Date: / / Page no:

Software Engineering

Software -

Software is a set of computer programs and associated documentation and data. This is in contrast to hardware, from which the system is built and which actually perform the work.

Software engineering -

Software engineering is a systematic engineering approach to software development. A software engineer is a person who applies the principles of software engineering to design, develop, maintain, test and evaluate computer software.

Problems and prospects of software development -

1) Integration issues -

It can be challenging to integrate what you are making with tools your audience may already be using. You need to find ways to make your product compatible using application program interfaces (API's) or partnering with these other tech brands on ways to work together.

2) Communication breakdowns -

Software development requires a team collaborating to address customer needs. Every member has to know a project's coding strategy, objective and goals. Otherwise, the fallout affects the manager's reputation and the team's output.

3) Unrealistic or mismanaged timelines -
One of the most common issues in project management is the infamous timeline. Being realistic in setting timelines with projects is critical.

4) Feature overload -
The largest obstacle that comes while launching a new software package is the desire to put too much into one application.

5) Underestimating the task at hand -
Developers know that there can always be bumps on the road to deployment. Save yourself some time and hardship by scheduling in some extra cushion time in case it is needed.

6) Underestimating the importance of quality assurance -
7) Feature creep - The excessive expansion of new features is a common obstacle.
8) Security related release delays - Developers are often focused on getting good, working code (software) out the door.
9) Not defining a target audience -
10) Underestimating the demand -

Date: / / Page no:

Software development life cycle (SDLC) -
A Software life cycle model is a pictorial and diagrammatic representation of the software life cycle. SDLC cycle represents the process of developing software. SDLC framework includes the following steps -

Stage 1 -
Planning and requirement analysis -
Requirement analysis is the most important and necessary stage in SDLC. Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

Stage 2 - Defining requirements -
Once the requirement ~~& management~~ analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

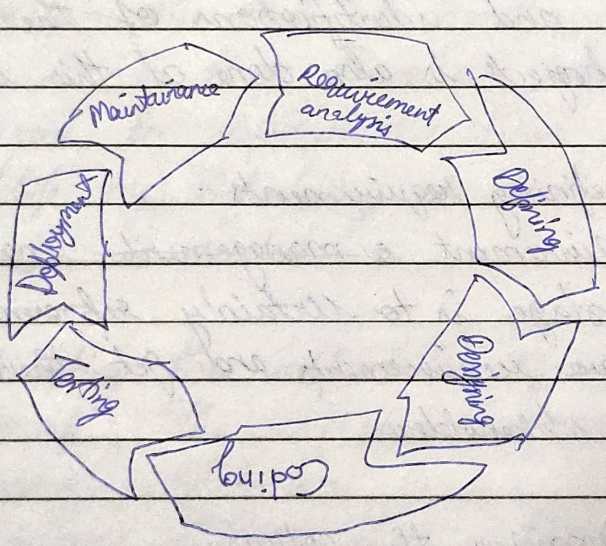
Stage 3 - Designing the software
The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project.

Stage 4 - Developing the project
In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code.

Date: / / Page no:

Stage 6- Deployment -
Once the software is certified, and no bugs or errors are stated, then it is deployed. Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

Stage 7- Maintainance
Once the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.



