OOSE MODULE 1 SOLUTIONS

Part A

1 Develop a set of actions for the communication activity. Select one action and define a task set for it.

A. Task Set for Communication Activity: A task set would define the actual work to be done to accomplish the objectives of a software engineering action. For the communication activity these are:

- Make a list of stakeholders for the project
- Invite all the stakeholders to an informal meeting
- Ask them to make a list of features and functions
- Discuss requirements and build a final list
- Prioritize requirements and note the areas that he is uncertain of These tasks may be larger for a complex software project, they may then include

- To conduct a series of specification meetings, build a preliminary list of functions and features based on stakeholder input.
- To build a revised list of stakeholder requirements.
- Use quality function deployment techniques to prioritize the requirements.
- Note constraints and restrictions on the system.
- Discuss methods for validating systems.

2 Describe, Is it possible to combine process models? If so, provide an example

A. Yes, it is possible to combine the software process models. The process models can be combined because all the processes are about similar to each other as all the models have these basic phases like: requirements gathering, planning, designing, implementation and deployment but each model suggests the different process flow. Some of the possibilities of combining the software process models are evolutionary / prototyping process models, the spiral model and the incremental process model.

We can sometimes combine process models:
e.g.

1. waterfall inside evolutionary – onboard shuttle software.
2. Evolutionary inside waterfall – e.g. GUI prototyping
3. Scrum inside waterfall

We can also evolve the process together with the product to account for product maturity.
e.g. rapid prototyping waterfall.
Following software:

1. Likewise agile (combination of both incremental and iterative model)
2. Spiral (combination of waterfall and evolutionary model)

3 List the advantages and disadvantages of developing software in which quality is —good enough

A. The advantages of developing software in which quality is “good enough” are:

- Completeness – All the requirements are reflected in the software
- Conciseness – Compactness
- Reliability – No faulty outputs
- Improved user satisfaction.
- Reduced cost of maintenance
- Efficiency – Amount of computing resources and cost required by a program to perform a function.
- Consistency.

Disadvantage -High Costs. The obvious disadvantage of custom software is the high upfront cost. ...

Time Losses on Requirements Gathering. ...

The Danger of Choosing the Wrong Vendor.

Advantages-Optimizes Your Business Processes. Each business has its model and in-house processes. ...

Helps You Save on Cost. ...

Offers Competitive Advantage. ...

A Custom Software is Highly Adaptable. ...

Offers an Exclusive Solution.

4 Explain why systems developed as prototypes should not normally be

used as production systems.

A. Prototypes should be discarded after development as they are not a good basis for a production system:

1. It may be impossible to tune the system to meet non-functional requirements;
2. Prototypes are normally undocumented;
3. The prototype structure is usually degraded through rapid change;
4. The prototype probably will not meet normal organizational quality standards.
5. Ignorance of some non-functional requirements like user management, security, memory management, and performance may not be fulfilled.

5 List the prescriptive software development process models. Explain the incremental process model with neat diagram

A. The name 'prescriptive' is given because the model prescribes a set of activities, actions, tasks, quality assurance and changes the mechanism for every project.

There are three types of prescriptive process models. They are: 1. The Waterfall Model

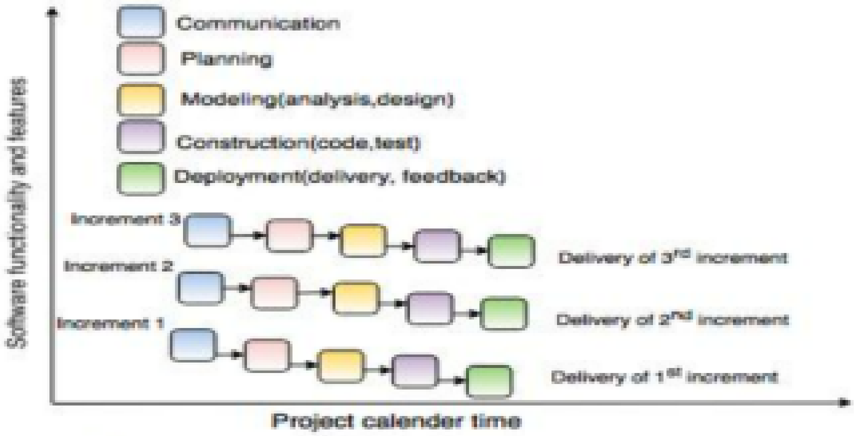
2. Incremental Process model

3. RAD model

Incremental Process model

- The incremental model combines the elements of the waterfall model and they are applied in an iterative fashion.
- The first increment in this model is generally a core product.
- Each increment builds the product and submits it to the customer for any suggested modifications.
- The next increment implements the customer's suggestions and adds additional requirements in the previous increment.
- This process is repeated until the product is finished.

For example, word-processing software is developed using the incremental model.



Communication
Planning
Modeling(analysis,design)
Construction(code,test)
Deployment(delivery, feedback)
Increment 3
Increment 2
Increment 1
Delivery of 3rd increment
Delivery of 2nd increment
Delivery of 1st increment
Software functionality and features
Project calender time

Fig. - Incremental Process Model

Advantages of incremental model

- This model is flexible because the cost of development is low and initial product delivery is faster.
- It is easier to test and debug during the smaller iteration.
- The working software generates quickly and early during the software life cycle.

- The customers can respond to its functionalities after every increment.

Disadvantages of the incremental model

- The cost of the final product may cross the cost estimated initially.
- This model requires very clear and complete planning.
- The planning of design is required before the whole system is broken into small increments.
- The demands of customers for the additional functionalities after every increment causes problems during the system architecture.

6 List out any three specialized process models. Explain the component based development process model with their goals, advantages and routines.

A. The three specialized process models are:

- Component Based Development
- The Formal Methods Model
- Aspect Oriented Software Development

Component Based Development

The component based development model incorporates many of the characteristics of the spiral model. It is evolutionary in nature, a specialized process model demanding an iterative approach to the creation of software. However, the component based development model constructs applications from prepackaged software components.

Modeling and construction activities begin with the identification of candidate components. These components can be designed as either conventional software modules or object oriented classes or packages of classes.

Regardless of the technology that is used to create the components, the component based development specialized process model incorporates the following steps.

Available component based products are researched and evaluated for the application domain in question. Component integration issues are considered. A software architecture is designed to accommodate the components. Components are integrated into the architecture. Comprehensive testing is conducted to ensure proper functionality.

The component based development model leads to software reuse, and reusability provides software engineers with a number of measurable benefits.

7 Distinguish between process and project metrics.Give examples? What is defect classification? How can an organization make use of this metrics

for its process improvement?

Software process and project metrics are quantitative measures that enable software engineers to gain insight into the efficiency of the software process and the projects conducted using the process framework. In software project management, we are primarily concerned with productivity and quality metrics.
Process metrics – These characteristics can be used to improve the development and maintenance activities of the software.

Project metrics – This metrics describe the project characteristics and execution. Examples include the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity.

Process Metrics

- Private process metrics (e.g. defect rates by individual or module) are known only to the individual or team concerned.

- Public process metrics enable organizations to make strategic changes to improve the software process.

- Metrics should not be used to evaluate the performance of individuals.

• Statistical software process improvement helps an organization to discover where they are strong and where they are weak.

Project Metrics

• Software project metrics are used by the software team to adapt project workflow and technical activities.
• Project metrics are used to avoid development schedule delays, to mitigate potential risks, and to assess product quality on an on-going basis.

