**Assignment 1: Software Requirements**

**Date: Monday Sept 07, 2022 Due date: Monday Sept 14, 2022**

**------------------------------------------------------------------------------------------------------------------------------------------**

**1. State the differences between software engineering and system engineering**

Software Engineering:

* A person who deals with the design and development of superior quality software applications/software products.
* Software Engineering deals with systematic and disciplined approach to design, development, and maintenance of software applications/products.
* In general, it concerns all aspects of software development, database, and other modules in software development life cycle.
* Software engineering mostly focuses on developing quality software by following the disciplined approach for cost estimation.
* Some Software Engineering Methods are Agile Methodologies, Model Driven Development, Continuous Integration and Continuous Development.

**System Engineering:**

* A System Engineer is a person who deals with the overall management of engineering projects during their life cycle (focusing more on physical aspects).
* System Engineers follows an interdisciplinary approach governing the total technical and managerial effort required to transform requirements into solutions.
* In general, they are concerned with all aspects of computer-based system development including hardware, software, and process engineering.
* System engineers mostly focus on users and domains.
* Systems Engineering Methods are Stakeholder Analysis, Configuration Management, Systematic Verification and Validation, Requirements Engineering etc.

**2. Does software wear-out? Explain your answer?**

Software is developed not manufactured. Software does not wear out like hardware.

When manufacturers build hardware, it tends to show errors or faults. Therefore, it goes through many trials and errors steps to remove the defects in it. As a result, the final product becomes ready to release and use in production. But then, when in use, there can be a certain number of causes that might decrease the performance of the hardware. For example, excessive temperature, dust, vibration, improper use and so on. All these effects cause the hardware to start showing failure rate again. With time, the failure rate keeps on rising. And at one point, the hardware just stops functioning. One can then do whatever they want to do with the piece of hardware, but just cannot make it work again. Let us not consider repairing it here. Repairing can make the hardware perform again, but still with time, the same effects will not function again.

On the other hand, software does not wear out. Like hardware, software also shows a high failure rate at its infant state. Then it gets modifications, and the defects get corrections and thus it comes to the idealized state. This idealized state continues.

However, software in a steady or idealized state may need modifications as user’s demand from the software can change, the platform on which runs can have updates which might not allow the software to perform etc. A software can get updates easily, unlike hardware, which requires physical interaction. However, after fulfilling one demand, another one might rise. And in this way, alternative software with implementation of current user demands can replace software. Though, not having a recent feature is not a defect, users tend to use the latest alternatives. If we consider this as a failure for the software, then the failure rate increases with time. This will make the software deteriorate due to change, but still the software can perform its operation as it was performing in the beginning.

That is why software does not wear out.

**3. Show with an example, how Software Engineering in industry is different from coding assignments.**

Software Engineering deals with distinct parts of the product development process from planning to deployment, and maintenance. Software Development Life Cycle (SDLC) helps in understanding every part of product development.

Let's see with an example. I am planning to develop a web-based application and my requirements will be changing in time as the new features might come onto the market which helps me in using them to display my data in UI.

So initially, I had decided to use Agile Methodology to do my work in a systematic and disciplined way. When I decided to use Agile Methodology, My Business and development team at the top level interacts and they create dummy web-based application using some tools (Like Paint and all...) to give an idea about how it looks like and divided my web-based application work into smaller chunks we call them as stories to start with real development process.

Once done with all business, the coding starts at this point and the development team will be involved in coding the web-based application using the technologies decided by the Business and development team at the top level. Only this part of the SDLC cycle coding assignments will be involved in.

After coding is done, we need to use Continues Development (CD) and Continues Integration (CI) to enhance and maintain the web-based application.

**4. Explain the following diagram: Software development process**

In software development process, At the beginning of the project it takes some time in our efforts to process the project like management part either related to planning the project, budget estimation of the project and many aspects related to management. Along with that we will have our first version of the project gets started and will be able to see some outputs which means productive work. May be due to work conflicts at the beginning we might also waste some time in our efforts as one can able to move forward in their work as they got stuck.

Once time starts moving, the management part starts decreasing and output starts increasing as the project started and versions are coming one after the other in regular intervals as planned. Even wastage of time will be less while approaching the end of the project.

X- axis shows the time span of the project from beginning till the end.

y-axis shows the percentage of effort used.

We can see three things inside the graph (1. Process, 2. Productive Work, 3. Thrashing) like how much percentage of effort we have placed from the beginning till the end of project on those three constraints can be shown in the graph.

**5. Is Code-and-fix model could be used to develop large project? Explain your answer?**

Code-and-fix model is used when we have short projects or low budget projects. Sometimes code-and-fix models save time in small projects. To use the code-and-fix model we need to have an extensive planning to start the project.

It is good to use code-and-fix when only small groups of people work together, in the large organizations we need to connect with different follow employees for the different modules of code in our project and integrate the code and deploy it, in this scenario it is difficult to do with code-and-fix model.

