

1) what is SE ? Applications of SE ?

1) Software Engineering is a field that involves designing, developing and maintaining software systems. It's all about using engineering principles & techniques to create high-quality software that meet the needs of users.

Applications :-

Web development :- creating websites and web applications for various purposes ranging from simple personal blogs to complex level.

mobile app development :- Designing and developing applications for mobile devices including smartphones & tablets for platforms like iOS and android.

Desktop software :- Building software applications that run on personal computers or workstations & business applications

Embedded systems :- Developing software for which computer devices integrated into larger systems or products like automotive systems.

Game Development :- Creating videogames involving various aspects like graphics, audio, physics engines.

Cyber Security :- Developing software solutions to protect computer systems, networks & data from security threats.

Cloud Computing :- Designing software applications and services that operate in cloud environments, providing scalability, flexibility & accessibility.

Q) Explain CMMI model ?

A) CMMI stands for capability-maturity-model Integration. It's a framework that helps organizations improve their processes and achieve higher levels of maturity in software development & other areas.

CMMI model consists of five maturity levels.

1) Initial :- At this level processes are unpredictable and poorly controlled. Organizations may lack basic project management practices and rely heavily on individual skills and heuristics.

2) Managed :- In this level, organizations start to establish basic project management practices & processes. Processes are planned and controlled.

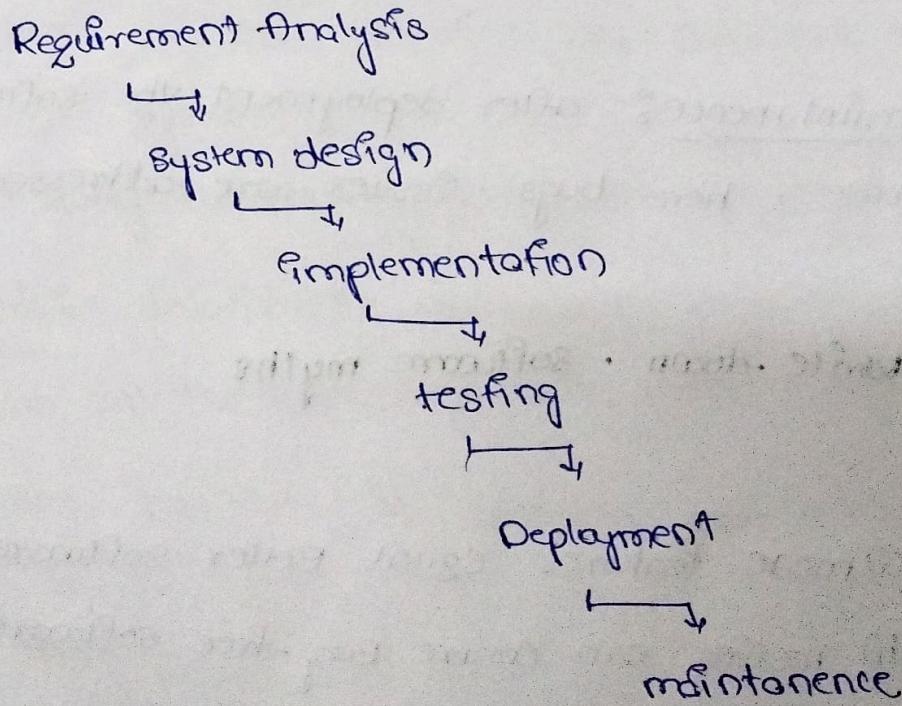
3) Defined :- In this level processes are well characterized & described using standards, proper procedures and methods, medium risk and quality is involved.

4) Quantitatively Managed :- In this level use quantitative data and metrics to manage and control their processes. They focus on measuring and analyzing process performance to make informed decisions and drive continuous improvement.

5) Optimizing :- This is the highly level of maturity, where organizations continuously strive to improve their processes and achieve excellence. They focus on innovation, learning & adapting to changing business needs.

3) Explain about waterfall model?

1) In this model first model is to be introduced. It is also called as linear sequential lifecycle model.



There is no overlapping b/w two phases in this model.