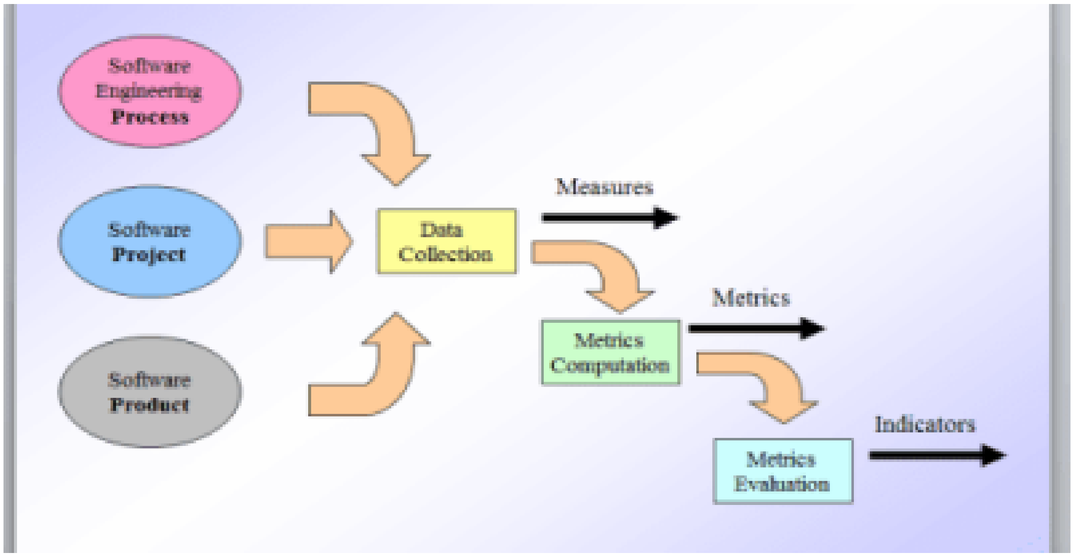
• Every project should measure its inputs (resources), outputs (deliverables), and results (effectiveness of deliverables).

Defects are classified from the QA team perspective as Priority and from the development perspective as Severity (complexity of code to fix it). These are two major classifications that play an important role in the timeframe and the amount of work that goes in to fix defects.

The only way to know how/where to improve any process is to

- Measure specific attributes of the process.
- Develop a set of meaningful metrics based on these attributes. ● Use the metrics to provide indicators that will lead to a strategy for

improvement.



8 Summarize the necessity of different process models? Describe the process model you would adopt for the car manufacturing project and justify your choice with its advantages and disadvantages

A. By using a business process model, everyone within an organization can reap the upsides. From employees who have their hands on deck to external stakeholders who have a vested interest in how things run.

This is because the first step in maximizing productivity and reducing waste is to know how processes currently run. Process models can lead to:

- **Increased efficiency:** Since the purpose of a process model is to make the process better, it should ultimately result in increased efficiency. With better quality outputs, you can boost your bottom line.
- **Standardization:** If different teams have to run the same process, having it visually displayed can aid in standardizing the process.

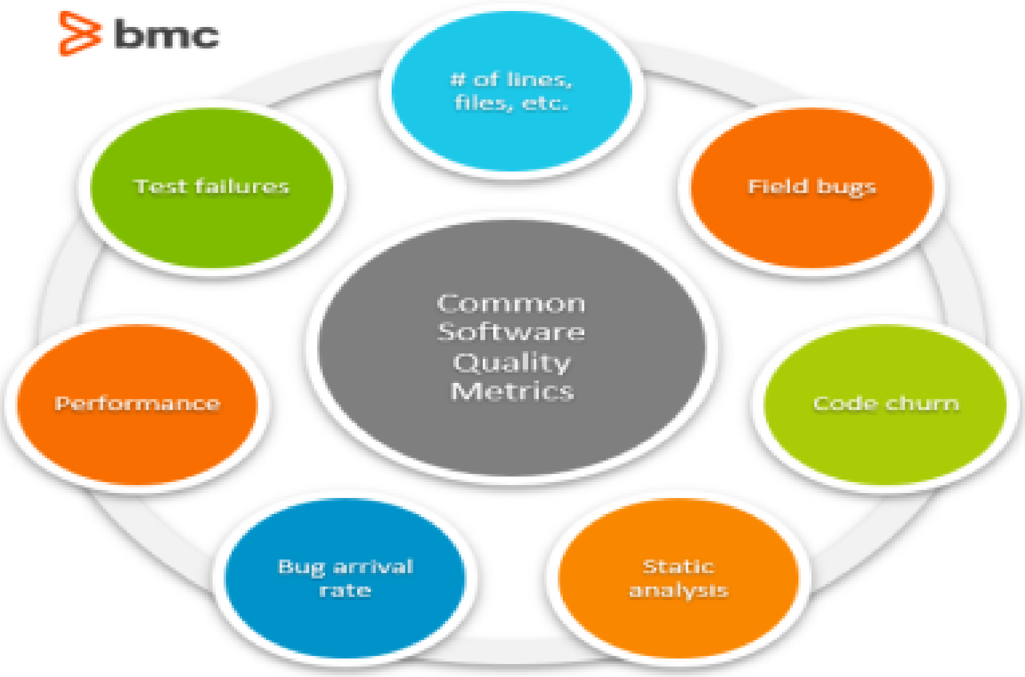
- **Transparency:** With process models, everyone becomes aware of what processes exist and the reasons for them. This not only helps to hold responsible parties accountable, but it can also boost morale when employees understand the bigger picture of why they do what they do.
- **Agility:** Creating an environment where process improvement is a norm will help to shift the entire corporate culture. In a business setting that strives for greatness, it becomes easier to enact change management and adjust processes to keep up with the times.

9. State the needs for metrics in software engineering. Explain how software quality is assured through software metrics.

A. Benefits of Software Metrics

The goal of tracking and analyzing software metrics is to determine the quality of the current product or process, improve that quality and predict the quality once the software development project is complete. On a more granular level, software development managers are trying to:

- Increase return on investment (ROI)
- Identify areas of improvement
- Manage workloads
- Reduce overtime

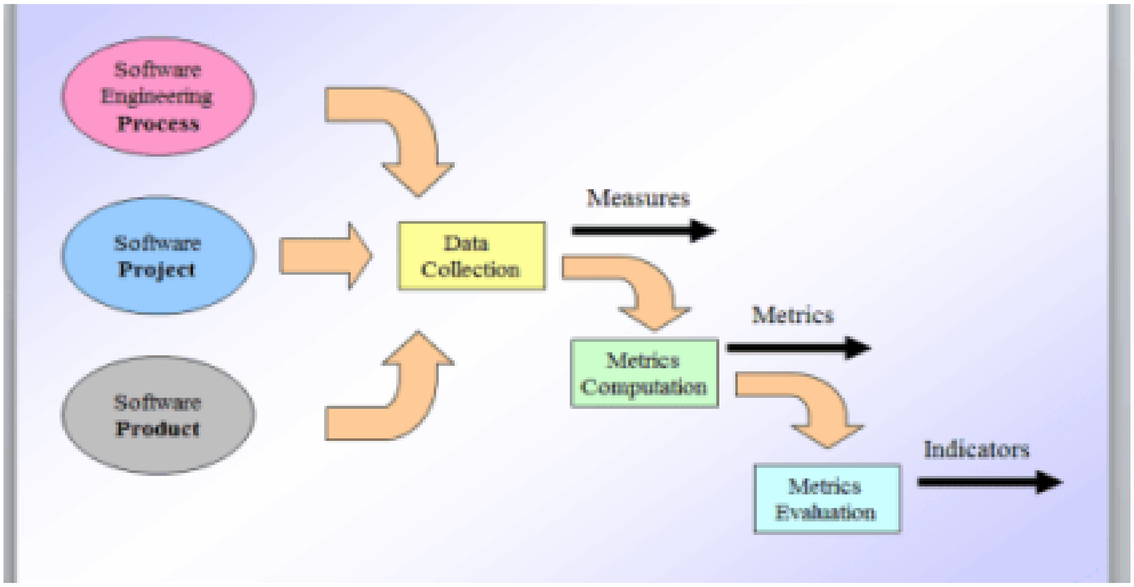


Reduce costs:

The following principles or factors are things that can be measured. Then use the results to test the quality of your software as it applies to the above quality aspects trying to be achieved.

