**MODULE :1**

**SE- OVERVIEW OF IT INDUSTRY**

1. What is software? What is software engineering?

**Software is a set of instructions,data or**

**Programs used to operate computers and execute**

**Specific tasks.software is a generic term used to refer**

**To applications,scripts and programs that run on a device.**

**SOFTWARE ENGINEERING**

**Software engineering is the branch of computer science**

**That deals with the design,development,testing and maintenance**

**Of software applications.**

2)Explain types of software.

**1)application software**

**2) system software**

**3)programming language**

**1) system software:- provides the basic functions for computer usage and helps to run the computer hardware and system.**

**-system software is used to manage the computer itself.**

**-it runs in the background maintaining the computer’s basic functions so users can run high level application software to perform certain tasks.**

**2)application software:-application software is a type of computer program that performs a specific personal, educational, and business function . Each application is designed to assist end-users in, accomplishing variety of tasks, which may be related productivity, creativity and communication.**

**Types of application software:1)mobile app**

**2)desktop app**

**3)web app**

**that run on mobile ex. Instagram,facebook etc.**

**stand-alone in a desktop or laptop computer.**

**3)programming language : programming language is the process of designing ,writing ,testing ,debugging, and maintaining the source code of computer programs.**

**-this software is pawritten in a programming language.**

**-the purpose of programming is to create a programe that exhibits a certain desired behavior.**

**Ex. c++, java, html, python.**

**3) what is SDLC? Explain each phase of SDLC.**

**-SDLC is a structure imposed on the development of a software product that defines the process for planning , implementation , testing , documentation , and ongoing maintenance and support. There are number of different development models.**

**-A software development life cycle is essentially a series of steps, or phases that provide a model for the development and lifecycle management of an application or piece of software.**

**SDLC phases:-**

|  |  |
| --- | --- |
| **Requirements collection/gathering** | **Establish customer needs** |
| **analysis** | **Model and specify the requirements-what||** |
| **Design** | **Model and specify the solution –why||** |
| **implementation** | **Construct a solution in software** |
| **testing** | **Validate the solution against the requirements** |
| **maintenance** | **Repair defects and adapt the solution to the new requirements.** |

**REQUIREMENT GATHERING:-**

**-features**

**- Usage scenarios**

**-although requirements may be documented in written form , they may be incomplete, unambiguous, or even incorrect.**

**-requirement will change.**

**-inadequately captured or expressed in the first place**

**-user and business needs change during the project.**

**-validation is needed throughout the software lifecycle ,not only when the- final system is delivered.**

**-build constant feedback into the project plan.**

**-early prototyping can help clarify the requirements**

**-requirements definitions usually consist of natural language ,**

**Supplemented by diagrams and tables**

**ANALYSIS PHASE:-**

**-the analysis phase defines the requirement of the system , independent of how these requirements will be accomplished.**

**-this place defines the problem that the customer is trying to solve.**

**-the deliverable result at the end of this phase is requirement document.**

**-ideally , this document states in a clear precise fashion what is to be built.**

**-this analysis represents the –“what” phase.**

**-the requirement documentaries to capture the requirements from the customer’s perspective by defining goals.**

**-this phase starts with requirements document delivered by the requirement phase and maps the requirement into architecture.**

**- the architecture defines the components , their interfaces and behavior**

**-this phase represents the –“How” phase.**

**DESIGN PHASE:-**

**-implementation plan**

**-critical priority analysis**

**-performance analysis**

**-test plan**

**-the design team can how expand upon the information established in the requirement document.**

**-the requirement document must guide this decision process.**

**-analyzing the trade -offs of necessary complexity allows for many things to remain simple which ,in turn will eventually lead to a higher quality product.the architecture team also converts the typical scenarios into a test plan.**

**IMPLEMENTATION PHASE:-**

**-in the implementation phase , the team builds the components either from scratch or by composition.**

**- Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though Version Oct 2020 Page 18 TOPS Technologies 1 there isstill room for innovation and flexibility.**

**-For example, a component may be narrowly designed for this particular system, or the Component may be made more general to satisfy a reusability guideline.**

**-Implementation – Code**

**- Critical Error Removal**

**- The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself. There are already many established techniques associated with implementation.**

**TESTING PHASE:-**

**- SimpLy stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.**

**- It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality. - A customer satisfied with the quality of a product will remain loyal and wait for new Functionality in the next version.**

**- Quality is a distinguishing attribute of a system indicating the degree of excellence.**

**- Regression Testing**

**- Internal Testing**

**- Unit Testing**

**- Application Testing**

**- Stress Testing**

**- The testing phase is a separate phase which is performed by a different team after the implementation is completed.**

**- There is merit in this approach; it is hard to see one‘s own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times. - Unfortunately, delegating (alternate) testing to another team leads to as lack (dull) attitude regarding quality by the implementation team.**

**- If the teams are to be known as craftsmen, then the teams should be responsible for establishing high quality across all phases.**

**- An attitude change must take place to guarantee quality. Regardless if testing is done after the-fact or continuously, testing is usually based on a regression technique split into severalmajor focuses, namely internal, unit, application, and stress .**

**MAINTENANCE PHASE:-**

**-Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.**

**- Software maintenance is also one of the phases in the System Development Life Cycle (SDLC), as it applies to software development. The maintenance phase is the phase which comes after deployment of the software into the field.**

**- The developing organization or team will have some mechanism to document and track Defects and deficiencies. Version Oct 2020 Page 19 TOPS Technologies 1 Now we'd like to briefly introduce to you a few diagram notations which you'll see in the tutorial below. External Entity An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people used to draw external entities on the edge of a diagram.**

**- configuration and version management**

**- reengineering (redesigning and refactoring)**

**- updating all analysis, design and user documentation**

**- Repeatable, automated tests enable evolution and refactoring Maintenance is the process of changing a system after it has been deployed. Corrective maintenance:identifying and repairing defects Adaptive maintenance: adapting the existing solution to the new platforms. Perfective Maintenance: implementing the new requirements In a spiral lifecycle, everything after the delivery and deployment of the first prototype can be considered ―maintenance‖!**

**- Software just like most other products is typically released with a known set of defects and deficiencies. The software is released with the issues because the development organization decides the utility and value of the software at a particular level of quality outweighs the impact of the known Defects and deficiencies.**

**WHAT IS DFD?**

**the information flows within a system. A neat and clear DFD can depict a good amount of the system A data flow diagram is a traditional way to visualize requirements graphically. It can be manual, automated or a combination of both.**

**It shows how information enters and leaves the system. What changes the information and where information is stored. The purpose of a DFD is to show the scop**

**What is flow chart?**

**A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbol are linked together with arrows showing the process flow direction.**

**A flow chart is a graphical representation of the operations involved in a data processing system.**

**-symbols are used to represent particular operations or data.**

**-flow lines indicate the sequence of operations. (top to down sequence.)**

**What is use case diagram?**

**A use case diagram is a graphical depiction of a user’s possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circle or elipse.**