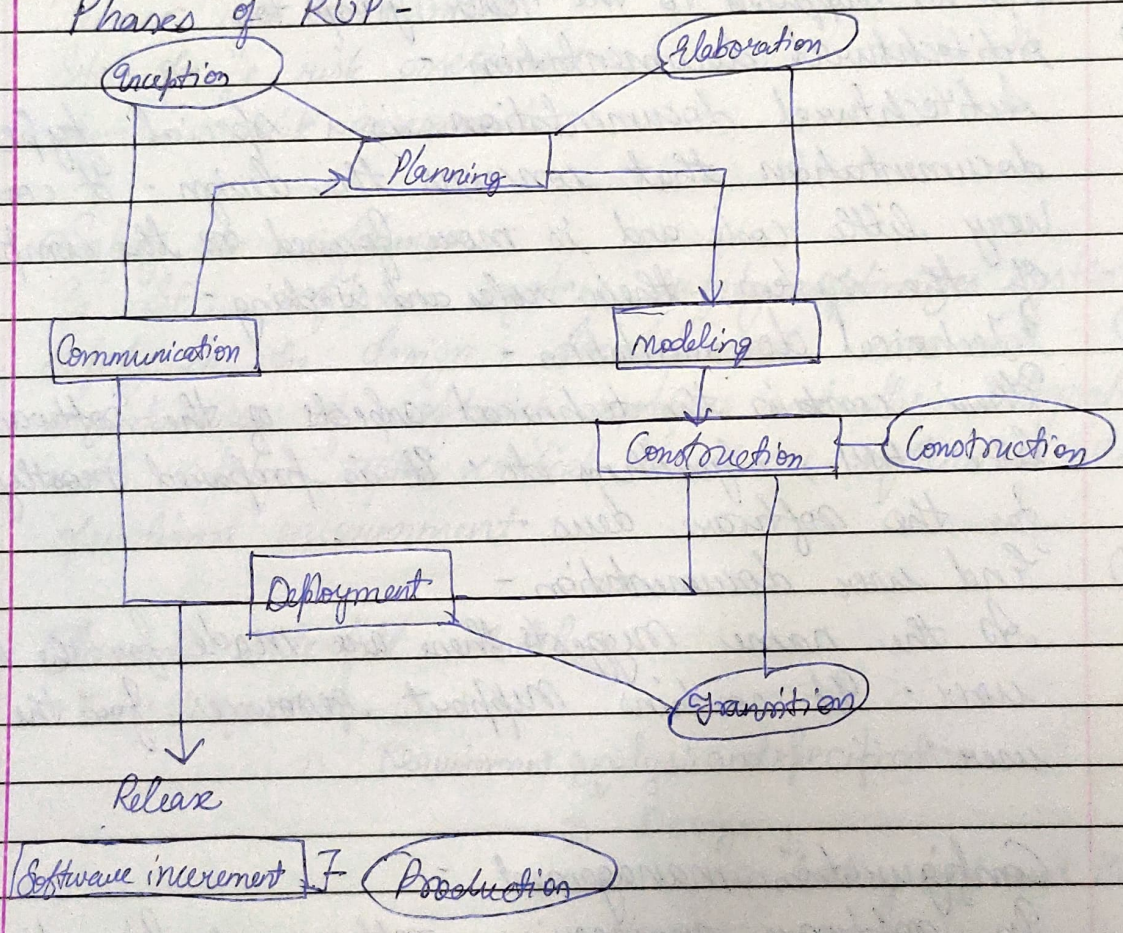
Open source software development -
Open source software is developed in a decentralized and collaborative way, relying on peer review and community production. Open source software is often cheaper, more flexible, and has more longevity than its proprietary peers because it is developed by communities rather than a single author or company.

Date: / / Page no:

The unified process -

Rational unified process (RUP) is a software development process for object oriented models. It is also known as unified process model. It is created by rational corporation and is designed and documented using UML (unified modeling language). This process is included in IBM rational method composer (RMC) product.

Phases of RUP -



Documentation -

Software documentation is a written piece of text that is often accompanied with a software program. This makes the life of all the members associated with the project more easy. It may contain anything from API documentation, built notes or

just help content. It is very critical process in software development. Types of software documentation are -

1) Requirement documentation -
It is the description of how the software shall perform and which environment setup would be appropriate to have the best out of it. These are generated while the software is under development and is supplied to the tester groups too.

2) Architectural documentation -
Architectural documentation is a special type of documentation that concerns the design. It contains very little code and is more focused on the components of the system, their roles and working.

3) Technical documentation -
These contain the technical aspects of the software like API, algorithms etc. It is prepared mostly for the software devs.

4) End user documentation -
As the name suggests these are made for the end user. It contains support resources for the end user.

Configuration management -
In software engineering, software configuration management is the task of tracking and controlling changes in the software, part of the larger cross disciplinary field of configuration management. SCM practices include revision control and the establishment of baselines.

Date: / / Page no:

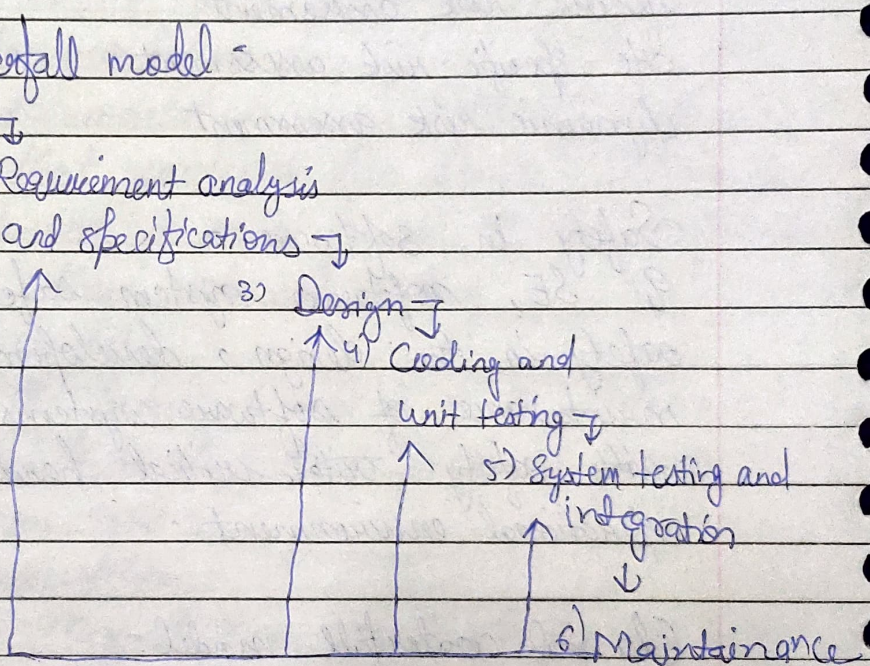
Risk assessment -
Software risk assessment is a process of identifying, analyzing, and prioritizing risks. In general, there are large, medium, and small software projects that each of them can be influenced by a risk.
Types of risk assessment -
Qualitative risk assessment -
Quantitative risk assessment
Generic risk assessment
Site-specific risk assessment
dynamic risk assessment

