CS5800-Adv Software Engineering-SRINATH KALLU-900741822

1. Explain various activities in design phase of the application

Answer:

Design phase: In this phase we consider the physical solutions of the application. The major role is played by the UI (USER INTERFACE).

* Design and integrate the network
* Design the application architecture
* Design the user interface
* Design the system interface
* Design and integrate the database
* Prototype the design details
* Design and integrate system controls

2.Explain the importance of non functional requirements and how they impact your architecture?

Answer:

Nonfunctional Requirements define system attributes like:

* Security:

This mainly concerns with the security of your application. For example hoe secure your data is , like are you using any security software’s or not.

* Reliability:

Reliability is an attribute of any computer related component that consistently perform according ti its Specifications.

* Performance:

It says that how fast your system can wrok or access.

* Maintability:

It should adopt any kind of technology even if there are new technology available in the market.

* Scalability:

It is the ability of the computer application or product to continue to function well when it is changed in size or volume in order to meet a user need.

* Usability:l

This deals with the interaction and graphical user interface. It is the measure of a product potential yo accomplish the goals of the user.

3.Give 10 best practices to follow in developing an application or software engineering?

Answer: The 10 best practices are explained below:

* Methodolgy:
* Architecture:

A good system architect will ensure that the suitable architecture is selected, keeping in mind the requirements as well as the limitations and constraints, if any. Best practices such as identifying the threats and anti-patterns in the system are very helpful

* Documentation:

As important as the actual software itself, are the documents that support it –project plan, requirement specifications, High Level Design (HLD), Low Level Design (LLD), test plans, test reports, status reports and user documentation. Many a time, these documents are a part of the deliverables specified by the customer or stakeholders as well. These documents help to maintain understanding of the software, ensure trackability, and remove dependency upon the core development team. They can be used as a reference in future by someone else, who might work on or use the software

* Maintability:

Even while the deployed software is operational, established processes and practices must be in place to support user problems and report them back to the maintenance team effectively. Customer resolution software and defect tracking mechanisms are vital here

* Data migration:

This shows how effectively the data is transfered between the computer storage types .

* Estimation:

Estimating the budget, time for developing comes under this phase

* Requirements:

The software , hardware, financial , security, everything comes under this phase.

* Testing and validation:

After the application is developed it goes to testing and validation phases.

* Quality control: this phase checks the quality of various phases the applivcation has undergone.
* Effective Installation and Deployment:

This is the final phase . After completing the above phases with out any aligations it shows that you application is ready to implement.

4. Why would you go for Waterfall methodology? How does it differ from Agile Methodology?

Answer:

Waterfall Model;

The water fall model is a linear sequential design approach for software development, in which the progress flows in one direction . It goes through the phases of conception, initiation, analysis, construction, testing, deployment and maintenance.

Agile Model:

It is the methodology of modeling and documenting software system based on best practices. It is a collection of values and principles, that can be applied on an software development.

|  |  |
| --- | --- |
| Waterfall Model  1.In this model the sequence of events are   * Gathering and documenting requirements * Design * Code and unit test * Perform system testing * Perform user acceptance testing * Fix any issues * Deliver the finished product   2.Planning and designing more straightforward  3.Progress is more easily measured.  4.Except for reviews , approvals, etc., a customer presence is not strictly required after the requirement phase.  5.Funding is easy .  6.The difficult part is gathering the information and documnerting in a way that is useful for the customer is often most difficult in the siftware development.  7.Another drawback is the possibility that the customer wil be not satisfied with their deliveried product. All the deliverables are based upon documented requirements , a customer may not see what will be delivered until it’s almost finished. By that time , changes can be difficult to implement | Agile Model   1. This approach emphasizes the rapid delivery of an application in complete functional components. Rather than creating tasks and schedules, all time is “time-boxed” into phases called “sprints”. 2. As work is completed, it can be reviewed and evaluated by the project team and customer, through daily builds and end-of-sprint demos. Agile relies on a very high level of customer involvement throughout the project, but especially during these reviews.   3.The customer has frequent and early opportunities to see the work being delivered, and to make decisions and changes throughout the development project.  4. The customer gains a strong sense of ownership by working extensively and directly with the project team throughout the project.  5. The very high degree of customer involvement, while great for the project, may present problems for some customers who simply may not have the time or interest for this type of participation.  6. The close working relationships in an Agile project are easiest to manage when the team members are located in the same physical space, which is not always possible |

5.Define terms-Software and Software and Software Engineering? How web Impacted Software Engineering?

Answer:

Software:

It is a part of computer system that consists of data or instructions in contrast to the physical hardware from which the system is built.

Software Engineering:

It deals with applying the principles and techniques of computer science, engineering, and mathematical analysis to the design, development, testing, and evaluation of the softwarwe is called as Software Engineering.

Impact of Web on Software Engineering:

* The days when there is no web technology, we used to do all the work with lots of effort. For example, to update something in an administration it used to take a lot of time. Because each system should be updated individually. They used to waste a lot of time and puts huge effort.
* The Web has encouraged the accessibility of programming administrations and the likelihood of growing exceedingly circulated benefit based frameworks. Online frameworks improvement has prompted vital advances in programming dialects and programming reuse.
* The development of the World Wide Web has had a profound effect on software engineering. The Web was primarily a universally accessible information store and it had slight effect on software systems.
* This meant that web-based systems could be developed where, instead of a special-purpose user interface, these systems could be accessed using a web browser. This led to the development of a vast range of new system products that delivered innovative services, accessed over the Web.

6. Explain Service Oriented Architecture? Give an example how would you use this in your project

Answer:

A service oriented architecture is a style of software design where services are provided to the other components by application components, through a communication protocol over a network.

The factors involved in Service Oriented Architecture:

* Application frontend:
* Service:
* Implemetation:
* Data:
* Business Logic:

We shall discuss in brief , how SOA is related to my project.

Application frontend:

In my application we have admin login, Employee login, departments, etc.,

Service:

The main service of my application is to provide the best employee to the multinational companies.

Implemetation:

The implementation is done by checking all the necessary things like security, markert,etc.,

Data:

Storage is the main thing for implementing the application.

Business Logic:

The main logic of my application is now a days the unemployment is rising high. Many large scale industries are sort of worthfull employes or missing the employees who are multitalented. So in order to provide the talented employees to the companies and making sure that the worthy candidates are not missing the proper jobs