

## Unit: 1: Software and Software Engineering

\* Explain Software Engineering with ~~its~~ Characteristics of Software.

=> Software Engineering:

The Software is a collection of Integrated Programs.

Software Engineering is the evolution of Software product using well-defined scientific principles and techniques.

The result of Software engineering is an effective and reliable software product.

Software Engineering is use to manage large software and to manage the dynamic nature of software.

Software Engineering is also use to manage quality and cost.

-> Characteristic of Software:

- 1 Software can be Engineered or Develop not Manufacturer.
- 2 Software can not be seen or touched.
- 3 Software provides desired functionality under the given condition.
- 4 Software's Use of System ~~software~~ resources in the most effective manner.
- 5 Software should be run on different - different platform.
- 6 Software can be Update or change easily.

\* Explain changing nature of Software.

or

\* Explain seven Categories of Software.

or

\* Explain Software Application Domain.

=> This are the seven Categories of Software.

- 1) System Software
- 2) Application Software
- 3) Embedded Software
- 4) Artificial Intelligence Software
- 5) Engineering and Scientific Software
- 6) Product Line Software
- 7) Web Apps Software.

1 System Software : System Software is a type of computer program that is designed to run a computer's hardware.

Ex. Windows, Mac OS etc.

2 Application Software: This software is designed to handle specific tasks of users.

Ex. MS Word, MS PowerPoint etc.

3 Embedded Software: This Software is use to run or control Particular hardware.

Ex. Television sets, Videogames etc.

4 Artificial Intelligence Software:

AI Software is use to make human intelligence process machines.

Ex. Google Assistant

5 Engineering and Scientific Software: This Software is design for make Numerical data calculations.

Ex. CAD, CAM etc.

6 Product line Software: This Software is design for make a collection of similar software system.

Ex. Phone, Laptop etc.

7 Web Apps Software: This software can run in your web browser

Ex. Gmails, Canva etc.

\* Explain or Define: Legacy Software

=> Legacy Software is an older version of a program or application or software.

Legacy Software can not be updated easily or difficult to replace.

Legacy Software can be costly in maintenance.

If we try to update Legacy Software then data of software might be lost.

Legacy Software have compatibility and functionality issues.

Legacy Software provides limited flexibility.

\* Explain Unique nature of Webapp.

=> This are the Common Attributes for Webapp.

1 Network Intensiveness: A Webapp must serve the needs of clients using Internet.

2 Concurrency: A large number of user may access the Webapp at one time.

3 Unpredictable Load: The number of users on Web app may be increase or decrease day by day.

4 Performance: Web app Provides best Performance to the User.

5 Availability: Web app can access any time and any location.

6 Data Driven: Always Web app should contain hypermedia such as text, graphics or audio.

- 7 Content Sensitive : The Content and Data remains same quality ~~a~~ in Webapps.
- 8 Continuous Evolution : Web app after the initial release is change ~~afte~~ according to market requirements.
- 9 Immediacy : Webapps can be release time-to-time in the market.
- 10 Security : Web apps are available via network access sensitive content must be protected
- 11 Aesthetics : Design of Web apps should be look and feel to the User.

\* Explain Layered structure in the Software Engineering.

=> Software Engineering is a Fully Layered technology.

To Develop Software we need to

go from one layer to another layer.

These are the Four Layered in Software Engineering.

- 1) Tools
- 2) Method
- 3) Process
- 4) A Quality Focus.



### 1 Quality Focus :

This layered defines the continuous process improvement principles of Software.

This Layered Protect the Quality of Data in the Software.

This Layered also Protect the data from the outside the user.



## 2 Process :

This Layered is the foundation or base layer of Software engineering.

This Layered covers all the activities, actions and tasks required to be carried out for the Software.

Process defines a framework that must be design in a effective way.

## 3 Methods :

This Layered define the method which is use to develop Software.

This Layered includes requirement analysis, design modeling or supports in Software.

## 4 Tools :

This Layered define the tools which are use to develop Software.

\* Explain Software Process and Process Framework,

=> Software Process Framework:

Software Process Framework is an abstraction of the software development Process.

There are three steps is includes in Software Process:

- 1) Tasks
- 2) Action
- 3) Activities

1 Tasks: This Process is include construction of a Software.

2 Action: This Process is includes Modeling of the software.

3 Activities: This Process is includes communication regarding the Software.

=> Process Framework:

The Process Framework is required for representing common process activities.

There are five processes to describe a Process Framework.

- 1) Communication
- 2) Planning
- 3) Modeling
- 4) Construction
- 5) Deployment

1 Communication: In this process, by communication, we have to gather information of a software.

We have to communicate with customer for software requirements.

2 Planning: After the communication with customer, we have to make a plan for creating a software.

3 Modeling: According to Planning of a Software, We have to Design a Software.

We have to Design Software to understand the problem.

4 Construction: According to Design of a Software, We have to create code for a Software.

5 Deployment: After the Construction of Software, We have to take customer feedback.

According to Customer Feedback, We have to change the Software.

\* Explain Umbrella Activities.

=> There are Eight types of Umbrella Activities.

1 Software Project tracking and Control: This activities we have to track the Software

work on the basis of Planning.

- 2 Risk Management: If We add the risk method or technique in the software then we have to think about the solution.
- 3 Software Quality Assurance: This are activities required to maintain Software Quality.
- 4 Technical Reviews: At every level of a software development, we have to take reviews from a technical team.
- 5 Software Configuration Management: Manage the software configuration when any change in the software occurs.
- 6 Work Product Preparation and Production: This are activities use to create models, document and forms in the software.
- 7 Reusability Management: We have to store reusable item in the backup.

8 Measurement: This activities used to assist the software team in delivering the required software.

\* Explain General Principles of a Software.

⇒ This are the General Principles of a Software:

1 The Reason it All Exists:  
Before the creation of software we to think about the software it is exists in a real world or not.

2 Keep It Simple, Stupid:  
We have to create software in a such way than user can understand easily but not in a stupid way.

3 Maintain the Vision:  
At the time of Creation, We have to follow the vision of a software.

4 What You Produce, Other will Consume:

At the time of creation, we have to create software such a way that a new person can understand easily.

5 Be Open to the Future:

When we create the software then we have to know about the future scope of a software.

6 Plan ahead for Reuse:

We have to create software such a way then we can use these things in a future.

7 Think:

At the time of creation, we have to think about all the possibilities of a software.

\* Explain Software Myths:

=> There are three possible levels of software myths:

1) Management level Myths

- 2) Customer Level Myths
- 3) Practitioner's Myths

## 1 Management level Myths:

Management level People are think, we can add more Programmer than we get fast Software development.

But, In a Reality this is only one Myths.

When more Programmer work together in a one Project than Programmer some time fell Misscommunication or Misscoordination.

When Misscommunication is occurces than Software can not be develop Properly.

## 2 Customer Level Myths:

Customer are think we have to just give a problem Statement for create a Softwar.



Customer think, we do not have to share all the requirement or information to the developer.

But, In a Reality this is only one Myths.

When Customer do not tell all the requirement of Software than developer can design the software according its own thought.

### 3 Practitioner's Myths:

Developer are think, Onces Program is run than we do not have to test.

But, In a Reality this is only one Myths.

After the Program is run, We have to test the Program or take User Feedback also.

If We can not test the Program than customer does not get Proper Software.