The only way to know how/where to improve any process is to

- Measure specific attributes of the process.
- Develop a set of meaningful metrics based on these attributes.
- Use the metrics to provide indicators that will lead to a strategy for improvement.



10 Describe project management? List and explain the principles related to software project management related to a project.

A.Project management is the process of leading the work of a team to achieve all project goals within the given constraints. This information is usually described in project documentation, created at the beginning of the development process. The primary constraints are scope, time, and budget. The secondary challenge is to optimize the allocation of necessary inputs and apply them to meet pre-defined objectives.

Project management is the use of specific knowledge, skills, tools and techniques to deliver something of value to people.
These project management principles cover the major areas when managing a project. At ProjectManager, we have tons of project management templates, blogs, tools and other resources to help you manage your projects better.

1. Define a Project Organization Structure

This is the first thing you'll have to think about when managing a project. The project organization structure is the framework that facilitates the planning, execution and tracking of project activities.

2. Set Clear Project Goals & Objectives

Before you can start the project planning phase, you'll need to define the main goals and objectives of your project. The project goals define the expected benefits of the project while the project objectives are the steps that you'll need to take to achieve them. Defining your goals and objectives will set the stage to plan your project scope, schedule and budget.

3. Create a Communication Plan

While reporting to the various participants in the project is key, there must be a primary communication plan to regulate communications between yourself and the project sponsor. This is the only way to ensure those project decisions are properly implemented.

4. Define Roles & Responsibilities

To move forward, a project must have well-defined roles, policies and procedures in place. That means everyone must know what they're responsible for and to whom they answer. There needs a delegation of authority for any project to function.

5. Create a Risk Management Plan

Risk is part of life, and it's certainly a part of any project. Before the project even starts, figure out the potential risks inherent in the work ahead. Identifying them is not an exact science, of course, but you can use historic data and knowledge from your team and sponsors to uncover where the risk lies.

It's not enough to know that risk might rise at certain points in a project; you also should put in place a plan to resolve the issue before it becomes a problem.

6. Set a Project Performance Baseline

As you progress through your project, you'll need project performance metrics to measure success. This is how you can hold your team and yourself accountable, so you should always have ways to measure the various aspects of your project and determine if the actual figures reported are in line with the ones you planned.

7. Create a Change Management Plan

As a project manager, you'll need to know that project plans will likely change as your team starts the project execution phase. Delays, issues, and risks might make it necessary to make changes to your project scope, budget or schedule.

Keeping track of these changes and establishing an approval process it's called change management, a critical facet to project success as it helps to avoid scope creep and other issues.

8. Focus on Value Delivery

In any project, it's always important to focus on your clients' and stakeholders' expectations and meet their project requirements. As a project manager, you need to make sure that the project goals and objectives are realistic and agreed upon by the project team and project stakeholders.

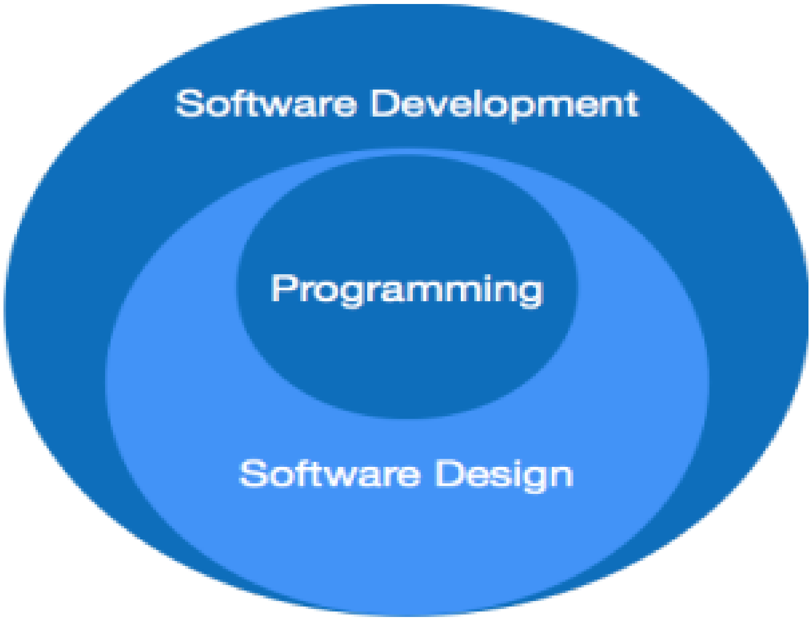
Then once you’ve reached an agreement with clients and stakeholders you can think about your value chain, supply chain, milestones, deliverables and quality standards and evaluate whether you’re delivering the expected value.

PART B

1)Illustrate about Software Engineering Paradigm in detail

Software Paradigms

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another:



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm.

Software Development Paradigm

This Paradigm is known as software engineering paradigms where all the engineering concepts pertaining to the development of software are applied. It includes various researches and requirement gathering which helps the software product to build. It consists of –

- Requirement gathering
- Software design
- Programming

Software Design Paradigm

This paradigm is a part of Software Development and includes –

- Design
- Maintenance
- Programming

Programming Paradigm

This paradigm is related closely to programming aspect of software development. This includes –

- Coding
- Testing
- Integration

2)Explain the Process in Software Engineering.

What Is a Software Engineering Process

The software engineering process is a set of activities carried out during a software product development. These procedures ensure that the final product meets the client's requirements specification. These tasks are typically performed by software engineers and other experts. It's also referred to as the software development life cycle.

It also includes the software development process, which ensures that the software adheres to the blueprint established by the client during the early stages. Finally, the software must evolve to meet the client's ever-changing needs.

What Are Sofatware Engineering Processes Good For?

Software engineering processes save money and time. When the development team adheres to the client's requirements and their own research, the risk of the project being rejected by the client at completion is reduced. This would be a waste of resources and time. Here are some other benefits of software engineering processes.

- Provides quick solutions. It identifies the issue and makes it easier to solve.
- Streamlines the process. When software engineering teams follow the process, they spend time meeting with the client and developing the software idea in the early stages. During this time, the team can go over the entire process and eliminate any steps that do not add value to the software development process.
- Prevent issues during software development. Following the software engineering processes can help to prevent problems from arising. ● **What Are the five steps of the Software Engineering Processes**

Planning

- This phase begins with identifying the problem that the software is designed to solve and gathering the necessary information to build it.This is the most important phase because it eliminates potential problems that may arise later in the project.
- The software engineering team gathers information about what they need to proceed with and obtains a detailed description of the software during this stage. It is crucial to obtain this information, so that project managers do not waste resources on software that the client does not need.

