diamond. It is a point where one flow enters and two or more flows leave. Control flow can follow only one of the outgoing paths. Outgoing edges often have guard conditions that indicate true-false or if-then-other conditions. It always includes two or more output arrows.
- **Unite :** This is represented by a diamond shape with two or more inputs and one output. A merge node represents the point where at least one control should reach before further processing continues.
- **Fork :** A fork is the point where parallel activities begin. For example, when a student is enrolled at a university, he can apply for a student card and a library card at the same time. A fork is graphically represented with a black bar with one input and several outputs.
- **Connect :** A connection is represented by a black bar with multiple input streams but one output stream. Physically, it represents the synchronization of all concurrent activities. Unlike a merge, in a merge, all incoming checks must be completed before further progress can be made. For example, a sales order is closed only when the customer receives the product and the sales company receives its payment.

- **Note :** UML allows you to attach a note to various diagram components to present some textual information. The information can be just a comment or it can be some limitation. A note can be attached to a decision point, for example to indicate branching criteria.
- **Partition :** Different components of an activity diagram can be logically grouped into different areas called **compartments or swimlanes**. They often correspond to different units of the organization or different actors. The drawing area can be divided into multiple sections using vertical (or horizontal) parallel lines. Sections in the activity diagram are optional.

Symbols of components :

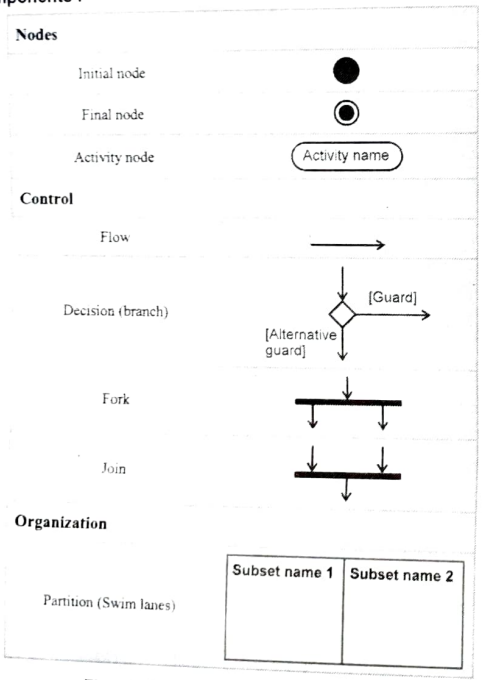

Fig. 2.1.11 : Notations of symbols

- **Example :** Consider an example of login page.

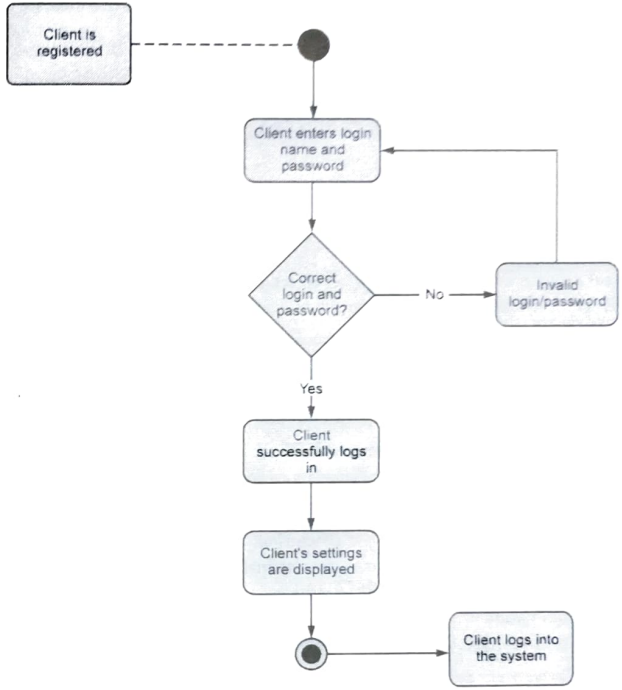

Fig. 2.1.12 : Activity diagram for a login page

2.1.4 State Diagram

- A state diagram represents the state of the system or the part of the system. State diagrams are also called as **state machines** or **state-chart diagram**.
- A state diagram is a pictorial representation of a system with all its states and the various events that lead to the transition from one state to another.
- For example, imagine a computer. Some possible states it could have are : running, shutdown, hibernate. The transition from the running state to the shutdown state occurs

when the user presses the "Shutdown" switch or clicks the "Shutdown" button as displayed by the operating system. To click shutdown button or press the power button are external events which causes the transition of state.

2.1.4.1 Components of State Diagram

- **State :** A state is any distinct stage that an object passes through in its lifetime. An object will remain in the same state until something happens which will move it from one state to other state. A state can be simple or composite state.
- The states can be classified in three different types :
  1. **Initial :** When object is created, it will remain in initial state. It is represented by black filled circle.
  2. **Final :** This is a final or end state of an object. In this state an object cannot move from one state to other state. It is represented by filled circle present within a circle.
  3. **Intermediate :** Any state, which is neither initial, nor final state.

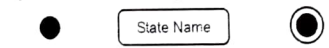

Fig. 2.1.13 : Initial, intermediate and final states of a state diagram

- Intermidiate state have two compartments which are separates by a horizontal line. One is name compartment and other is internal transitions compartment.
  - Name compartment : It contains name of the estate, which is a short , simple string.
  - Internal transitions compartment : It contains all activities performed by system when system is in this state. The syntax for internal activities is: action-label / action-expression.

Reserved action labels :

1. **Entry :** When system enters in this state.
2. **Exit :** When system exit this state.
3. **Do :** Ongoing behavior, performed as long as object is in the state.

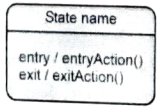

Fig. 2.1.14 : A typical state in a state diagram

Transition

- Transition is a flow from one state to another which shows how the states changes.
- It is represented by a solid directed line from current state to next state labelled by an event.

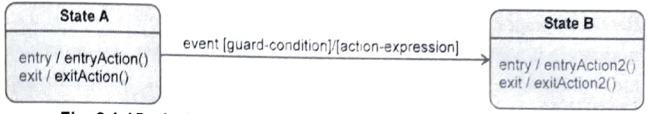

Fig. 2.1.15 : A state diagram showing transition from state A to B

2.1.5 Sequence Diagram

- It represents the behavioral aspects of the system. A sequence diagram shows interactions between objects sequence diagram.
- It represents the behavioral aspects of the system. A sequence diagram shows interactions between objects through passing messages from one object to another depending on time in a system.

2.1.5.1 Elements in a Sequence Diagram

- A sequence diagram contains system objects and their lifeline and the messages passing between them.
- **Object :** The objects appear at the top of the sequence diagram. The object is displayed in a rectangular box. The name of the object is preceded by a colon ":" and the name of the class from which the object is created. The entire string is underlined and displayed in a rectangular box. Also we can use only class name or only instance name.
- Objects that are created at the time of use case execution and are involved in message passing appear in the diagram where they are created.
- **Line of life :** A vertical line pointing down from the object's frame is shown as the object's lifeline. A rectangular bar on the life line indicates that it is currently active.
- **Messages :** Messages are shown as an arrow from the sender's lifeline to the recipient's lifeline and are labeled with the name of the message. The chronological order of messages passing through the object's lifelines shows the sequence in which they occur.
- There can be different types of messages :
  - **Synchronous messages :** The receiver starts processing the message after receiving it and the sender has to wait for it to be executed. A straight arrow with a closing and filling arrow from the sender's life line to the receiver's end represents a synchronous message.
  - **Asynchronous messages :** With asynchronous messages, the sender does not have to wait for the receiver to process the message. A function call that creates a thread can be represented as an asynchronous message in a sequence diagram. A straight arrow with an open arrow from the sender's lifeline pole to the receiver's end represents an asynchronous message.

- **Return message :** To call a function when we need to return the value of the object from which it was called, then we use return message. However, it is optional and we use it when we want to model our system in greater detail. This message is represented by a dashed arrow with an open arrow from the sender's life line to the recipient's end.

Message Type	Notation
Synchronous message	→
Asynchronous message	→
Response message	⇠

- **Example :** Consider an example of online examination system.