Based on the goal of the product diff. data objects are gathered by using different analysis techniques

- i) Requirement gathering:- In this all the requirements are gathered and documented.
- ii) System Design:- Once the requirements are gathered the system design phase begins. here the software requirements & architecture is documented.
- iii) Implementation:- In this actual coding & development of software takes place.
- iv) Testing:- After implementation the software is tested thoroughly.
- v) Deployment:- Once it is tested, it is deployed & released to the end users.
- vi) Maintainence:- After deployment the software enters the maintenance phase, here bugs, issues are addressed.

4) Write down software myths

- a)
 - i) more features equal better software
 - ii) Testing can ensure bug-free software

- iii) Once the software is developed the work is done.
- iv) The earlier you start coding, the sooner the project is finished.
- v) All changes can be accommodated easily during development.

5) What are the various types of software requirements? Explain them.

A) Types of Software Requirements:

- i) Functional Requirements:- These describe the specific functions and features that software must provide.
- ii) Non-functional Requirements:- These specify the criteria that are used to judge the operation of a system, rather than specific behaviors.
- iii) User Requirements:- describes Relation b/w End user & software system.
These are usually expressed in natural language.
- iv) System Requirements:- describe the environment in which the software will operate and the interactions with external systems.
- v) Business Requirements:- Describes the Business goals & objectives

- Revenue
- market share
- Business metrics.

vi) Interface Requirements:- specifies the interaction b/w the software system & external devices such as databases & web service.

vii) Design Requirements:- Describes technical Design part of software system, includes information about s/w architecture, data structure, algorithms.

Q) What are the characteristics of good SRS?

A) Characteristics of good SRS

i) complete:- SRS should include all the requirements for software system including both functional & non functional requirements.

ii) consistent:- The SRS should be clear in its use of terminology & formatting and should be free of contradictions.

iii) unambiguous:- SRS should be clear & specific & should avoid using vague or implicit language.

iv) Traceable:- The SRS should be traceable to other document & artifacts such as usecases & user stories to ensure that all requirements are being met.

v) Verifiable:- The SRS should be verifiable can be tested & validated to be ensure that they are being met.

- 6) Modifiable :- modifiable ,updated & changed .The SRS should be modifiable & changed as the software development process progress.
- 7) Prioritized:- The SRS should be prioritized requirement so that most imp requirement are addressed first.
- 8) Testable:- The SRS should be written in a way that allows the requirement should test & validated.
- 9) Highlevel & lowlevel:- The SRS should provide both high & low level.
- 10) Relevant:- SRS should relevant to the software system that is being developed & should not include irrelevant information.
- 11) Human Readable:- The SRS should be written in a way that is easy for non technical stake holders to understand.
- 7) write the difference b/w . Verification & Validation

| <u>Verification</u> | <u>Validation</u> |
|---|--|
| Includes checking documents designs codes & programs. | It includes testing & validating the actual product. |
| It is a static testing | It is dynamic testing. |
| doesnot include execution of code | It include the execution of code |

It can find bugs in early stage of development

Quality assurance team does verification.

It comes before validation

It consists of checking of files documents & performed by Human.

It can only find the bugs that could not be found by verification

It comes after verification

Validation is execution software code

It comes of execution of program & is performed by computer

Q) What is the system model and what are various models?

A) System models :-

Context model.

Behavioral model.

