**Unite-1**

**Q.1) Define Software Engineering?**

* Software engineering is the application of principles used in the field of engineering, which usually deals with physical systems, to the design, development, testing, deployment and management of software systems.

**Q.2) What is Software crisis? Explain with suitable diagram.**

* Software Crisis is a term used for the difficulty of writing useful and efficient computer programs in the required time. The crisis was due to using the same workforce, same methods, same tools . Even though rapidly increasing in software demand, the complexity of software, and software challenges. The condition arises due above issues is called a software crisis

|  |
| --- |
| **SOftware crisis.png** |

**Q.3) What is Software myths? Explain with suitable diagram.**

|  |
| --- |
| **Myth1.png** |

* myths are false beliefs or interpretations which creates major problems for management and technical people.
* **1) Management myths:**

1. We have all the standards and procedures available for software development.

**Facts:** Experts do not know all the requirements for the software development.

1. The addition of the latest hardware programs will improve the software development.

**Facts:** The role of the latest hardware is not very high on standard software development but rather that hardware may be misused.

1. With the addition of more people and program planners to Software development can help meet project deadlines or completion or release.

**Facts:** Adding more people will merely make the problem worse, because educating and training session the newcomers.

* **2) Customer myths:** Customer has myths leading to false expectations & that’s why you create dissatisfaction with the software developer.
  1. A general statement of intent is enough to start writing plans (software development) and details of objectives can be done over time. **Facts:** Official and detailed description of the database function are important.
  2. Software requirements continually change, but change can be easily accommodated because software is flexible. **Facts:** Impact of change varies with the time. At early (before design or code has been started), the cost impact is relatively small but grows with respect to time passes
* **3) Practitioner’s Myths:**

1. They believe that their work has been completed with the writing of the plan. **Facts:** It is true up to every 60-70 %.
2. There is no other way to achieve software quality, until it is “running”. **Facts:** Not true all the time.
3. An operating system is the only product that can be successfully exported project. **Facts:** Along with this right document brochures and booklets are also required to provide guidance & software support.

|  |
| --- |
| **generic view.png** |

**Q.4) Explain generic View of Software Engineering.**

* These can be categorized into three generic phases as follows:
* **1) Definition Phase:** The definition phase focuses on “What”. what Information needed for processing. What functions are required. what Expectations about the capacity. Interface which is established. what Area of the validation. During this, three major tasks will performed:1) System or information engineering. 2) Software project planning. 3)Requirements analysis.
* **Development Phase:** The development phase focuses on “How”. How data are to be structured. How function is to be implemented within a s/w architecture,. How interfaces are to be characterized, how the design will be translated into a programming language. How testing will be performed. During this, three major tasks will performed- 1)Software design 2) Code generation, 3)software testing.
* **Support Phase:** The support phase focuses on “Change”.  Associated with error correction, Adaptations required as the software’s environment evolves, Changes due to enhancements in changing customer requirements. Four types of change are encountered during the support phase:1) Correction 2) Adaptation 3) Enhancement 4) Prevention.

**Q.5) Explain with suitable diagram Software Engineering a layered approach.**

* **1)Tools:** This layer contains automated or semi-automated tools that offer support for the framework and the method each software engineering project will follow.
* **2) Method:** This layer contains the methods, the technical knowledge and “how-tos” in order to develop software.
* **3) Process:** This layer consists of the framework that must be established for the effective delivery of software**.**
* **4)A Quality Focus:**This layer is the fundamental layer for software engineering. Focuses to test the end product to see if it meets its specifications. Efficiency, usability, maintenance and reusability are some of the elements to be met by new software0

**Q.6) Expalin five process maturity levels of SEI.**

* To determine an organization’s current state of process maturity, the SEI uses an assessment that results in a five point grading scheme. The grading scheme determines compliance with a capability maturity model (CMM). SEI developed ﬁve process maturity levels that are deﬁned in the following manner,
* **1)Initial-**The software process is characterized as inconsistent, and occasionally even chaotic. Success of the organization majorly depends on an individual effort, talent, and heroics.
* **2)  Repeatable -** This level of Software Development Organization has a basic and consistent project management processes to track cost, schedule, and functionality.
* **3) Defined -** The software process for both management and engineering activities are documented, standardized.
* **4) Managed -** Management can effectively control the software development effort using precise measurements. At this level, organization set a quantitative quality goal for both software process and software maintenance.
* **5) Optimizing -** The Key characteristic of this level is focusing on continually improving process performance through both incremental and innovative technological improvement.