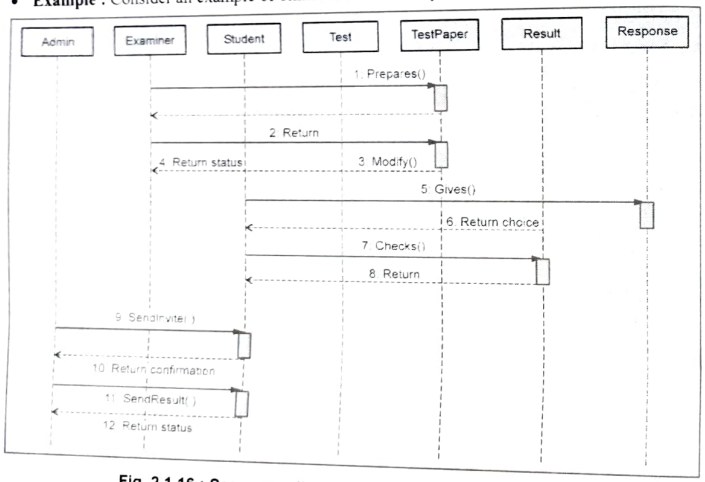

Fig. 2.1.16 : Sequence diagram of an online exam system

2.1.6 Deployment Diagram

- It models the physical architecture of a system. It is used for describing hardware components where the software components are actually deployed.
- The main purposes of deployment diagram.
  - To represent the hardware topology of the system.

  - To represent how different hardware components interacted on which software components are deployed.
  - To describe processing of nodes.

2.1.6.1 Symbol and Notation of Deployment Diagram

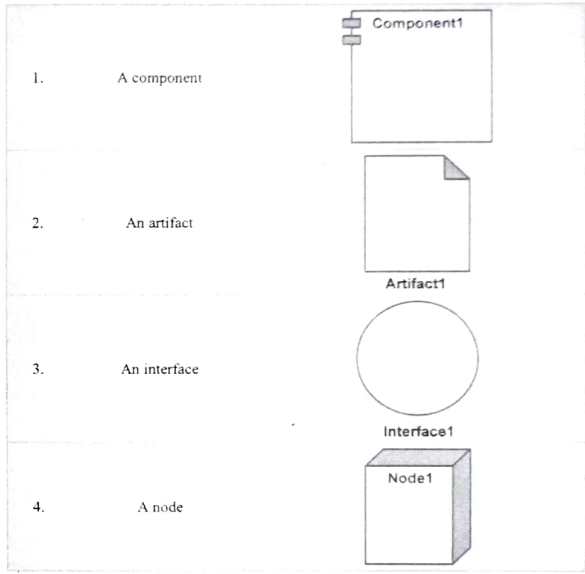

Fig. 2.1.17 : Symbol and notation of deployment diagram

Review Questions

1. Explain use case diagram with its components.
2. What is activity diagram. What are components of activity diagram.
3. Define state diagram. What are components of state diagram.
4. Define sequence diagram. What are components of state diagram.
5. Explain deployment diagram with symbol and notation.

6. Write a short note on : Class, Use case, Activity diagram, State diagram, Sequence diagram, Deployment diagram

2.2 Project Evolution

2.2.1 Introduction

- We all know that so many IT projects failed costs Billions of Dollars to the world. Wastage of money, working hours and man power. Initially people are trying to create projects with their instinct without following any methodology. Projects may get extended to achieve goals which not clear or well define. Projects may not provide services or result as expectations. Sometimes due personal interests of developers projects may get stretch somewhere else. Behind every failed IT project we "lost everything for nothing". This is not true at least we force to adapt project management. Whatever mistakes we have done in past, we learn our lesson very well. Now project did not came from someone's instinct or hobby or liking. Before project should be get develop it must be get evaluated to check whether project is worth doing.
- It is not necessary that every project evaluation is done on financial basis but it could be evaluated on the basis of problem stated and propose solution is capable of increasing business by minimizing efforts to get result or can provide a platform where we can build business. Sometimes it may possible that desire results extended the cost of project but if it will combine with other projects it will give good returns in business. All these possibilities check with sophisticate evaluation methods.

2.2.2 What is Project Evolution

- Project evaluation is the process of measuring the success of a project, program or portfolio. This is done by collecting data about the project and using an evaluation method that allows evaluators to find opportunities to improve performance. Project evaluation is also essential to keep stakeholders informed of project status and any changes that may be required to the budget or schedule.
- Every aspect of the project, such as cost, scope, risks or Return On Investment (ROI), is measured to see if it is progressing according to plan. If there are bumps in the road, this data can inform how projects can be improved. Basically, you're asking the project a series of questions designed to find out what's working, what can be improved and whether the project is useful. Tools such as project dashboards and trackers assist in the evaluation process by making key data available.

- The project evaluation process has been around as long as the projects themselves. But when it comes to the science of project management, project evaluation can be divided into three main types or methods : Pre-project evaluation, interim evaluation and post-project evaluation. Let's take a look at the project evaluation process, what it entails and how you can improve your technique.

2.2.2.1 Project Evaluation Methods

- There are three points in the project where evaluation is most needed. Evaluating your project at any time, following points should be planned.

1. Pre-project assessment

- In a sense, you pre-evaluate your project when you write your project charter to present to stakeholders. You can't effectively plan, staff and manage a new project if you haven't evaluated it first. A pre-project evaluation is the only sure way you can determine the effectiveness of a project before it is implemented.

2. Ongoing evaluation of the project

- To make sure your project continues on schedule and meets all the planning and budget milestones you've set, it's important to constantly monitor and report on your work in real time. Only with project metrics can you measure the success of your project and whether you are meeting your project goals and objectives. It is strongly recommended to use project management software for real-time continuous evaluation of projects.

3. Post-project evaluation

- Think of it as an autopsy. A post-project evaluation is when you review the project documentation, interview the project team and principals and analyze all relevant data to understand what worked and what went wrong. Only by creating this clear picture you can solve problems in upcoming projects.

2.2.2.2 Evaluation Criteria

Evaluation criteria	Description
Relevance and strategic fit project	The extent to which the development objectives the interventions are in line with the requirements of the beneficiaries, country needs, global priorities and policies of partners and donors.

Project proposal validity	The extent to which the project proposal is logical and coherent.
Project progress and effectiveness	The extent to which the project's immediate objectives were achieved or expected to be achieved, taking into account their relative importance.
Resource efficiency	The measure of how efficiently resources/inputs (funds, expertise, time, etc.) are translated into results.
Management effectiveness arrangement	The extent to which management capacities and measures are in place implement supports achieving results.
Impact orientation a sustainability of the project	Strategic orientation of the project to create a significant contribution to the wider, long-term and sustainable developmental changes. The likelihood that the results of the project are permanent and can be maintained or even scaled and replicated by project partners after the main assistance is completed.

2.2.2.3 Project Evaluation Steps

- Regardless of when you decide to conduct a project evaluation, the process always has four phases : Planning, implementation, completion and dissemination of messages.

1. Planning

- The ultimate goal of this step is to create a project evaluation plan, a document that explains all the details of organization's project evaluation process. When planning a project evaluation, it is important to identify the stakeholders and what their short and long term goals are. One need to make sure that goals and objectives for the project are clear and it is important to have criteria in place to tell whether those goals and objectives are being met.
- So write a series of questions to ask stakeholders. These questions should include topics such as project framework, best practices and metrics that determine success.
- By including stakeholders in project evaluation plan, it will help to gain direction as the project progresses while developing a relationship with stakeholders. They will receive progress reports from project manager throughout the phases of the project and by building this initial relationship, to gain their confidence that organization can manage the project to their satisfaction.

2. Implementation

- While the project is running, we need to monitor all aspects to make sure we are on schedule and on budget. One of the things especially should monitor during a project is the percentage of completion. This is already done when create status reports and meetings with development team. To make sure project on track, hold the team accountable for completing tasks on time and keep track of key dates so everyone should know when tasks are due.
- Do not forget to monitor the quality. It doesn't matter if you deliver the project on time if the product is bad. Maintain quality controls and do not delegate this responsibility. Instead, take it upon yourself.
- Maintaining a close relationship with the project budget is as important as monitoring schedule and quality. Track costs. They will change throughout the project, so don't panic. However, be transparent if you notice a growing need for additional funds. Let your steering committee know as soon as possible so there are no surprises.

3. Completion

- When Project is completed, still there is work to do. Data collected throughout assessment is organize and analyses for the fixing issues reported while running the processes. This data is use for forecasting of short term and long term effects from the experiences during assessment of project.

4. Reporting and Dissemination

- Once the assessment is complete, the results should be recorded. To do so, you create a project evaluation report, a document that provides lessons for the future. Deliver a message to stakeholders to keep them updated on the progress of the project.
- How will you spread the message ? Organization may already have a protocol in place for this. Perhaps stakeholders prefer to meet face-to-face to get results. Or maybe they prefer a PDF with easy-to-read tables and charts. Make sure you know your audience and tailor your message to them.