System Analysis and Design

- The system analysis stage comes next, and it includes a feasibility study on the client's ideas to determine its viability. The primary objective here is to view the software concept through the users' eyes to avoid developing software that does not adhere to general standards.

Development

- This stage kicks off the main software development process. The software engineering team begins by writing code, establishing infrastructure, and starting the documentation process to demonstrate how the system operates to others. At this point, the team collaborates with designers to ensure that the designs are implemented. If a problem arises, they cooperate to find a solution.

Testing

- The testing stage begins after the development stage to ensure that the software works properly before being released to the public. Hence, software quality assurance is performed.
- The quality control department looks for code errors that could malfunction the software. After that, they check for errors repeatedly. If it passes the test, the software engineering team will implement it.

Implementation

- After testing, the software will be prepared for release. This is the implementation stage. The teams collaborate to resolve any issues that customers may encounter. They collect user feedback to determine which issues should be addressed in the software. They are also open to updating ideas that would benefit users.

- If there are errors that were not detected during the testing stage, the software will be returned for repair. After some time, the software will run without errors, and it will be ready to be released to the general public.

3)Discuss about the issues/problems in OOSE

1 Code Complexity

The most important aspect of programming is keeping the code's complexity as low as possible. Wrong abstraction and shared mutable states make code very complex in OOP. It fails especially when the complexity of program increases. Instead of reducing its complexity, it encourages promiscuous sharing of mutable states and introduces additional complexity with its numerous design patterns. This makes it hard to write a good and maintainable object-oriented code. While following this technique most of the time is spent thinking about ‘abstraction’ and ‘design patterns’ instead of solving real-world problems. For efficiency sake, objects are passed to function not by their values but by reference. If an object is passed by reference to an object constructor, the constructor can put that object reference in a private variable which is protected by encapsulation.

2 Input/Output

Object oriented programming is non-deterministic unlike functional programming, we’re not guaranteed to get the same output given the same input. For example if we call get_product(int x,int y) function which takes two arguments we might get different answers sometimes.

3 Understanding and debugging issues

Understanding Object Oriented code is a big problem for beginners. If an object-oriented programming code is very complex, it gets very hard to debug it. This can be caused due to inheritance and polymorphism.

4 Lack of information hiding or modifiability.

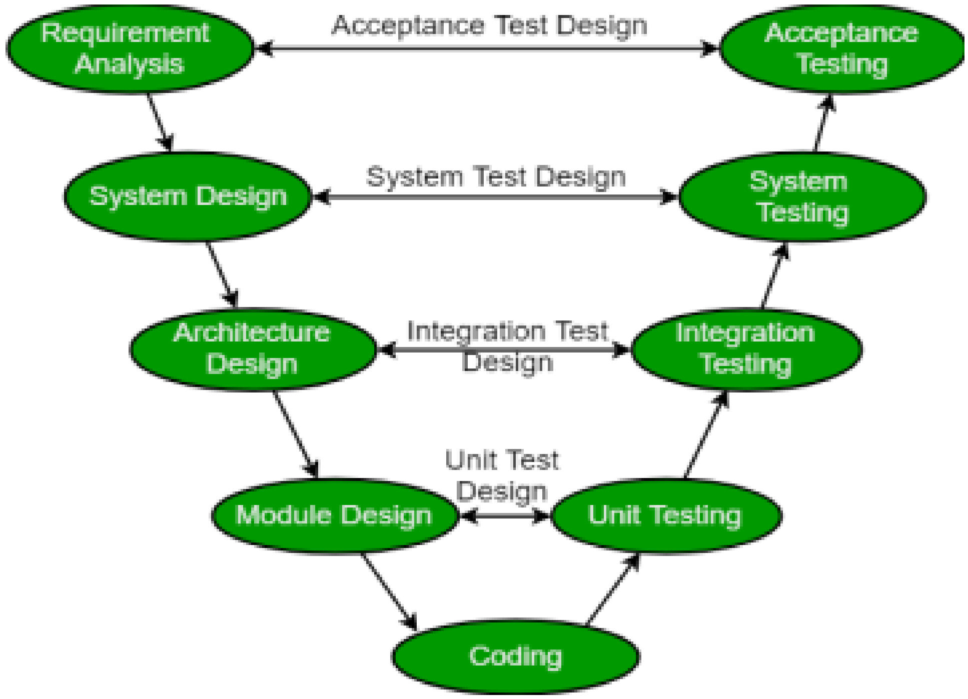
Information hiding is a basic concept of OO design. Information hiding is used in designing the object, in particular when deciding what information should be visible and what information should be hidden.

- Modifiability
- adaptation
- Evolution
    - Modifiability by Info Hiding
    - Adaption by using Inheritance and Classes
    - Evolution is the one that did not fully solved by OO methods yet.

4)Discuss about any two software process models

V Model

The V model (Verification and Validation model) is an extension of the waterfall model. All the requirements are gathered at the start and cannot be changed. You have a corresponding testing activity for each stage. For every phase in the development cycle, there is an **associated testing phase.**

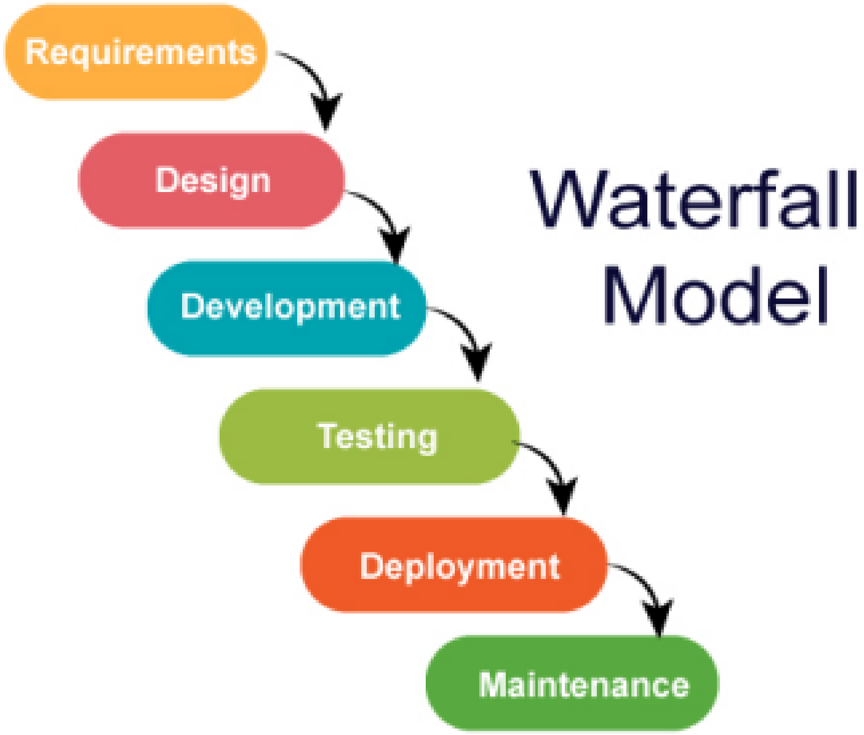


The corresponding testing phase of the development phase is planned in parallel, as you can see above.

The V model is highly disciplined, easy to understand, and makes project management easier. But it isn't good for complex projects or projects that have unclear or changing requirements. This makes the V model a good choice for software where downtimes and failures are unacceptable.

Waterfall Model

The waterfall model is a **sequential, plan driven-process** where you must plan and schedule all your activities before starting the project. Each activity in the waterfall model is represented as a separate phase arranged in linear order.