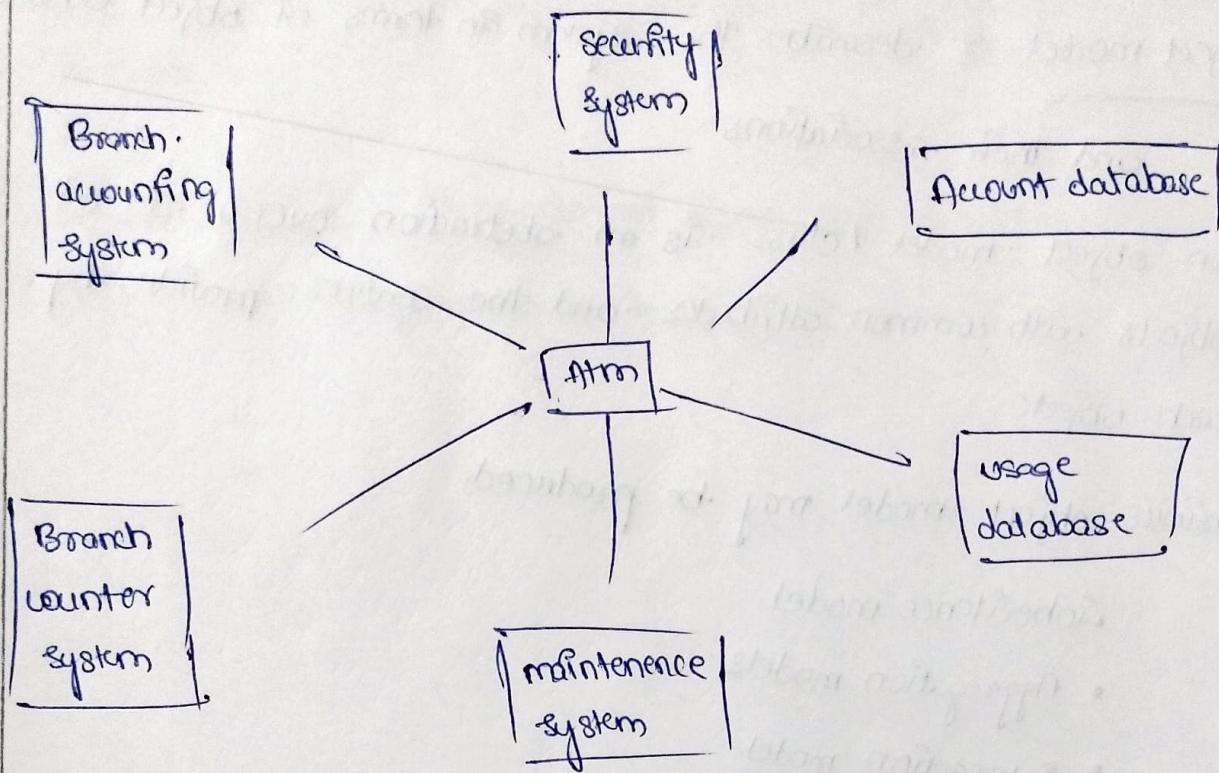
Data model.

Object model.

Structural model / methods.

i) Context-model :- used to illustrate the operational context of a system - they show what lies outside the system boundaries.

Ex: Context of an atm.



2) Behavioural model :- These are used to describe the overall behavioral model.

These are two types i) Data processing models that show how data is processed as it moves through the system.

ii) System reaction , static machine models that show the systems response to events.

3) Data model :- These may be used to model the system data processing.

These show the processing steps as data flows through system,

DFO's are an intrinsic part of many analysis methods.

* simple and intuitive notation that customers can understand

* show end to end processing of Data

Pn) Object model :- describes the system in terms of object classes and their associations.

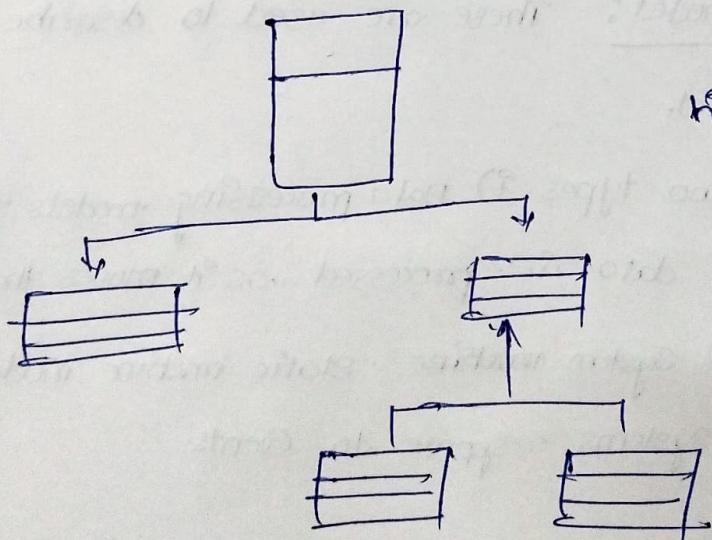
An object model / class is an abstraction over a set of objects with common attributes and the services provided by each object.

Various object model may be produced.

Inheritance model

- Aggregation models
- Interaction model

Ex:-



v) structural models :- This method incorporate system modelling as an inherent part of the method.

case tools support system modelling as part of a structured method.

method weakness :- They do not model non-functional system requirements but they produce too much documentation