2.2.3 Importance of Project Evaluation

- Where possible, the benefits of the implemented project should be assessed with a financial value. For commercial organizations, it may be related to increased profits due to either increased revenue or cost savings. For non-profits, we would try to quantify the benefits, although we cannot enter the exact financial value. In the example we used earlier in

relation to the IT system that improved diagnosis of a specific disease, an increase in the diagnosis rate may be indicated.

- **Outline an implementation plan :** In addition to the ICT aspects of the project, activities such as marketing, promotion and operational and maintenance infrastructure must be considered. One consideration will be what the project activities might be outsourced and which are best kept in-house.
- Implementation management will also be detailed here. Responsibilities are assigned for tasks identified in the framework implementation plan. Key decision points or milestones where there is a status check the status of the implementation needs to be ascertained. As we will see, for a large implementation a number projects that can be managed as a program may be needed.
- **Costs :** After outlining the steps needed to set up the operations needed for the design, schedule the expected costs can now be presented related to the planned approach.
- There will obviously be some uncertainty about some of the costs, particularly when it comes to the details of the requirements have not yet been processed.
- **Financial case :** There are a number of ways in which income and expenditure information can be analyzed.
- **Risks :** We note here that many estimates the costs and especially the benefits of the project will be speculative at this stage and in the risk section should take this into account.
- Similarly, we can distinguish project risk - related to the threat of successful project implementation - from business risk - related to factors threatening the benefits of the delivered project. In the business case, the main focus is at business risk.

2.2.4 Cost - Benefit Analysis

- Even where the estimated benefits exceed the estimated costs, this is often necessary decide whether the proposed project is the best of several options. Not all projects can be carried out at any time and in any case the most valuable projects should get most sources.
- A cost-benefit analysis consists of two steps :
  - Identification of all costs and benefits of project implementation and operation of the supplied application these include development costs, operating costs and benefits expected from the new system. If the proposed system is a replacement, these estimates should reflect the change in costs and benefits thanks to the new system. For example, the new sales order processing system could only claim benefit the organization by increasing sales through the use of the new system.

  - Expressing : These costs and benefits in common units we must express each cost and benefit - and that net profit, which is the difference between the two - in money. Most direct costs can be easily quantified in monetary terms and can be divided into the following categories :
  - Development costs, including development staff costs;
  - Set-up costs, which consist of the costs of getting the system up and running, especially any new hardware, but also including file conversion costs, recruitment and personnel training;
  - Operating costs related to the operation of the system after installation.

Cash flow forecasting

- Just as important as estimating the total costs and benefits of a project is creating a cash flow forecast that shows when expenses and income will occur.

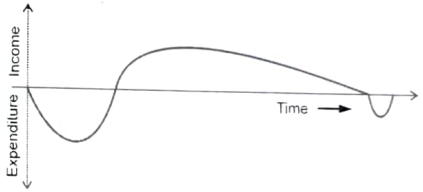

Fig. 2.2.1 : Typical product life cycle cash flow

- During the development of the project, we have to spend money, such as employee wages. Such expenses cannot wait until income is received (either from the use of the software developed for own use or from sale). We need to know that we can finance it development expenses either from the company's own resources or through borrowings.
- A forecast is needed when expenses such as salary payment and any income can be expected. Accurate cash flow forecasting is difficult because it is done early in a project's life cycle (at least before any significant spending occurs) and many items (especially the benefits of using the software) may take several years to estimate future.
- It is customary to ignore the effects of inflation when estimating future cash flows. Inflation rate forecasts tend to be insecure. Additionally, if expenses increase due to inflation, income is likely to increase as well proportionately.

Cost - benefit assessment techniques

- We will now look at some methods for comparing projects based on their cash flow forecasts. Table 2.2.1 illustrates cash flow forecasts for four projects. In any case, it is

assumed that the cash flows are held at the end of each year. For short-term projects or where there are significant seasonal cash flows quarterly or even monthly cash flow forecasts might be appropriate.

Year	Project 1	Project 2	Project 3	Project 4
0	– 100,000	– 1,000,000	– 100,000	– 120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net profit	**50,000**	**100,000**	**50,000**	**77,000**

Table 2.2.1 : Four project cash flow projections - figures are end of year totals(£)

The net profit

- The net profit of a project is the difference between total costs and total revenues over the life of the project. Project 2 in Table 2.2.1 shows the highest net profit, but at the expense of a large investment. If we were to invest £1m, we could undertake all the other three projects and get even bigger the net profit. Also note that all projects involve an element of risk and we may not be prepared to risk 1 million.
- In addition, simple net income does not take into account the timing of cash flows. Projects 1 and 3 each have a net profit of 50,000 and thus according to this a selection criterion would be equally preferable. Most income comes late over the life of project 1, while project 3 brings steady income throughout its lifetime.
- Having to wait for a return has the disadvantage that the investment must be financed longer. Add to that the fact that, other things being equal, more is estimated distant futures are less reliable than short-term estimates and we see that the two projects are not equivalent preferable.

Payback time

- The payback period is the time required to break even or repay the initial investment. Normally a project with the shortest payback period will be chosen on the basis that the organization will want to minimize the time the project is "in debt".
- The advantage of the payback period is that it is easy to calculate and is not particularly sensitive to small forecast errors. Its disadvantage as a selection technique is that it ignores

overall profitability project - effectively totally ignores any income (or expense) once the project breaks even. That is a fact that projects 2 and 4 are more profitable overall than project 3 is ignored.

Return on investment

- Return On Investment (ROI), also known as accounting rate of return (ARR), provides a way comparing net profitability with required investment. There are some variations on the formula used calculate ROI, but the straightforward common version is :

$$\text{ROI} = \frac{\text{Average annual profit}}{\text{Total investment}} \times 100$$

- Calculating the ROI for project 1, the net profit is £50,000 and the total investment is £100,000. The return on investment is therefore calculated as,

$$\text{ROI} = \frac{\text{Average annual profit}}{\text{Total investment}} \times 100 = \frac{50,000/5}{100,000} \times 100 = \mathbf{10\ \%}$$

- Return on investment provides a simple and easily calculated measure of return on capital. Unfortunately, suffers from two serious disadvantages. Like pure profitability, it doesn't take timing into account cash flow. More importantly, this rate of return has no relation to the interest rates offered or charged by banks (or any other prevailing interest rate) because it does not take into account the timing of cash flows or compounding interest. It is therefore potentially very misleading.

Net present value

- The net present value calculation is a project evaluation technique that takes into account the profitability of the project and the timing of the cash flows that are produced. This is based on the idea that getting 100 pounds today is better than waiting until get it next year. For example, we could invest the £100 in the bank today and to have £100 plus interest per annum. If we say that the present value of £100 per year is £91, we mean that £100 per year is equivalent to £91 now.
- The equivalent of £91 now and £100 a year means we're discounting future income by approximately 10 %. If we now received £91 and invested it for a a year at an annual interest rate of 10 % it would be worth £100 a year. The annual rate at which we discount future profits is called the **discount rate** - 10 % in the above example.

- Similarly, £100 received in two years would have a present value of approx. £83 - in other words, £83 invested at 10 % interest would yield approximately £100 over two years. The present value of any future cash flow can be obtained by using the following formula,

$$\text{Present value} = \frac{\text{Value in year t}}{(1 + r)^t}$$

- Where r is the discount rate expressed as a decimal value and t is the number of years into the future after which cash flow occurs. Alternatively and rather more simply, the present value of the cash flow can be calculated by multiplying the cash flow by the appropriate discount factor. Small table discount factors are given in Table 2.2.2. The NPV for the project is obtained by discounting each cash flow (both negative and positive) and adding the discounted values. It is usually assumed that any the initial investment is made immediately (denoted as year 0) and is not discounted. Subsequent cash flows are usually assumed to occur at the end of each year and are discounted by the appropriate amount.

	Discount rate (%)					
Year	**5**	**6**	**8**	**10**	**12**	**15**
1	0.9524	0.9434	0.9259	0.9091	0.8929	0.8696
2	0.9070	0.8900	0.8573	0.8264	0.7972	0.7561
3	0.8638	0.8396	0.7938	0.7513	0.7118	0.6575
4	0.8227	0.7921	0.7350	0.6830	0.6355	0.5718
5	0.7835	0.7423	0.6806	0.6209	0.5674	0.4972
6	0.7462	0.7050	0.6302	0.5645	0.5066	0.4323
7	0.7107	0.6651	0.5835	0.5132	0.4523	0.3759
8	0.6768	0.6274	0.5403	0.4665	0.4039	0.3269
9	0.6446	0.5919	0.5002	0.4241	0.3606	0.2843
10	0.6139	0.5584	0.4632	0.3855	0.3220	0.2472
15	0.4810	0.4173	0.3152	0.2394	0.1827	0.1229
20	0.3769	0.3118	0.2145	0.1486	0.1037	0.0611
25	0.2953	0.2330	0.1460	0.0923	0.0588	0.0304

Table 2.2.2 : NPV discount factors

- Assuming a 10 % discount rate, the NPV for project 1 (Table 2.2.1) would be calculated as in Table 2.2.3. The net present value for project 1, using a 10 % discount rate, is therefore 618. Using a 10 % discount rate, calculate the net present values for projects 2, 3 and 4 and decide which, on the basis of this, is the most beneficial to pursue.