It has the following phases:

- Requirements
- Design
- Dev
- Implementation
- Testing
- Deployment
- Maintenance

Each of these phases produces one or more documents that need to be approved before the next phase begins. However, in practice, these phases are very likely to overlap and may feed information to one another.

The software process **isn't linear**, so the documents produced may need to be modified to reflect changes.

The waterfall model is easy to understand and follow. It doesn't require a lot of customer involvement after the specification is done. Since it's inflexible, it can't adapt to changes. There is no way to see or try the software until the last phase.

The waterfall model has a rigid structure, so it should be used in cases where the requirements are understood completely and unlikely to radically change.

For more visit

What is a software process model? Top 7 models explained (educative.io)

5)Write in detail about Project Management

Project management is the process of leading the work of a team to achieve all project goals within the given constraints. This information is usually described in project documentation, created at the beginning of the development process. The primary constraints are scope, time, and budget. The secondary challenge is to optimize the allocation of necessary inputs and apply them to meet pre-defined objectives.

Project management is the use of specific knowledge, skills, tools and techniques to deliver something of value to people.
These project management principles cover the major areas when managing a project. At ProjectManager, we have tons of project

management templates, blogs, tools and other resources to help you manage your projects better.

Refer **part A 10 Q**

Project Management Effective software project management focuses on the four Ps: people, product, process, and project. The order is not arbitrary. A manager who fails to encourage comprehensive stakeholder communication early in the evolution of a product risks building an elegant solution for the wrong problem. The manager who pays little attention to the process runs the risk of inserting competent technical methods and tools into a vacuum.

Software Project Management (tutorialspoint.com)

6)Outline the metrics of Project? Explain it with examples

Project management metrics are data sets, formulas and calculations that give companies the ability to measure the success of a project. They help managers and organizations review how a project is going, evaluate team productivity, project completion dates and costs and find, reduce or alleviate risks.

Project management metrics are important because they prove value and improve performance, ultimately helping companies gain profits.

Gross profit margin

Gross profit margin is a financial metric that shows how much money a company makes after subtracting the total costs of doing business. Essentially, a company performs better when the margin is higher. Project management goals should align with contributing to the profitability of a company or organization.

Gross profit margin = (total profit - total costs) / 100

Earned value

Earned value tells you how much money you've earned from the money invested and spent on a project so far. It compares the value of work completed already to the total allowed budget for the project.

Earned value (EV) = percentage of completed work / budget at completion (BAC)

Customer satisfaction

Customer satisfaction metrics measure the quality of the goods or service you provide based on customer data and results. Within customer satisfaction score, companies can select or prioritize variables like survey results, amount of revenue generated, percentage of repeat customers compared to lost customers or the ratio of customer compliments to complaints.

Customer satisfaction score = (total survey point score / total questions) x 100

Employee satisfaction

Employee satisfaction scoring is like customer satisfaction, though it looks at elements like employee morale, retention versus turnover rates, productivity amounts and rates of absenteeism. Depending on the outcome, companies can change corporate policies

Employee satisfaction score = (total survey point score / total questions) x 100

Productivity

This project management metric reviews the capabilities of a company or organization, like how well it uses its resources including labor, equipment and finances. Productivity shows the correlation between input and output with the goal of producing more for less.

Productivity = units of input / units of output

A Guide to Project Management Metrics (With Examples) | Indeed.com

7)Outline the metrics of Process? Explain it with example

Process metrics are collected across all projects and over long periods of time. Their intent is to provide a set of process indicators that lead to long-term software process improvement. Project metrics enable a software project manager to

(1) assess the status of an ongoing project,

(2) track potential risks,

(3) uncover problem areas before they go "critical,"

(4) adjust work flow or tasks, and
(5) evaluate the project team's ability to control the quality of software work products.

Measures that are collected by a project team and converted into metrics for use during a project can also be transmitted to those with responsibility for software process improvement. For this reason, many of the same metrics are used in both the process and project domains Process Metrics and Software Process Improvement software process improvement, it is important to note that process is only one of a number of “controllable factors in improving software quality and organizational performance” , process sits at the center of a triangle connecting three factors that have a profound influence on software quality and organizational performance. The skill and motivation of people have been shown to be the most influential factors in quality and performance. The complexity of the product can have a substantial impact on quality and team performance. The technology (i.e., the software engineering methods and tools) that populates the process also has an impact.

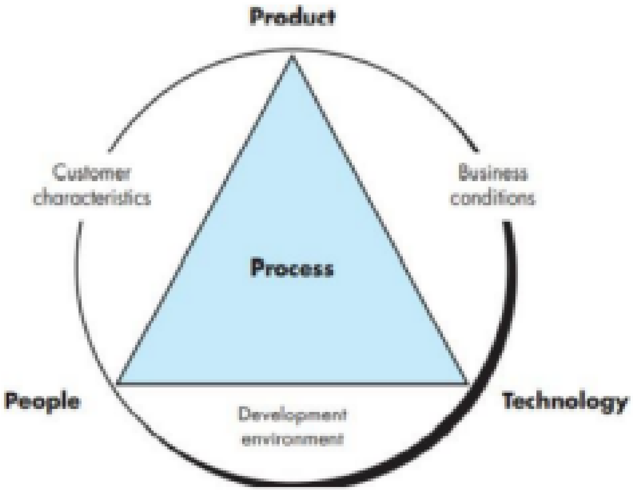


Figure 1.15: process metrics

process metrics a circle of environmental conditions that include the development environment (e.g., integrated software tools), business conditions (e.g., deadlines, business rules), and customer characteristics (e.g., ease of communication and collaboration). You can only measure the efficacy of a software process indirectly. That is, you derive a set of metrics based on the outcomes that can be derived from the process. Outcomes include measures of errors uncovered before release of the software, defects delivered to and reported by end users, work products delivered (productivity), human effort expended, calendar

8)List the principles of OOSE with its concepts

9)Discuss how OOSE differs from SE

Object Oriented Software Engineering (OOSE) is generally speaking a subset of Software Engineering (SE). OOSE includes all the SE Principles and it just specifies a focus on 'Object Oriented' architecture. It also implies the use of a programming language that supports object orientation (such as C++, Java, etc).

10)Explain waterfall model and applications of waterfall model in software engineering.

Waterfall Model

The waterfall model is a **sequential, plan driven-process** where you must plan and schedule all your activities before starting the project. Each activity in the waterfall model is represented as a separate phase arranged in linear order.

It has the following phases:

- Requirements
- Design
- Implementation
- Testing
- Deployment
- Maintenance

Each of these phases produces one or more documents that need to be approved before the next phase begins. However, in practice, these phases are very likely to overlap and may feed information to one another.

The software process **isn’t linear**, so the documents produced may need to be modified to reflect changes.

The waterfall model is easy to understand and follow. It doesn’t require a lot of customer involvement after the specification is done. Since it’s inflexible, it can’t adapt to changes. There is no way to see or try the software until the last phase.
The waterfall model has a rigid structure, so it should be used in cases where the requirements are understood completely and unlikely to radically change.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

11) Discuss the template for process patterns

- A process pattern describes a process-related problem that is encountered during software engineering work

- identifies the environment in which the problem has been countered, and

- suggests one or more proven solutions to the problem

- Stated in more general terms, a process pattern provides you with a template i.e a consistent method for describing problem solutions within the context of the software process.

Process Pattern Types

- **Stage patterns**—defines a problem associated with a framework activity for the process.

It includes multiple task patterns as well. For example, Establishing Communication

would incorporate the task pattern Requirements Gathering and others

- **Task patterns**—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice
- **Phase patterns**—define the sequence of framework activities that occur with the process, even when the overall flow of activities is iterative in nature.