Safety in software -
In SE, software system safety optimizes system safety in the design, development, use and maintainance of software systems and their integration with safety-critical hardware systems in an operational environment.

Classical waterfall model -
1) Feasibility study
2) Requirement analysis and specification
3) Design
4) Coding and unit testing
5) System testing and integration
6) Maintainance

Advantages	Disadvantages
Base model	No feedback
Simple and easy	No experiment
Suitable for small projects	No parallelism
	High risk
	60% efforts in maintainance

Iterative waterfall model -



Advantages	Disadvantages
Base model	No phase overlapping
Simple and easy	No intermediate delivery
small projects	Rigid (No changes)
Feedbacks	Less customer interaction

Date: / / Page no:

V - Shaped model -

It is also known verification and validation model.
Extension of waterfall model.
Testing is associated with every phase of lifecycle.
Verification phase (Requirement analysis, system design, architecture design, module design).
Validation phase (Unit testing, integration, system, acceptance, testing)

Advantages -

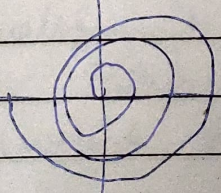
Time saving.
Good understanding of project in the beginning.
Every component must be testable.
Progress can be tracked easily.
Proactive defect tracking.

Disadvantages -

No feedback so less scope of change.
Risk analysis not done
Not good for big or object oriented projects.

Spiral model -

1) objective determination and identify alternative solution
2) Identify and resolve risks
3) Develop next version of product
4) Review and plan for next phase.

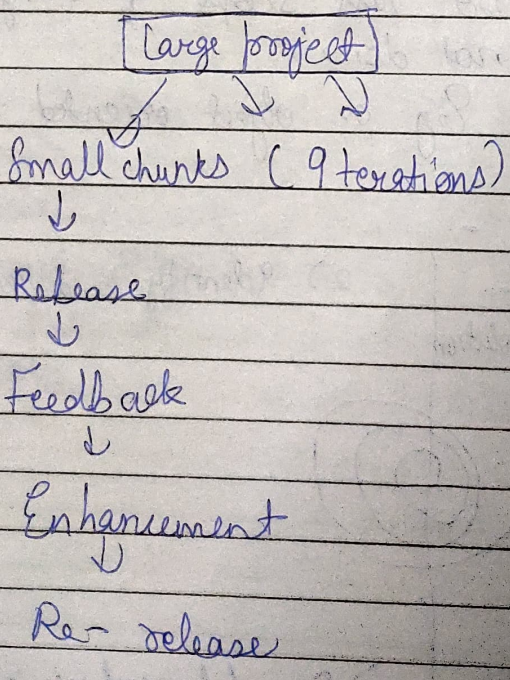


Date: / / Page no:

Features -
Risk handling
Radius of spiral = Cost
Angular dimension = Progress
Meta model (Means it uses multiple model)

Advantages -	Disadvantages -
Risk handling	Complex
Large projects	Expensive
Flexible	Too much risk analysis
Customer specification	Time

Agile model (Latest) -
Agile model follows "move quickly" - It releases software very fast.



Date: / / Page no:

Advantages	Disadvantages
1) Frequent delivery	Less documentation
2) Face to face communication with client	Maintainance problem
3) Changes (flexible)	
4) Time (Less time taken)	

SCRUM model -

1) It comes under agile model.
2) One of the most popular agile methodology.
3) Scrum is leightaright, iterative and incremental framework.
4) Scrum breaks down the development phases into stages or cycles called "sprints".
5) The development time for each sprint is maximised and dedicated, thereby managing only one sprint at a time.
6) Scrum team has scrum master and product owner with constant communication on daily basis.
7) Keywords: Backlog, sprint, daily scrum, scrum master, product owner.

Advantages -

1) Freedom and adaption
2) High-quality, low risk product.
3) Reduce the development time up to 40%.
4) Scrum customer satisfaction is very important
5) Reviewing the current sprint before moving to new one.