In startups we see code-and-fix model is used as the it is a smaller organization with one limited people working on the project for smaller modules.

For Example, one person is working on an application development. If we divide the project into smaller modules. One can code one module, fix the issues in the module, and test the module. Repeat the process for all the modules until one finish's all the modules to finish the project.

In another scenario, In the larger organizations group of people working as a group of teams to delivery one product. If we use code-and-fix model, we will end up with lot of conflicts and individual modules are with individual employees. Even once everyone done with the development, fix, and testing of their own modules we need to integrate all modules, fix issues and test will be a big issue as there is no communication while coding, every individual might use diverse ways of coding which might not go well when integrated.

So, it is not recommended to use the code-and-fix model when dealing with larger projects or larger organizations with a lot of people working together on the same project.

**6. Describe the waterfall model for software development and list three of its advantages for software development**

The waterfall model is the classic life cycle model. It is also known as common sense model. As the name suggests, the flow of model will be from top to bottom. Moving upstream is going to cost a lot for the project. In the waterfall model, the next stage will be started only once the previous stage has finished fully and successfully.

Various stages in the waterfall model are:

1. System Requirement
2. Software Requirement
3. Preliminary Design
4. Detailed Design
5. Code and Debug
6. Test
7. Maintain

* If any project has a clear requirement which are not going to be changing as time passes, or new features adding to the project. Then we can very well use waterfall model for effective way of developing software product.
* This model is easy to understand and follow. It is easy to manage due to rigidity of the model – each phase has a specific deliverable and a review process.
* Early determination of end goals. When we are using the waterfall model, we should be aware of the requirements, and planning is done before starting the code. So, End goals will be visible before the start of the project.

In the Waterfall model, the work will be in stages. One cannot stop in the middle, get to the starting point, and do updating in the waterfall model. We cannot use waterfall model in a project If it is going to get regular updates. We cannot do continues Integration (CI) or Continues Development (CD) using waterfall model.

**7.. Describe one real world scenario where the Waterfall model is applicable**

For example, we will see how we can use waterfall model while constructing a simple chair.

At the beginning of the waterfall model, we must do Analysis. Gathering system requirements are nothing but the tools needed to do the Job. We need to plan the tools and material needed to construct the chair. Like we will list all the requirements like

* type of chair we want to make
* Tools needed to make the chair like carpeting tools and others.

Once done with the system requirement and software requirements analysis, we need to have a design. So, we will create a sample design, like what the chair is going to look like. We will do some documentation for the same. We are going to have the documentation in every stage of the waterfall model.

After we are done with design, we need to construct an actual chair in the next stage. By using the requirements and design we got, we will start constructing the chair.

Once construction is done, we are going to test it. We are going to put some weight on the chair and see to what extent the chair can resist. If the chair fails to resist the weight, we need to construct from the start. This is going to cost a lot for the project. Which is why we need to use some other model to construct chairs to decrease the risk involved in finishing the project.

**8. Is it appropriate to use the Waterfall model to develop an application that is likely more changes in requirements will be requested by the customer? Explain your answer? If not, suggest the most proper software development life cycle model**

No, we cannot use waterfall model when we are expecting more changes in requirements. In the waterfall model we need to be clear about the requirements before we begin the process. Upstream in waterfall model is going to cost lot of effort in the projects.

When we are expecting constant changes in the requirements of the project, I recommend following the Agile methodology, evolutionary prototyping, or staged delivery where there is lot of interaction between the development team and the customers.

**9. Assume you have been requested to develop a system, whereas its requirements are dynamically changing and were not collected yet. Is it applicable to use staged model? Explain your answer.**

Yes, we can use staged model in this scenario. Unlike the waterfall model, staged delivery model helps in interacting with the customers and knowing the feedback in regular intervals by delivering a part of project every time. If any random requirement comes from the customer, then it will be added to the process in the next stage of the development process and delivered to the customer. Once this stage is done, we will also have the feedback from the customer regarding new requirement.

**10. Assume your customer requested you to develop a high-quality application related to human safety. Identify the most appropriate software life cycle model you should use.**

I would like to use evolutionary prototyping, or staged delivery model as we need to add new features depending on the safety measures we are going to take, either it might be a phone call facility or a bus facility late at night. But Agile methodology will also help us in achieving the above scenarios.

But for example, When I had an idea of developing some application for safety of people in a town with some features like sending their location, emergency contact dial, bus pick up, helpline and many others with good UI. In this case, at the beginning I knew all my requirements and had the design ready.

So, I will go ahead and use the waterfall model. Once I am ready with my application with my first draft, it will have all the functionalities and will be deployed of use. After some time, if we wanted to change some features, or update some features in the application, we can go ahead and use the waterfall model.

While developing some standard applications, with well-known requirements can be developed using waterfall model. Even though the requirements might change in the later part of the project we can set under maintenance where we loop the waterfall model cycle for that enhancement.