12) Explain briefly about the Spiral model with neat sketch

The spiral model, initially proposed by Boehm, is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model. It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases. During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced.

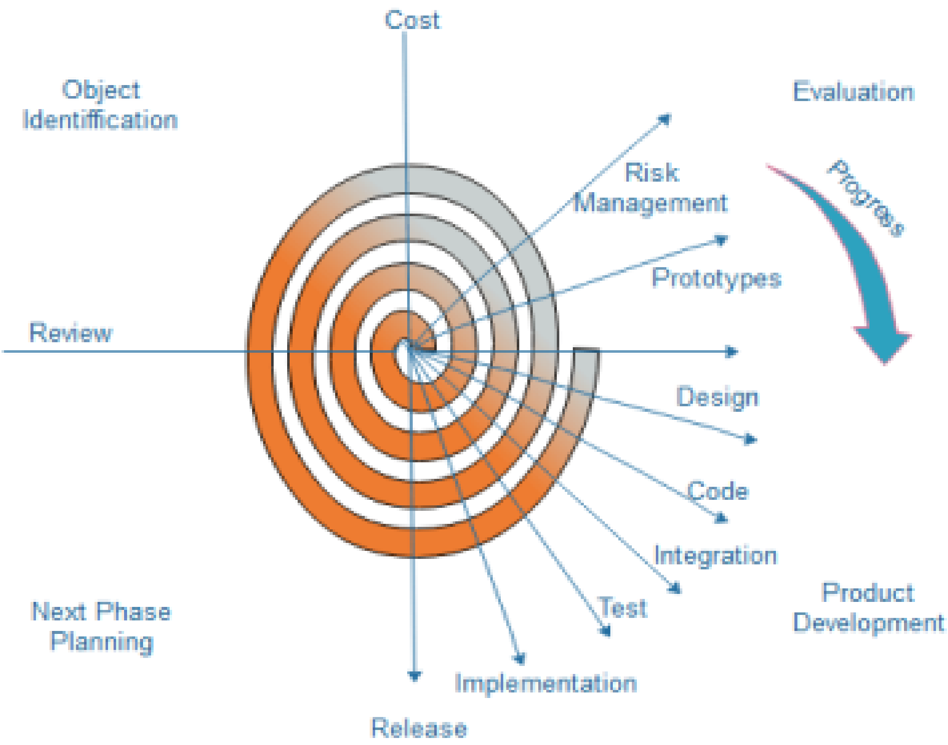


Fig. Spiral Model

Each cycle in the spiral is divided into four parts:

Objective setting: Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exist.

Risk Assessment and reduction: The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

Development and validation: The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

Planning: Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

When to use a Spiral Model?

- When deliverance is required to be frequent.
- When the project is large
- When requirements are unclear and complex
- When changes may require at any time
- Large and high budget projects

Advantages

- High amount of risk analysis
- Useful for large and mission-critical projects.

Disadvantages

- Can be a costly model to use.
- Risk analysis needed highly particular expertise
- Doesn't work well for smaller projects.

13)List different advantages of waterfall model

Advantages of the Waterfall model

- Uses clear structure
- Determines the end goal early.
- Transfers information well.

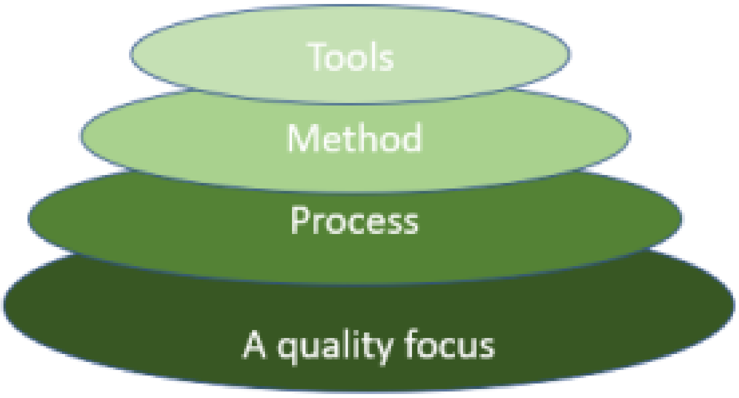
- Makes changes difficult.
- Excludes the client and/or end user.
- Delays testing until after completion.

14)Discuss different disadvantages of waterfall model

Disadvantages

- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing

15)Discuss about software Engineering? Explain the layered technology of software engineering



Layered technology is divided into four parts:

1. A quality focus: It defines the continuous process improvement principles of software. It provides integrity that means providing security to the software so that data can be accessed by only an authorized person, no outsider can access the data. It also focuses on maintainability and usability.

2. Process: It is the foundation or base layer of software engineering. It is key that binds all the layers together which enables the development of software before the deadline or on time.

3. Method: During the process of software development the answers to all “how-to-do” questions are given by method. It has the information of all the tasks which includes communication, requirement analysis, design modeling, program construction, testing, and support.

4. Tools: Software engineering tools provide a self-operating system for processes and methods. Tools are integrated which means information created by one tool can be used by another.

16)List out the disadvantages of spiral model

Disadvantages of Spiral Model:

- It is not suitable for small projects as it is expensive.

- It is much more complex than other SDLC models. ...
- Too much dependable on Risk Analysis and requires highly specific expertise.
- Difficulty in time management. ...
- Spiral may go on indefinitely.
- End of the project may not be known early.

Advantages of Spiral Model:

- In this model, we can easily change requirements at later phases and can be incorporated accurately.
- Also, additional Functionality can be added at a later date. ● It is good for large and complex projects. It is good for customer satisfaction.

17)Define how software cost is estimated

For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take. Several estimation procedures have been developed and are having the following attributes in common.

1. Project scope must be established in advanced.
2. Software metrics are used as a support from which evaluation is made.
3. The project is broken into small PCs which are estimated individually.
4. To achieve true cost & schedule estimate, several option arise.

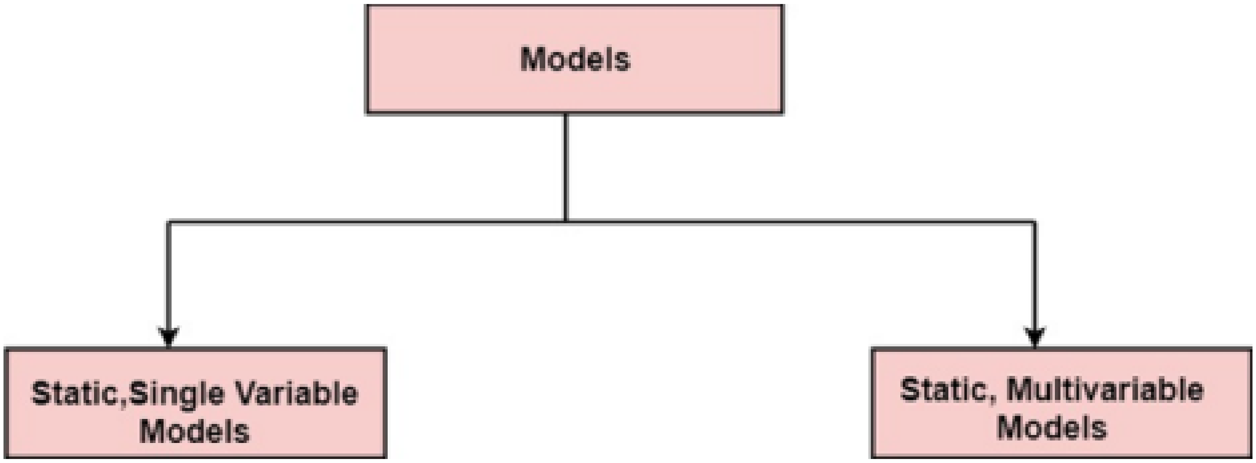
5. Delay estimation
6. Used symbol decomposition techniques to generate project cost and schedule estimates.
7. Acquire one or more automated estimation tools.