Year	**Project 1 cash flow**	**Discount factor @ 10 %**	**Discounted cash flow**
0	– 100,000	1.000	– 100,000
1	10,000	0.9091	9,091
2	10,000	0.8264	8,264
3	10,000	0.7513	7,513
4	10,000	0.6830	6,830
5	100,000	0.6209	62,090
Net Profit	40,000	618	

Table 2.2.3 : Applying the discount factors to project 1

- It is interesting to note that the net present values for projects 1 and 3 are significantly different - although both yield the same net profit and both have the same return on investment. The difference in NPV it reflects the fact that for Project 1 we have to wait longer for most of the revenue.
- The main NPV problem for deciding between projects is choosing an appropriate discount rate. Some organizations have a standard rate, but if not, a discount rate should be chosen reflect available interest rates (borrowing costs when the project must be financed by loans) plus some a premium reflecting the fact that software projects are usually riskier than lending money to a bank. The exact discount rate is usually less important than ensuring that the same discount rate is used for everyone compared projects. However, it is important to check that the order of the projects is not sensitive small changes in the discount rate.

Example 2.2.1 : *Calculate the net present value for each of the projects A, B and C shown in Table 2.2.4 using each of the discount rates 8 %, 10 % and 12 %.*

For each of the discount rates, decide which the best project is. What can you conclude from these results ?

Solution :

Year	Project A (£)	Project B (£)	Project C (£)
0	– 8000	– 8000	– 10000
1	4000	1000	2000
2	4000	2000	2000
3	2000	4000	6000
4	1000	3000	2000
5	500	9000	2000
6	500	– 6000	2000
Net Profit	**4000**	**5000**	**6000**

Internal rate of return

- One disadvantage of NPV as a measure of profitability is that although it can be used to compare projects, may not be directly comparable to income from other investments or the cost of borrowing capital. These costs are usually quoted as a percentage interest rate. The internal rate of return (IRR) attempts to do just that they provide a measure of profitability as a percentage return that is directly comparable to interest rates. So a project that showed an estimated IRR of 10 % would be worthwhile if capital could be borrowed for less money than 10 % or if the capital could not be invested elsewhere with a return of more than 10 %.
- The IRR is calculated as the percentage discount rate that would produce an NPV of zero. It's the easiest calculated using a spreadsheet or other computer program that provides IRR calculation functions. For example, Microsoft Excel provides IRR functions that are provided with an initial estimate or initial value (which can be zero), finds and returns the IRR.
- One shortcoming of the IRR is that it does not indicate the absolute size of the return. Project with NPV 100,000 and an IRR of 15 % may be more attractive than a product with an NPV of 10,000 and an IRR of 18 % - the return on capital is lower, but the net benefits are greater.
- Another objection to the internal rate of return is that under certain conditions it is possible to find more than one rate that produces zero NPV. However, if there are multiple solutions, it is always advisable to take them the lowest value and ignore the others.

- However, NPV and IRR are not the complete answer to the economic evaluation of a project.
- The overall assessment must also consider cash flow financing issues - we will be in favor of, for example, be able to repay the interest on any money borrowed in due course ?
- While a project's IRR may indicate a profitable project, future profits from a relatively risky project they can be much less reliable than the earnings of say, an investment in a bank. We could do more a detailed risk analysis as described below.
- We must also consider any individual project within the financial and economic framework of the organization as a whole - If we fund this one, will we be able to fund other worthwhile projects ?

Review Questions

1. What is project evaluation . Which different methods can be used for to evaluate your project ?
2. Explain project evaluation with project evaluation steps in detail.
3. What is the importance of project evaluation.
4. Explain Cost - Benefit analysis with example.
5. Explain cost benefit analysis with terms net profit, Payback time, Return on investment, Net present value, Internal rate of return.

2.3 Process Evaluation and Improvement

- Software process assessment is a disciplined examination of software processes used by an organization based on a process model. The assessment includes identifying and characterizing current practices, identifying areas of strength and weakness and the ability of current practices to control or avoid significant causes of poor (software) quality, cost and schedule.

Software evaluation (or audit) can be of three types

1. Self-assessment (first-party assessment) is carried out internally by the organization's own employees.
2. The evaluation, on the other hand, is performed by an external evaluation team or the organization is evaluated by the customer.
3. A third-party assessment is carried out by an external party or (e.g. a supplier assessed by a third party to verify its ability to contract with a customer).

- Software process evaluation is done in an open and collaborative environment. They are intended for the needs of the organization to improve its software processes and the results are confidential to the organization. The assessed organization must have members of the assessment team.

Software process maturity assessment

- The scope of a software process assessment can cover all processes in an organization, a selected subset of software processes or a specific project. Most standards-based process assessment approaches are always based on the concept of process maturity.
- If the goal is to evaluate an organization, the results of a process evaluation may vary, even with successive applications of the same method. There are two reasons for the different results. They are,
- The organization under investigation must be specified. For a large company, several definitions of organization are possible, so the actual scope of the assessment may vary in successive assessments. Even in what appears to be the same organization, the sample of projects chosen to represent the organization can affect the scope and outcome. If the target unit of evaluation is at the project level, the evaluation should include all meaningful factors that contribute to the success or failure of the project. It should not be limited by the set dimensions of a given process maturity model. Here, the degree of implementation and their effectiveness documented by project data is assessed.
- Process maturity becomes relevant when an organization intends to embark on an overall long-term improvement strategy. Evaluation of software projects should be independent to be objective.

2.3.1 Software Process Evaluation Cycle

- According to Paulek and colleagues (1995), the CMM-based evaluation approach uses a six-step cycle. They are -
  - Select a team - Team members should be professionals with software engineering and management knowledge.
  - Representatives of the evaluated site will fill out a standard process maturity questionnaire.
  - The evaluation team will analyze the questionnaire responses and identify areas that require further investigation according to the key areas of the CMM process.
  - The evaluation team will conduct a site visit to understand the software process that the site follows.

  - The assessment team creates a list of findings that identify the strengths and weaknesses of the organization's software process.
  - The evaluation team will prepare a Key Process Area (KPA) profile analysis and present the results to the appropriate audience.
- For example, the assessment team must be led by an authorized SEI lead assessor. The team must consist of four to ten team members. At least one team member must be from the assessed organization and all team members must have completed the CMM SEI Introduction Course (or equivalent) and the CBA IPI SEI Team Training Course. Team members must also meet certain selection rules.
- In terms of data collection, the IPI CBA relies on four methods -
  1. Standard adulthood questionnaire
  2. Individual and group interviews
  3. Document review
  4. Feedback from review of draft findings with evaluators.

SCAMPI

- The Standard CMMI Assessment Method for Process Improvement (SCAMPI) was developed to meet the requirements of the CMMI model (Software Engineering Institute, 2000). It is also based on CBA IPI.
- Both the IPI and SCAMPI CBAs consist of three phases -
  1. Plan and prepare
  2. Do an on-site assessment
  3. Report results.
- Activities for the planning and preparation phase include -
  - Determine the scope of the assessment
  - Develop an assessment plan
  - Prepare and train the evaluation team
  - Conduct a brief participant assessment
  - Administer the CMMI assessment questionnaire
  - Explore the answers to the questionnaire
  - Perform an initial document review.

- Activities for the on-site assessment phase include -
  - Hold a kick-off meeting
  - Conduct interviews
  - Consolidate information
  - Prepare a presentation of draft findings
  - Submit a draft finding
  - Consolidate, evaluate and prepare final findings.
- Results reporting phase activities include -
  - Present final findings
  - Conduct an executive session
  - End the evaluation.
- CMM (Capability Maturity Model) and CMMI (Capability Maturity Model Integration).
- CMM was developed by SEI (Software Engineering Institute) and later evolved into CMMI. It is an approach based on which the process maturity of the organization is determined.