**Q.7) What is Software Development Life Cycle with diagram?**

|  |
| --- |
| **SDLC 2.png** |

* **-**SDLC is a systematic process for building software that ensures the quality and correctness of the software built. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. SDLC provides a framework for a standard set of activities  & deliverables. SDLC defines a methodology for improving the quality of software and the overall development process.

1. **Requirements Gathering and Analysis:** This phase is the starting point of the SDLC, where the software requirements are gathered from the stakeholders, analyzed, and documented. The goal of this phase is to understand the customer's needs and to ensure that the software development team has a clear understanding of what is expected from the project.
2. **Design:** In this phase, the software design is created based on the requirements gathered in the previous phase. The design should include the architecture, user interface, and the overall structure of the software. This phase also includes the selection of the appropriate technology and tools to be used in the development process.
3. **Implementation:** In this phase, the actual software development takes place. The code is written, tested, and integrated with other modules to create a complete product.
4. **Testing:** This phase is crucial to ensure the software meets the quality standards and specifications. It involves conducting various types of tests, such as unit testing, integration testing, system testing, and acceptance testing, to identify and fix any bugs or defects in the software.
5. **Deployment:** In this phase, the software is deployed to a production environment and made available to the end-users. This phase also includes the necessary training for the users and the preparation of the necessary documentation for the software.
6. **Maintenance:** This phase involves ongoing support and maintenance of the software, including fixing any bugs or defects, updating the software to address changes in the business requirements, and enhancing the software to improve its performance and usability.

|  |
| --- |
| Prototyping model.jpg |

**Q.8) Explain Prototyping Model with diagram.**

* Some time customer does not identify detailed I/O and Developer not sure about algorithm/machine/ Operating System. In such case will select this model. In this model prototype is built, tested, and reworked until an acceptable prototype is achieved. It is an iterative, trial and error method which takes place between developer and client. This model refers to the functionality of the product under development, but may not actually hold the exact logic of the original software.
* **RAD model-** Rapid application development (RAD)is an incremental software development process model that emphasizes an extremely short development cycle. If requirements are well understood and project scope is constrained, the RAD model create a “fully functional system” within very short time periods . Primarily used for information systems applications, the RAD approach consists of the following phases-

1. **1)Business modelling** - How information ﬂow among business model by asking question. What information is generated? Who generates it? Where does the information go? Who processes it?
2. **2) Data modelling-** In this actual information ﬂow of each object are identiﬁed and the relationships between these objects deﬁned.
3. **3)Process modelling-**Deals with necessary info. flow. This model defines implementation of a business function like adding, modifying, deleting, or retrieving a data object.
4. **Application generation-** It works to reuse existing program components by using automated tools.
5. **Testing and turnover-** RAD emphasizes reuse, so many of the components have already been tested but however, new components must be tested.

**Q.9) Explain Incremental Model with diagram**

* **Evolutionary Software Process Models-** In Evolutionary models are iterative in nature. This model is a combination of  incremental and iterative models. In the evolutionary model, all work divided into smaller chunks. -The incremental model combines elements of the linear sequential model (applied repetitively) with the iterative techniques of prototyping. - First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. After that many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.

|  |
| --- |
|  |

* **The Spiral Model –** This Spiral model is a combination of iterative development process model and sequential linear development model. Specific activities performed in one iteration (spiral) where the output is a small prototype of the large software. The same activities are then repeated for all the spirals until the entire software is built - This model is best used for large projects which involve continuous enhancements.

|  |
| --- |
|  |

**Q.10) What is Software Development & Process Framework ?**

* **Software Development & Process:** As a Framework for the tasks that are required to build high-quality software. But in fact technical methods and automated tools . **OR** A structured set of activities required to develop a software system. **OR** As a Framework for the tasks that are required to build high-quality software. But in fact technical methods and automated tools.
* **Process Framework:** Software Process Framework is an abstraction of the software development process. It details the steps and chronological order of a process.

|  |
| --- |
| ProcessFramFinal.png |

* **Software process includes:**
* **Tasks –** focus on a small, specific objective.
* **Action –** set of tasks that produce a major work product.
* **Activities –** group of related tasks and actions for a major objective.
  1. The process of framework defines a small set of activities that are applicable to all types of projects.
  2. The software process framework is a collection of task sets
  3. Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.