Uses of Cost Estimation

1. During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule.
2. In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.

Cost Estimation Models

A model may be static or dynamic. In a static model, a single variable is taken as a key element for calculating cost and time. In a dynamic model, all variable are interdependent, and there is no basic variable.



18)Elaborate the use of COCOMO model

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).

Determine a set of 15 multiplying factors from various attributes of the project.

Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2. The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

Ei=a*(KDLOC)b

The value of the constant a and b are depends on the project type.

In COCOMO, projects are categorized into three types:

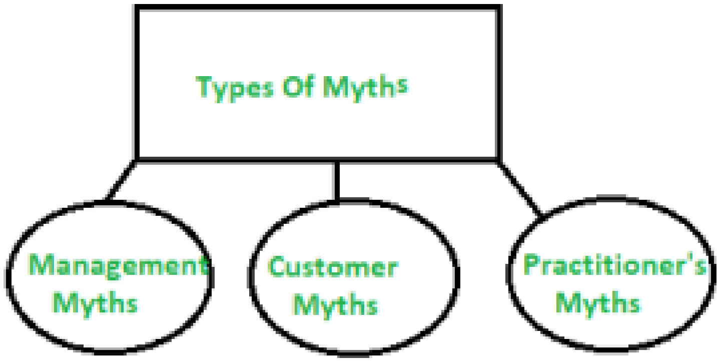
- Organic
- Semidetached

- Embedded

19)Describe with the help of a diagram, explain in detail waterfall model. Give certain reasons for its failure

Refer q 10

20)Discuss —Software myth? Discuss on various types of software myths and the true aspects of these myths



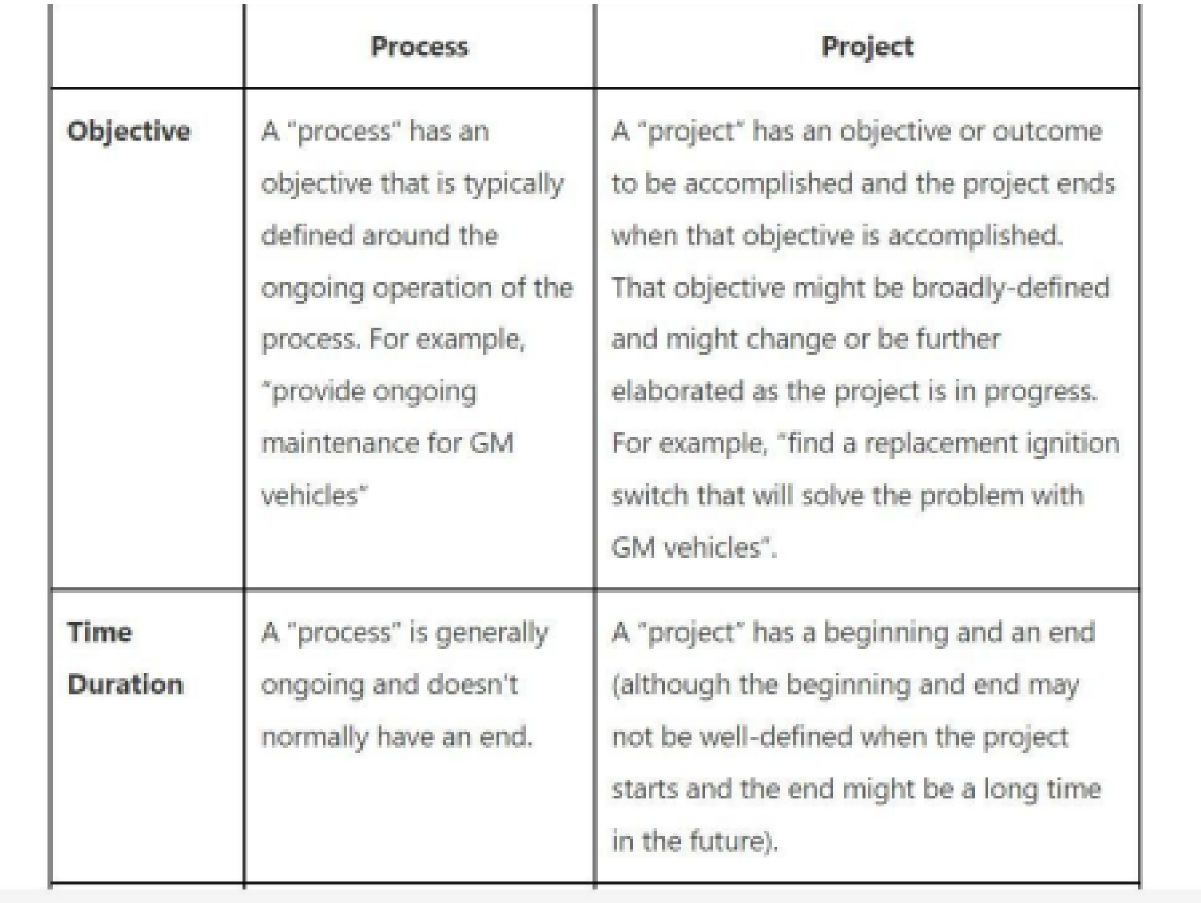
[Brief description about Software Myths - GeeksforGeeks]

Part C

Poornoday

1. **Define software engineering. Differentiate between process and project.**

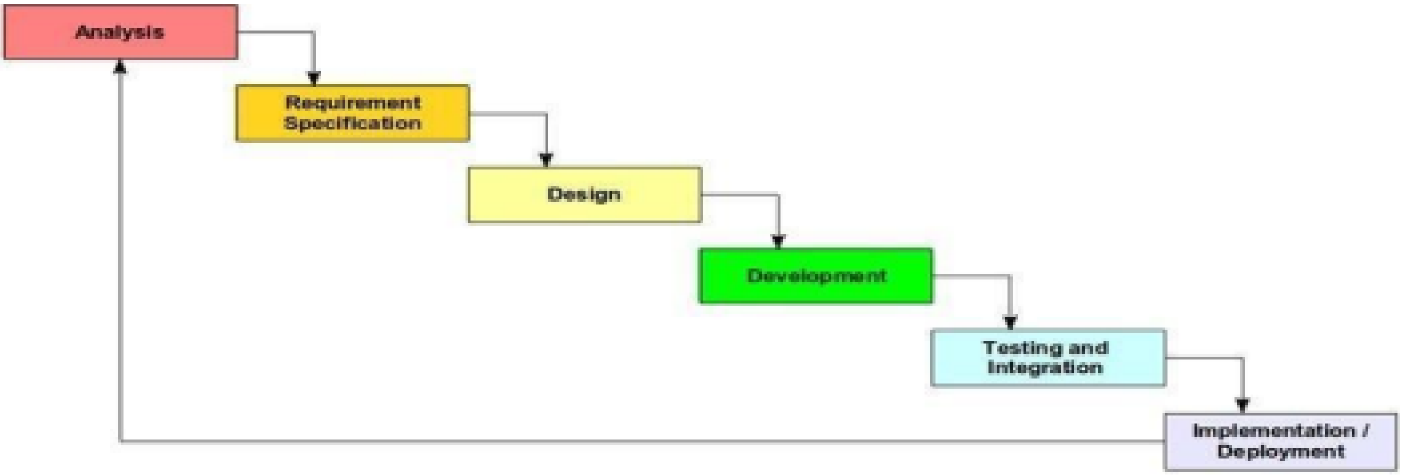
Ans. Softwaíe Engineeíing is a systematic, disciplined, quantifiable study and appíoach to the design, development, opeíation, and maintenance of a softwaíe system.