Five levels of CMM maturity :

- **Entry level :** The processes are not organized and the success of the project depends only on the competence of the individual working on it. He may not be able to repeat past successes in future projects. The probability of exceeding the estimated cost and schedule is high.
- **Repeatable level :** At this level, past successes could be repeated because the organization uses project management techniques to track costs and schedule. Managing according to a documented plan helps to improve the process.
- **Defined level :** A set of standard organizational processes is defined and slightly modified to include the requirements of each project. This ensures consistency throughout the work of the organization.
- **Controlled level :** Controlling processes using quantitative techniques improves performance. Processes are evaluated through data collection and analysis.
- **Level of optimization :** Processes are monitored and improved through feedback from current work. Innovative techniques are used to deal with the changing business objectives and environment.

- CMMI maturity levels include :
  - Initial
  - Managed
  - Defined
  - Quantitative management
  - Optimized.
- CMMI competency levels include :
  1. **Level 0 :** Incomplete - Incomplete processes are processes that are not performed or partially performed.
  2. **Level 1 :** Met - Processes satisfy specific objectives, yet certain quality, cost and schedule objectives are not met. Useful work can be done.
  3. **Level 2 :** Controlled - cost, quality and schedule are controlled and processes are monitored using managerial techniques.
  4. **Level 3 :** Defined - Includes management and in addition follows a set of standard processes specified by the organization that change for each project.
  5. **Level 4 :** Quantitative Management - Statistical and quantitative techniques are used to control processes.
  6. **Level 5 :** Optimized - Focuses on continuous improvement of the quantitatively driven process through innovation and the nature of the processes.

CMMI Standard Assessment Method for Process Improvement (SCAMPI)

- It is a method used by the Software Engineering Institute (SEI) to provide quality assessments with respect to Capability Maturity Model Integration (CMMI). Assessment includes the five stages of initiation, diagnosis, establishment, action and learning. The evaluation process includes preparation, on-site activities, findings and evaluation, final report, etc.

CMM-Based Assessment for Internal Process Improvement (CBA IPI)

- It is an assessment method based on the SEI CMM (Capability Maturity Model) that diagnoses, enables and encourages an organization to understand its maturity. It gives an organization insight into its software development capability by assessing the strength and weakness of the current process.

SPICE (ISO/IEC15504)

- This standard is one of the joint tasks of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). They assist organizations in creating an objective assessment of the effectiveness of the software process and related business management functions.
- It consists of six levels as :

1. Not implemented.
2. Done informally.
3. Planned and monitored.
4. Well defined.
5. Quantitative control.
6. Continuously improved.

ISO 9001 : 2000 for software

- It is used for organizations that aim to improve the overall quality of products, processes and services. It assesses an organization's ability to consistently provide products that meet customer requirements. Here, the main goal of the organization should be to increase customer satisfaction.
- It follows the Plan Do Check Act (PDCA) cycle, which includes :
  - Planning by defining the processes and their needs needed to develop a better product.
  - Carrying out the necessary actions according to the plan.
  - Checking whether quality assurance measures are met according to requirements.

Review Question

1. What is process evaluation ? Explain software process evaluation cycle.

2.4 Process Measurement

- A measurement is an expression of the size, quantity, amount or dimension of a particular attribute of a product or process. Software measurement is the titration imputation of a characteristic of a software product or software process. He is an authority on software engineering. The software measurement process is defined and controlled by the ISO standard.

2.4.1 Process Measurement Principles

- The software measurement process can be characterized by five activities -

1. **Formulation :** Deriving software measures and metrics suitable for representing the software under consideration.
2. **Collection :** The mechanism used to collect the data needed to derive the formulated metrics.
3. **Analysis :** Calculation of metrics and application of mathematical tools.
4. **Interpretation :** Evaluation of matrics lead to increases the quality of representation.
5. **Feedback :** Recommendations derived from the interpretation of product metrics provided to the software team.

The need for software measurement :

- The software is measured on :
  - Establish the quality of the current product or process.
  - To predict the future quality of a product or process.
  - Improve the quality of a product or process.
  - Regulate project status in relation to budget and schedule.

2.4.2 Software Measurement Classification

- There are two types of software metering :

1. **Direct measurement :** In direct measurement, a product, process or thing is measured directly using a standard scale.
2. **Indirect measurement :** In indirect measurement, quantity or quality is measured using related parameters, i.e. using a reference.

2.4.3 Metrics

- A metric is a measure of the level at which any impute belongs to a system product or process.
- Process metrics will only be useful if they are effectively characterized and validated to demonstrate their value.

- There are four functions related to process metrics :
  - Planning
  - Organizing
  - Control
  - Improving.

2.4.3.1 Metrics Software Features

- **Quantitative :** Metrics must be quantitative in nature. This means that metrics can be expressed in values.
- **Understandable :** The calculation of metrics should be easy to understand and the method of calculating metrics should be clearly defined.
- **Usability :** Metrics should be usable in the early stages of software development.
- **Repeatable :** Metric values should be the same and consistent when measured repeatedly.
- **Economical :** Calculation of metrics should be economic.
- **Language independence :** Metrics should not depend on any programming language.

2.4.3.2 Classification of Process Metrics

- There are three types of process metrics :

1. **Product metrics :** Product metrics are used to evaluate the health of the product, monitor risks and uncover potential problem areas. The team's ability to control quality is assessed.
2. **Process metrics :** Process metrics pay special attention to improving the long-term process of a team or organization.
3. **Project metrics :** The project matrix describes the project characteristics and execution process.
   - Software developer number
   - Staffing patterns throughout the software life cycle
   - Cost and schedule
   - Productivity.

2.4.4 The GQM Paradigm

- The Goal/Question/Metric (GQM) method is a proven technique used for goal-oriented measurement. It consists of the following three basic elements :

  1. Soccer goal 2. Question 3. Metric

- In the GQM method, measurement is goal-oriented. First, the objectives must be clearly described so that they can be measured during software development.
- In this method, goals are defined that are transformed into questions and metrics. Then the questions are answered and it is checked whether these answers meet the objectives or not. So this method follows a top-down approach by dividing the objectives and then mapping the objectives into questions and then these questions are transformed into metrics and the method also follows a bottom-up approach by analyzing the measurements and checking whether the objectives are met or not.

2.4.4.1 Steps in the GQM Method

1. Objectives are clearly defined and described.
2. Transforming objectives into appropriate questions.
3. Questions are transformed into metrics.

2.4.4.2 Stages of the GQM Method

- **Planning :** In the first phase, a project plan is prepared by identifying the basic requirements.
- **Definition :** In the second phase, the objectives, questions and metrics are described. These three are clearly defined.
- **Data collection :** Current data is being collected in this phase.
- **Interpretation :** It is the last stage in which answers to the questions posed in the previous stages are provided and the achievement of the goal is verified.

Review Questions

1. Define process measurement. What is need for process measurement ?
2. What is metrics ? Explain different features of metrics software. Explain different types of process metrics.
3. Write a short note on : The GQM Paradigm.

2.5 Process Analysis

- An analytical model is a technical representation of a system. It acts as a link between the system description and the design model. In analytical modeling, system information, behavior and functionality are defined and translated into architecture-level design, components and interfaces in design modeling.

2.5.1 Objectives of Analytical Modeling

- It must establish a way of creating a software design.
- It must describe the customer's requirements.
- It must define a set of requirements that can be verified once the software is built.

2.5.2 Elements of the Analytical Model

- **Data dictionary :** It is a repository that consists of a description of all data objects used or created by the software. It stores the collection of data present in the software. It is a very important element of the analytical model. It acts as a centralized repository and also helps in modeling data objects defined during software requirements.
- **Entity Relationship Diagram (ERD) :** It shows the relationship between data objects and is used when performing data modeling activities. The attributes of each object in the Entity-Relationship Diagram can be described using a data object description. It provides a foundation for data design activity.
- **Data Flow Diagram (DFD) :** It describes the functions that transform the data stream and also shows how the data is transformed as it goes from input to output. It provides additional information that is used in the analysis of the information domain and serves as a basis for modeling the function. It also allows the engineer to develop functional and information domain models simultaneously.
- **State transition diagram :** It shows different ways of behavior (states) of the system and also shows the transitions from one state to another in the system. It also provides details of how the system behaves as a result of the consequences of external events. It represents the behavior of the system by presenting its states and the events that cause the system state to change. It also describes what actions are taken as a result of the occurrence of a specific event.
- **Process specifications :** Stores a description of each function present in the data flow diagram. It describes the input to the function, the algorithm that is applied to transform the input and the output that is produced. It also shows the regulations and constraints placed

on the performance characteristics that apply to the process and the design constraints that could affect how the process will be implemented.

- **Control specifications :** Stores additional information about the control aspects of the software. It is used to indicate how the software behaves when an event occurs and which processes are invoked as a result of the occurrence of the event. It also provides details of the processes that are performed to manage events.
- **Description of the data object :** It stores and provides complete knowledge about the data object present and used in the software. It also gives us details about the attributes of the data object present in the entity relationship diagram. So it includes all data objects and their attributes.

