**BA/DA TRAINING**

**Assignment – 2(July 8,2019)**

**SHAFALI GUPTA**

**Q1: Difference between Product based, and Service based companies.**

**Ans: Product based company**

As the name says typically companies, which creates a product either for enterprises or for people comes under this category. For example, Microsoft. However, you may not call all employees in Microsoft as product developer. These product-based companies also require people who maintains existing product and helping customers. Almost most of the product companies provides warranty and that is one of the selling points in a product.

**Service based company**

A typical Indian IT company like TCS, which provides services to its customers at a price agreed by both parties usually bound by Service level agreements. These companies offer technical consulting as well to clients. Say Microsoft releases azure and customer wants to upgrade or rebuild his sites to azure, if he approaches Microsoft the chances are very less that they might accept it, it's not because they can't do but it's because the cost will be very high and usually it's not that profitable compared to a new product. Service based companies fill this void. They will have plethora of verticals in which they will have readymade solutions and experts who can understand the user requirement and suggest the best way to do it and implement it using its own people or other companies.

For obvious reasons product development needs good understand of systems and different data structures and algorithms. The bar for service-based companies is less compare to product Development Company.

Indian IT companies induces fresh college graduates and trains them so that they can be utilize in various projects. It will not be sane case with product development.

**Q2: What are stakeholders?**

**Ans:** A stakeholder is a party that has an interest in a company and can affect or to be affected by the business. The primary stakeholders in a typical corporation are its investors, employees, customers and suppliers.

There are many people involved in getting a project from inception to a successful completion. You are going to have to know how to manage each and everyone one of them, even those who do not work directly under you. One such person is the project stakeholder.

**Q3: Challenges while implementing a software engineering.**

**Ans: I**mplementing a new software solution carries with it several challenges. The process can be overwhelming, confusing and lengthy. Knowing the common challenges implementation projects present can help organizations avoid them. With a solid project plan and realistic goals, even the most complex implementation can realize success and return on investment in a reasonable amount of time.

1. **Understanding large and complex system requirements is difficult –**  
   The word ‘large’ represents 2 aspects:
   * Large constraints in terms of security, etc. due to a large number of users.
   * Large number of functions to be implemented.

The complex system requirements include those requirements, which are unclear and difficult to implement.

1. **Undefined system boundaries –**  
   There might be no defined set of implementation requirements. The customer may go on to include several unrelated and unnecessary functions besides the important ones, resulting in an extremely large implementation cost, which may exceed the decided budget.
2. **Customers/Stakeholders are not clear about their needs. –**  
   Sometimes, the customers themselves maybe unsure about the exhaustive list of functionalities they wish to see in the software. This might happen when they have a very basic idea about their needs but have not planned much about the implementation part.
3. **Conflicting requirements are there:**There is a possibility that two different stakeholders of the project express demands, which contradict each other’s implementation. In addition, a single stakeholder might also sometimes express two incompatible requirements.
4. **Changing requirements is another issue:**   
   In case of successive interviews or reviews from the customer, there is a possibility that the customer expresses a change in the initial set of specified requirements. While it is easy to accommodate some of the requirements, it is often difficult to deal with such changing requirements.
5. **Partitioning the system suitably to reduce complexity:**   
   The projects can sometimes be broken down into small modules or functionalities, which handled by separate teams. Often, more complex and large projects require more partitioning. It needs to ensure that the partitions are non-overlapping and independent of each other.
6. **Validating and Tracing requirements:**   
   Crosschecking the listed requirements before starting the implementation part is very important. In addition, there should be forward as well as backward traceability. For example, all the entity names should be the same everywhere, i.e., there should not be a case where ‘STUDENT’ and ‘STUDENTS’ are used at separate places to refer to the same entity.
7. **Identifying critical requirements:**Identifying the set of requirements, which needs to implement at any cost, is very important. The requirements will be prioritized so that crucial ones can be implemented first with the highest priority.
8. **Resolving the “to be determined” part of the requirements:**   
   The TBD set of requirements include those requirements which are yet to be resolved in the future. The number of such requirements should be kept as low as possible.
9. **Proper documentation, proper meeting time and budget constraints:**Ensuring a proper documentation is an inherent challenge, especially in case of changing requirements. The time and budget constraints too need to be handled carefully and systematically.

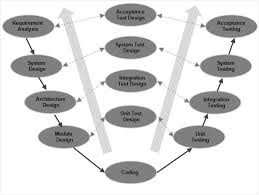
**Q4: V-model Concept**

**Ans:** V model is a linear development methodology used during a Software development life cycle SDLC. V-Model encourages testing preparation to occur in parallel with requirements, design, and development phases. It is also known as Verification and Validation model.

Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the ‘V’ and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.

Works well for smaller projects where requirements are very well understood. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated testing phase. Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.



**Advantages:**

* V-Model is used for small projects where project requirements are clear.
* This is a highly disciplined model and Phases are completed one at a time.
* Simple and easy to understand and use.
* This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product.

**Disadvantages:**

* V model is its inability to adapt to any necessary changes during the development life cycle.
* It is not a good for complex and object-oriented projects.
* It is not suitable for projects where requirements are not clear and contains high risk of changing.
* This model does not support iteration of phases.
* Timeline restriction