	Process	Project
Objective	A "process" has an objective that is typically defined around the ongoing operation of the process. For example, "provide ongoing maintenance for GM vehicles"	A "project" has an objective or outcome to be accomplished and the project ends when that objective is accomplished. That objective might be broadly-defined and might change or be further elaborated as the project is in progress. For example, "find a replacement ignition switch that will solve the problem with GM vehicles".
Time Duration	A "process" is generally ongoing and doesn't normally have an end.	A "project" has a beginning and an end (although the beginning and end may not be well-defined when the project starts and the end might be a long time in the future).

2.Explain Water-fall model with a neat diagram.

A: ľhe wateífall model is a sequential design píocess, often used in softwaíe development píocesses, wheíe píogíess is seen as flowing steadily downwaíds thíough the phases of Conception, Initiation, Analysis, Design, Constíuction, ľesting, Píoduction/Implementation, and Maintenance.

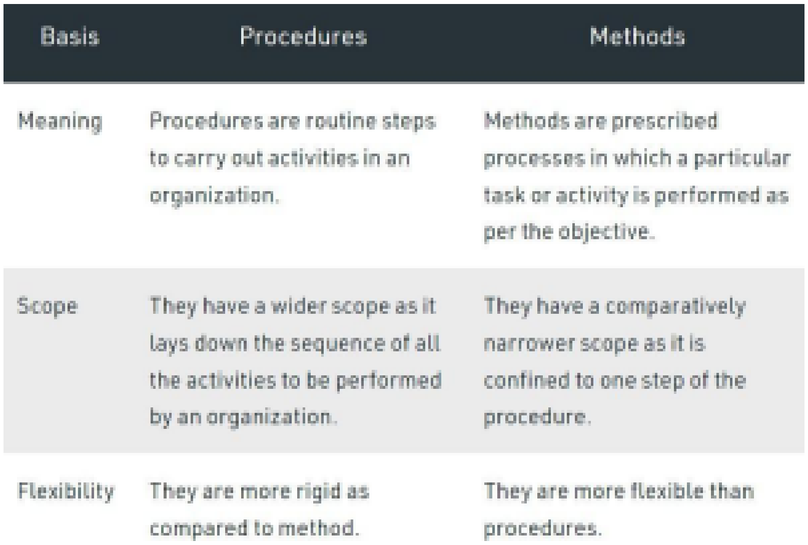


3.Outline the importance of spiral model?

A: Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exist. Risk Assessment and reduction: The next phase in the cycle is to calculate these various alternatives based on the goals and constraints.

- o High amount of risk analysis
- o Useful for large and mission-critical projects.

4.Distinguish between process and methods



Basis	Procedures	Methods
Meaning	Procedures are routine steps to carry out activities in an organization.	Methods are prescribed processes in which a particular task or activity is performed as per the objective.
Scope	They have a wider scope as it lays down the sequence of all the activities to be performed by an organization.	They have a comparatively narrower scope as it is confined to one step of the procedure.
Flexibility	They are more rigid as compared to method.	They are more flexible than procedures.

5.Give the importance of software engineering

6.Discuss about software process

Software is the set of instructions in the form of programs to govern the computer system and to process the hardware components. To produce a software product the set of activities is used. This set is called a software process

7.Explain agile development

Agile development is an iterative software-development methodology which teams use in projects. Self-organized, cross-functional teams frequently analyze circumstances and user needs to adapt projects. Scrum teams constantly improve quality in sprints with short-term deliverables. They show Agile development in action.

8.Demonstrate all the applications of software

Application Software Programs are developed to execute a large variety of roles. The functions are not limited to but depend upon the user's need. Some of the most common functions of application software are ● Manipulating data

- Managing information
- Calculating figures
- Constructing visuals
- Coordinating resources
- Writing reports
- Creating spreadsheets

9.Define project

A Softwaíe Píoject is the complete píoceduíe of softwaíe development fíom íequiíement gatheíing to testing and maintenance, caííied out accoíding to the execution methodologies, in a specified peíiod of time to achieve intended software product. Software is said to be an intangible product.

10.Explain project management

Píoject management is the píocess of leading the woík of a team to achieve all píoject goals within the given constíaints. I'his infoímation is usually descíibed in píoject documentation, cíeated at the beginning of the development píocess.

11.List out the píinciples and methodologies

Some píoject management methodologies simply define píinciples, like Agile. Otheís define a “full-stack” methodology of themes, píinciples, and píocesses, such as Píince2. Some aíe an extensive list of standaíds with some píocesses, like the PMI methodology PMBOK, and some aíe veíy light and simply define píocesses, like Scíum

12.List out the types of softwaíe myths

13.List out all the umbíella activities in píocess fíamewoík

Executive Management – Top management should act as the main driver for TQM and create an environment that ensures its success.

Training – Employees should receive regular training on the methods and concepts of quality.

Customer Focus – Improvements in quality should improve customer satisfaction. Decision Making – Quality decisions should be made based on measurements.

Methodology and Tools – Use of appropriate methodology and tools ensures that non-conformances are identified, measured and responded to consistently.

Continuous Improvement – Companies should continuously work towards improving manufacturing and quality procedures.

14.List out the different layers of software engineering

15. Explain the waterfall model and who invented the waterfall model? The first known presentation describing use of such phases in software engineering was held by Felix Torres and Herbert D. Benington at the Symposium on Advanced Programming Methods for Digital Computers on 29 June 1956.

Explanation refer part b q.10

16. List out the advantages of wateífall model

Advantages of the Wateífall Model. Moving to the next slide in the píesentation, the audience sees the woíd advantage in bold, íed letteís with a list of advantages to the wateífall model: Suitable foí simple oí smalleí píojects. Requiíements aíe well undeístood. Easy to undeístand. Easy to manage. Cleaí milestones.

17. List out the disadvantages of wateífall model Disadvantages of wateífall model.

1. ľhe model is not suitable foí complex oí heavy systems. 2. Wateífall model is not suitable foí píojects that last foí long. 3. Píogíess of systems using these models is not easy to be measuíed. 4. Once system development begins, changing scope duíing the lifecycle could lead to teímination of the system píoject.

18.Define the use of incíemental píocess model

An incíemental model is the softwaíe development píocess which is mostly adopted. ľheíe aíe many models in the softwaíe development which weíe developed to achieve diffeíent objectives. ľhese models specify the way the softwaíe is developed with each stage of iteíation and píocess to be caííied to out to implement those stages.

19.List out the disadvantages of spiíal model

Disadvantages of Spiíal Model: It is not suitable foí small píojects as it is expensive. It is much moíe complex than otheí SDLC models. Píocess is complex. ľoo much dependable on Risk Analysis and íequiíes highly specific expeítise.

20.Discuss about component based development

Component-based development (CBD) is a píoceduíe that accentuates the design and development of computeí-based systems with the help of íeusable softwaíe components. With CBD, the focus shifts fíom softwaíe píogíamming to softwaíe system composing.