2.5.3 Process Analysis Techniques

1. Business Process Modeling Notation (BPMN)

- BPMN (Business Process Modeling and Notation) is a graphical representation of your business process using simple objects that helps the organization communicate in a standard way. The various objects used in BPMN include :
  - Stream objects
  - Connecting objects
  - Swimming lanes
  - Artifacts
- A well-designed BPMN model should be able to provide details about the activities performed during the process, such as,

Who performs these activities ?

- What data elements are required for these activities ?
- The biggest advantage of using BPMN is easier sharing and most modeling tools support BPMN. (Refer Fig. 2.5.1 on next page)

2. UML (Unified Modeling Language)

- UML is a modeling standard primarily used for the specification, development, visualization and documentation of a software system. To capture important business processes and artifacts, UML provides objects like,
  - State
  - Object
  - Activity
  - Class diagram

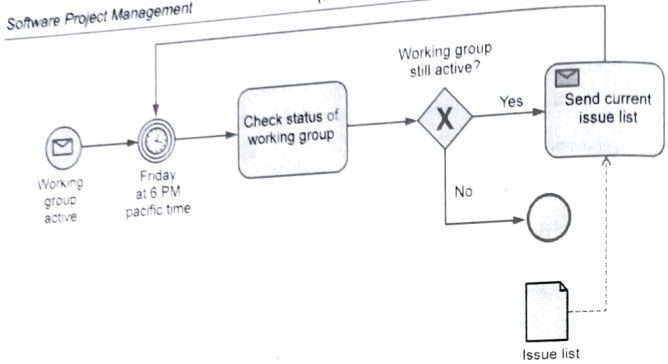

Fig. 2.5.1

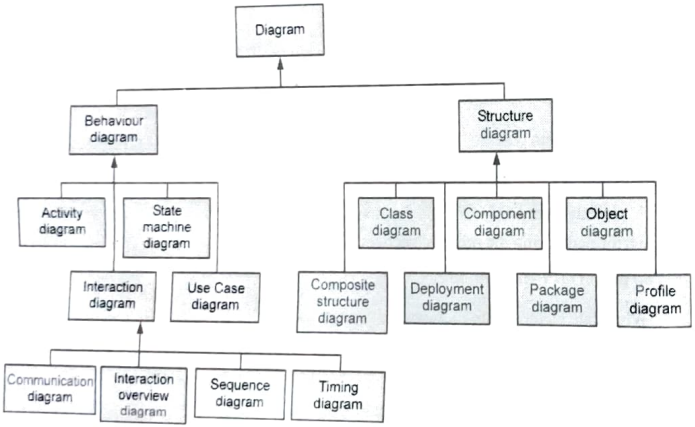

Fig. 2.5.2 : UML - class diagram

- There are 14 UML diagrams that help in modeling such as use case diagram, interaction diagram, class diagram, component diagram, sequence diagram, etc. UML models are important in the IT segment as they become a means of communication between all stakeholders. A UML-based business model can be a direct input to a requirements tool. UML diagram can be of two types : Behavioral model and structural model. A behavioral model tries to give information about what the system does, while a structural model gives what the system consists of.

3. Flow chart technique

- A flowchart is a visual representation of the sequential flow and control logic of a set of related activities or actions. There are different flowchart formats, which include linear, top-down and cross functional (swim lanes). A flowchart can be used for various activities such as representing data flows, system interactions, etc. The advantage of using a flowchart is that it is easy to read and write even for non-technical team members and can show a parallel process by function, critical process properties, etc.

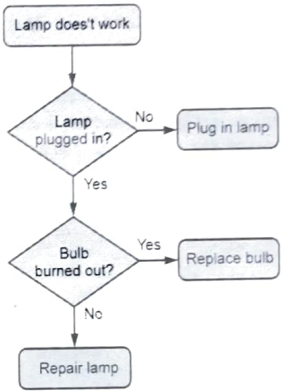

Fig. 2.5.3 : Flow chart

4. Data flow diagram

- Data flow diagrams show how the system processes data in terms of inputs and outputs. The components of a data flow diagram include,

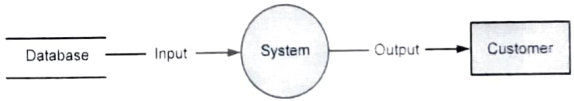

Fig. 2.5.4 : Data flow diagram

- A logical data flow diagram shows the system activities while a physical data flow diagram shows the system infrastructure. A data flow diagram can be designed early in the requirements elicitation process in the analysis phase of the System Development Life Cycle (SDLC) to define the scope of the project. For easy analysis, a data flow diagram can be broken down into its sub-processes known as a "levelled DFD".

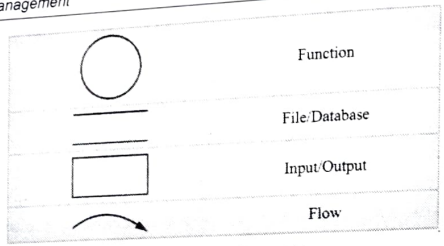

Fig. 2.5.5 : Symbols notations

5. Role Activity Diagrams - (RAD)

- A role activity diagram is similar to a flowchart type notation. In a role activity diagram, role instances are participants in a process that has a start state and an end state. RAD requires deep knowledge of the process or organization to identify roles. RAD components include,
  - Activities
  - External events
  - States

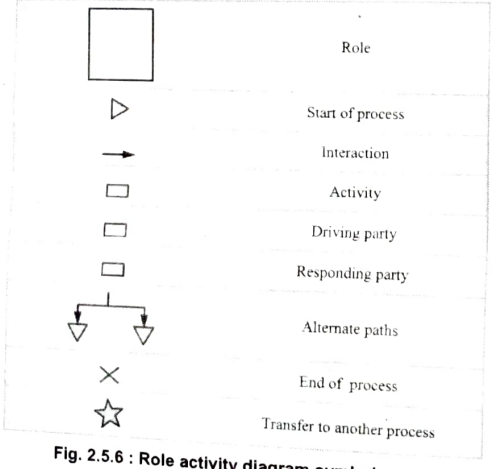

Fig. 2.5.6 : Role activity diagram symbols

- Roles group activities into units of responsibility according to the set of responsibilities they perform. An activity may be performed in isolation with a role or may require co-ordination with activities in other roles.
- External events are points at which state changes occur.
- States are useful for mapping the activities of a role as it progresses from state to state. When a certain state is reached, it means that a certain goal has been achieved.
- The RAD is helpful in supporting communication as it is easy to read while providing a detailed view of the process and permitting activities.

6. Gantt charts

- A Gantt chart is a graphical representation of a schedule that helps coordinate, plan and track specific tasks in a project. It represents the total time span of the object, divided into increments. A Gantt chart presents a list of all tasks to be performed on the vertical axis, while the horizontal axis shows the estimated duration of the activity or the name of the person assigned to the activity. A single chart can demonstrate many activities.

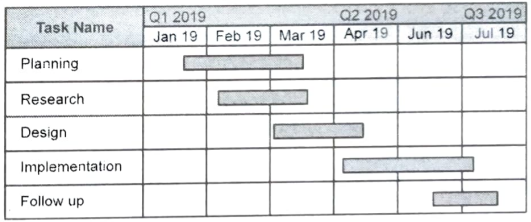

Fig. 2.5.7 : Gantt charts

7. IDEF (Integrated Definition for Functional Modeling)

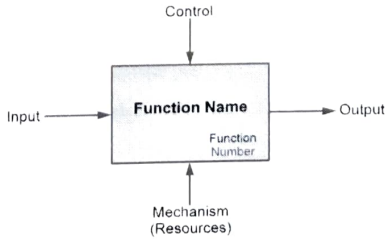

Fig. 2.5.8 : IDEF

- IDEF or Integrated Definition for Functional Modeling is the common name referred to classes of enterprise modeling languages. It is used to model activities necessary to support system analysis, design or integration. There are about 16 methods for IDEF, the most useful versions of IDEF are IDEF3 and IDEF0.

8. Colored Petri Nets (CPN)

- CPN or Colored Petri Nets, is a graphically oriented language for the specification, verification, design and simulation of systems. Colorful Petri nets are a combination of graphics and text. Its main components are places, transitions and arcs.
- Petri net objects have a specific inscription such as :
  - **Places :** Has inscription like .Name, .Color Set, Initial marking etc. While
  - **Transition :** Has inscription like Name (for identification) and .Guard (Boolean expression consists of some variables).
  - **Arcs :** Labeled as .Arc. When an arc expression is evaluated, it returns multiple sets of token colors.

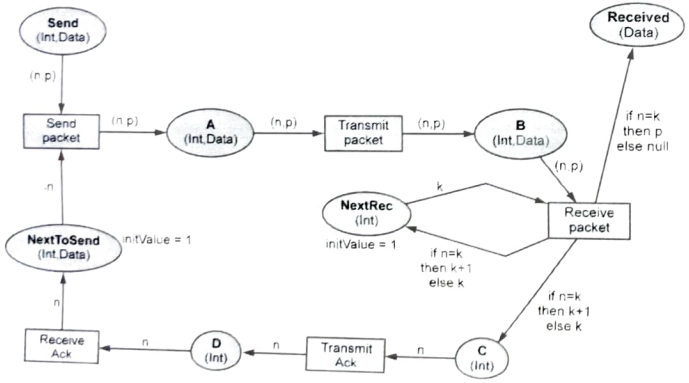

Fig. 2.5.9 : Colored Petri Nets (CPN)

9. Workflow technique

- A workflow technique is a visual diagram that represents one or more business processes to clarify an understanding of the process or make recommendations for process improvement. Like other diagrams such as flowchart, UML activity and process map, the workflow technique is the oldest and popular technique. It is even used by BA to write down notes when calling requests. The process consists of four stages :

1. Collection of information
2. Workflow modeling
3. Business process modeling
4. Implementation, verification and execution.

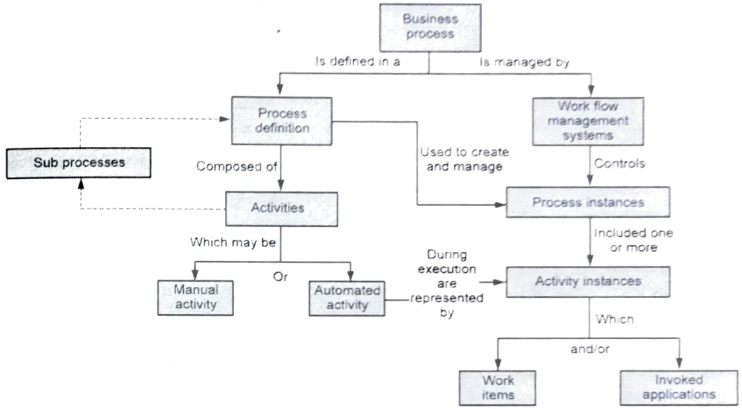

Fig. 2.5.10 : Workflow technique

10. Object-oriented methods

- The object-oriented modeling method uses an object-oriented paradigm and a modeling language for system design. Emphasizes finding and describing the object in the problem domain.
- The purpose of the object-oriented method is to,
  - To help characterize the system
  - To know what are the various relevant objects
  - How they are related
  - How to specify or model a problem to create an effective design
  - Analyze requirements and their implications.
- This method is applicable to a system that has dynamic (often changing) requirements. It is the process of deriving use cases, flow of activities and flow of events for a system. Object-oriented analysis can be done through textual needs, system stakeholder communication and a vision document.

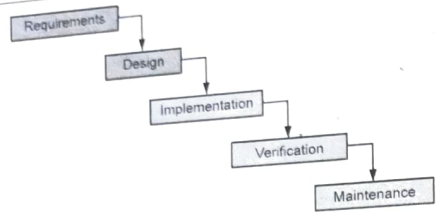

Fig. 2.5.11

- An object has a state and changes to the state are represented by behavior. So when an object receives a message, the state is changed by the behavior.

11. Gap analysis

- Gap analysis is a technique used to determine the difference between the proposed state and the current state for any business and its functions. It answers questions like what is the current status of the project ? Where do we want to be etc. Different stages of gap analysis include.
  - Control system
  - Development requirements
  - Comparison
  - Impacts
  - Recommendation.

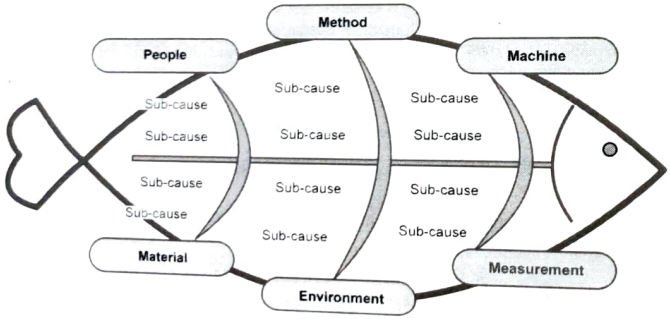

Fig. 2.5.12

Review Questions

1. Define process analysis. Explain different techniques of process analysis.
2. What is process analysis? Explain any 5 process analysis techniques.
3. List elements of the Analytical model / Process model.

2.6 Process Change

- Change management in software development refers to the transition from the current state of a software product to another improved state of the product. Manages, supports and manages artifact changes such as code changes, process changes or documentation changes. Where CCP (Change Control Process) mainly identifies, documents and authorizes changes to a software application.
- Every software development process follows a Software Development Life Cycle (SDLC) where each phase is followed accordingly to finally deliver a quality software product. Change management does not fall into any phase of the SDLC, yet it is of great importance in the entire software development process. There are different types of change management tools that are used for different purposes such as accepting, controlling, representing and executing the desired change. For example, change management tools for flowcharts, project planning, data collection, etc.

2.6.1 Change Management Process

- When any software application/product undergoes changes in the IT environment, it undergoes a number of the following processes :
  - Create a change request
  - Review and assessment of the change request
  - Planning for change
  - Testing the change
  - Create a change proposal
  - Making changes
  - Check performance of changes
  - Closing the process.

2.6.2 Importance of Change Management

- To improve performance
- To increase engagement
- To boost innovation
- For the inclusion of new technologies
- To implement new requirements
- To reduce costs.

2.6.3 Source of Change

- During the development process, there may be several reasons why it is necessary to implement certain changes to the product. These resources are as follows :
  - Business reorganization
  - New market conditions
  - New equipment
  - Correcting any bugs/errors
  - New customer needs
  - Improving performance or reliability
  - Budget or scheduling constraints.

2.6.4 Key Points to Consider When Managing Change

- Reason for change
- The result of the change
- The part to be changed
- A person changes
- Risks associated with change

- An alternative to change
- Resources needed for change
- Relationship between changes.

2.6.5 Steps to Process Change

1. Identify the change

- Identify the type of change. Describe it using a set of features such as name, type, scope of change and project identifier. This will make the change more understandable for everyone involved in your project.

2. Assess the impact

- Assess the risk the change poses and the impact it will have on the software, the team working on it and those who will use it. Evaluate the technical aspects of the change, its potential side effects and the overall impact on other parts of the system. Consider any changes to the project budget, timeline and resources (both human and IT) needed to implement the change.

3. Decide to implement the change

- Decide to implement the change or get approval to implement it. It is extremely important to know who will make the decisions and how the change will be made. You need to discuss this process at the very beginning of the project so that you know how to handle change requests when they arise. When making decisions, make sure your software development team is involved and informed at every stage of the process.

4. Plan to roll out the change

- If necessary, revise the original project plan, including the impact of each proposed change on the software and any components that may need to be created or removed to implement the change. Include information about any new deadlines or backlogs that the entire organization may face when implementing the change. Also think about other challenges that the change may bring, such as adjusting the workload for all employees.

Review Question

1. Explain process change with following points :
   A. Change management process B. Importance
   C. Source of change D. Steps to process change

2.7 Case Study : Online Shopping System

2.7.1 Use Case Diagram

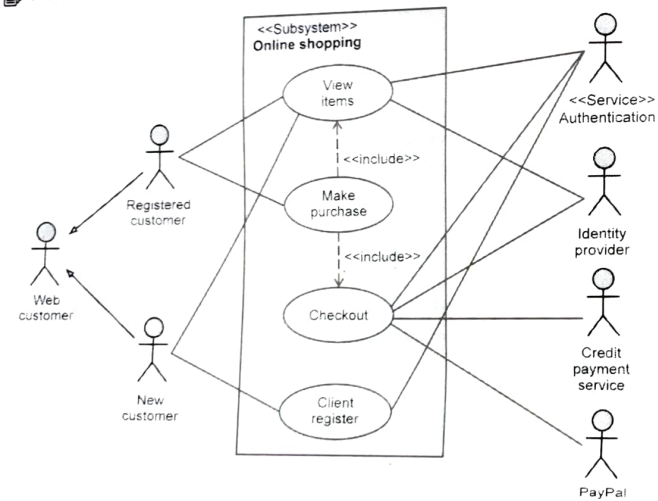

Fig. 2.7.1 : Use case diagram

2.7.2 Class Diagram

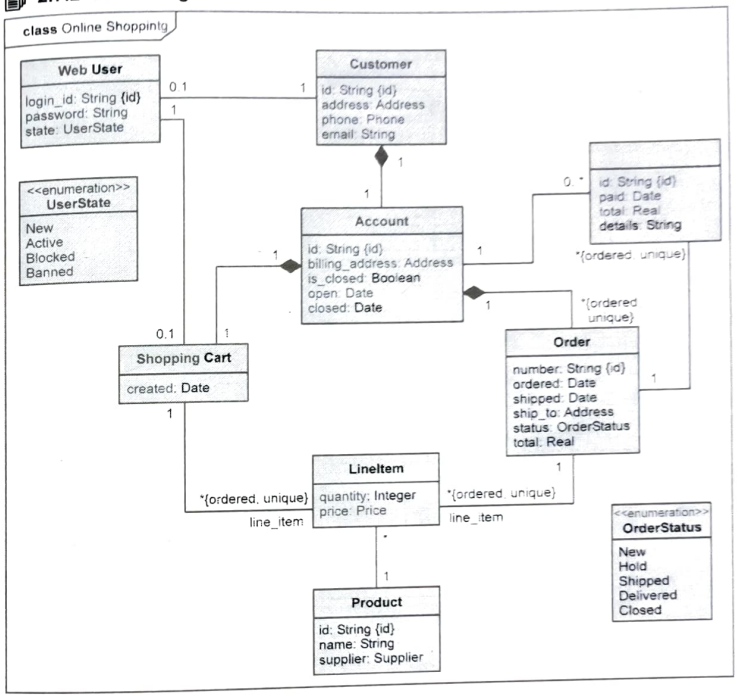

Fig. 2.7.2 : Class diagram

2.7.3 User Side Activity Diagram

- In User side activity diagram, it describes all the functionality or operation of users can do on our website.

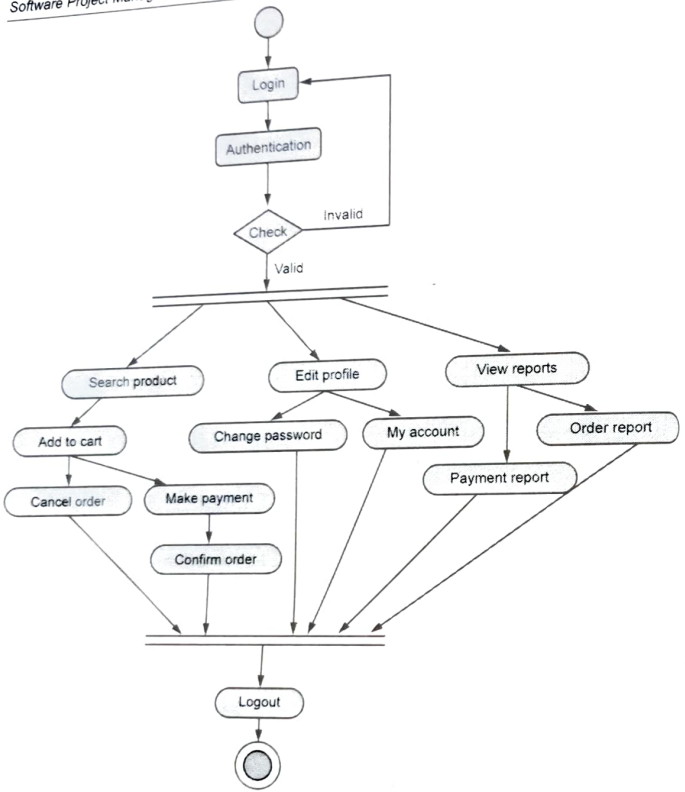

Fig. 2.7.3 (a) : Activity diagram for user side

Activity diagram for online shopping website - admin side

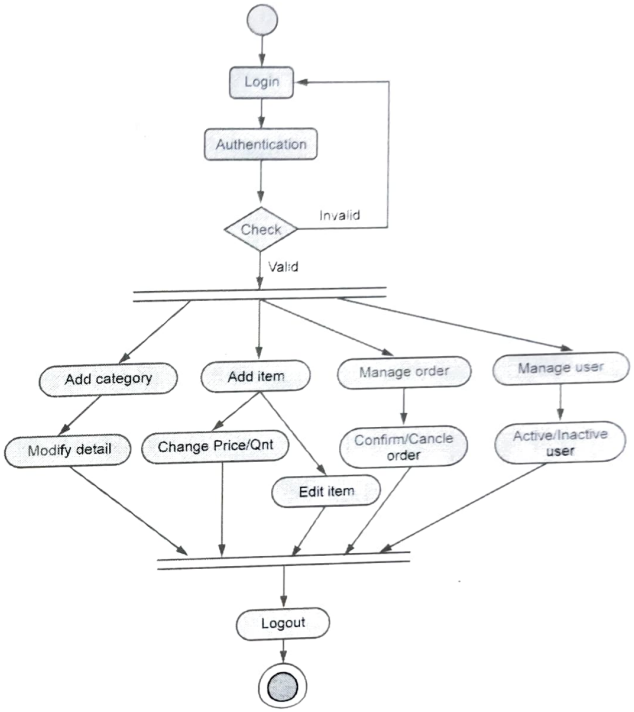

Fig. 2.7.3 (b) : Activity diagram for admin side

2.7.4 State Diagram

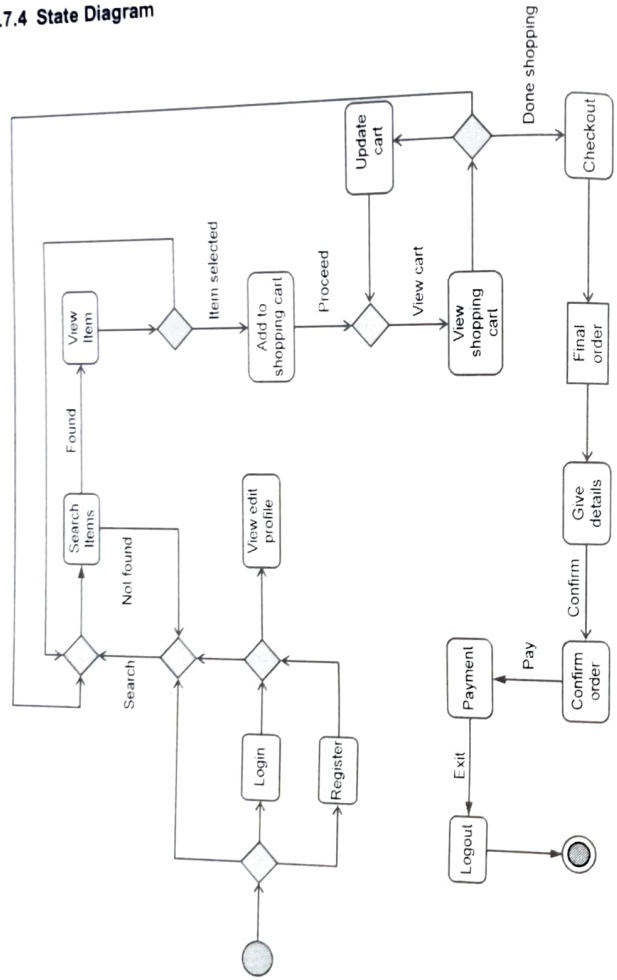

Fig. 2.7.4 : State diagram online shopping system

2.7.5 Sequence Diagram

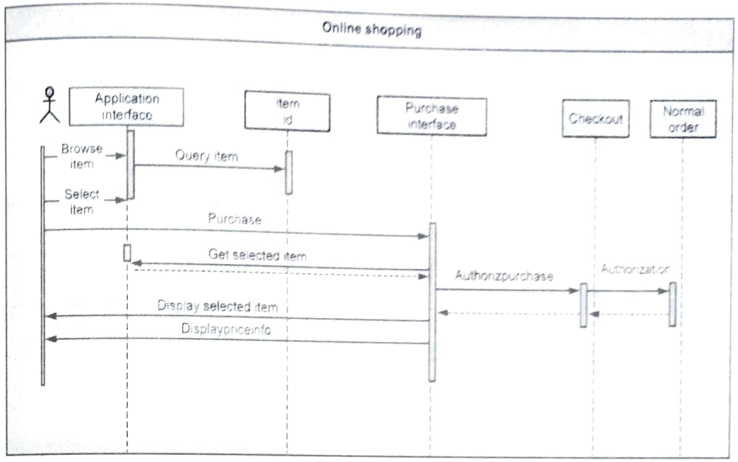

Fig. 2.7.5 : Sequence diagram

❑❑❑