**Shashank Mondrati: Final Exam: Software Engineering Part-I**

**1. a) What is software modeling?**

**Software Modeling refers to the work of implementing the software application, before any coding is done in any coding platform i.e Java, C++, Python. It is an important part of the software development process, and for that process software Models are built, and therefore they are analyzed before the software product or system, and they are used to direct the software production department.**

**b) Why do we need to use tools for software modeling?**

**We can use tools because it can help us find if there are any problems or defects in when designing the software. Helps us find consistencies, and makes sure the designers are on the right path for clients to deliver their software. With the help of software modeling tools, the designers can predict the outcome/behaviour and responses of the system on what it is made to do in various situations when the software is encountered. One other ways we can use tools for prioritizing areas for testing purposes, for example: areas that are in testing phases, we can work further upon.**

**c) List several different software modeling tools.**

**Lucid Chart, Concept Draw**

**d) How do you use software modeling to accurately (or better to say realistically) estimate the amount of the work needed to be done per task.**

**It is hard to put into words, but software modeling tools help us in describing the software. Putting in pictures helps us understand what the software is supposed to do. Tools help us find the correct domains, the software specifications, like the platform it can run on, memory allocations, user permissions, allocations to various parts, these tools also help us in defining and depicting the software’s behaviour at every response through the help of these tools. An advantage is, when designing the code, helps us get a clearer idea of what an individual is supposed to do with the amount of resources and time required by each task’s completion. Above all, software modeling helps us find the approximate amount of time required and what needs to be done per taks for the individual.**

**e) Pick a model of your project for part 2 of final exam (try to make sure you pick one that is different from your teammates) and explain how you could determine tasks' complexities and estimate the load from it. Which model works the best for estimation and why?**

**Software model for our work would be the Waterfall model, it is a linear sequential flow. I thought it’s best for me to go in a straight linear way, with no shortcuts. The advantages are its simple and linear, and quite understandable, and the requirements are already finalized. In Waterfall modeling, usually 20-40% time is spent on finding the right requirements, and about 30-40% is spent on coding the software, and the remaining is spent on testing and alpha-beta release operations.**

**Source:** [**Waterfall Project Management Methodology · Blog · ActiveCollab**](https://activecollab.com/blog/project-management/waterfall-project-management-methodology#:~:text=Usually%2C%2020%E2%80%9340%25%20of,before%20the%20previous%20one%20ends.)

**2. Go to the following sites and read about “Principles of Manifesto for agile software development”. Then, explain the general idea behind these principles .**

**<http://martinfowler.com/agile.html>**

[**http://agilemanifesto.org/principles.html**](http://agilemanifesto.org/principles.html)

**Pick 2 of the principles of Manifesto and explain them. In your opinion, how essential your two choices are in software project development? Explain by giving examples related to your project**

**Principles of Manifesto:**

**When it comes to the principles of Manifesto, the main and utmost priority is customer satisfaction, when they are satisfied both sides win automatically. Customers are satisfied when delivery is made through early and continuously using via software. In the agile production process, even in the late quarter, by making some changes to the software will yield promising results in the game so that the customers can feel advantageous to their product via the software.**

**Delivering software more frequently, not taking too much less time is a principle of the manifesto. Taking too much to deliver a product is not a good sign of customer satisfaction, delivering frequently should lie from a couple of weeks to a couple of months, the shorter the duration the more the happier the customers are from the software they use.**

**While developing the software, business strategists and developers must work together on the project so they can have insight on what needs to be done, and the appearance design on the software. One of the best methods to receive and send information is via face-to-face conversation, with that method, new questions may arise, solutions can be solved easily, and best ways to gain insight on the software being developed. Giving constant attention and technical changes, gives out promising results, and not to mention that the best ideas for the development of the software comes from self-organizing teams.**

**The general idea is, keep working towards customer satisfaction through early delivery of the software, keeping up with regular times discussing the feedback from the customers, making some changes, reflecting on work will make agile work faster, and promising results.**

**Two Principles of Agile:**

1. **“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”**

**Above principles refers to individuals who have the insight to be motivated, and can add necessary changes to the software, if we can give them the tools they desire, they can make a sword with a dull blade. That principle applies with my group too, where one of my groupmate is blessed with HTML coding. We gave them the tools they desired, and came up with a plan in the shortest time possible. With their knowledge, they can come up with good ideas, and motivate others, and we can place their trust on them to get the project done, or any project that needs to be done.**

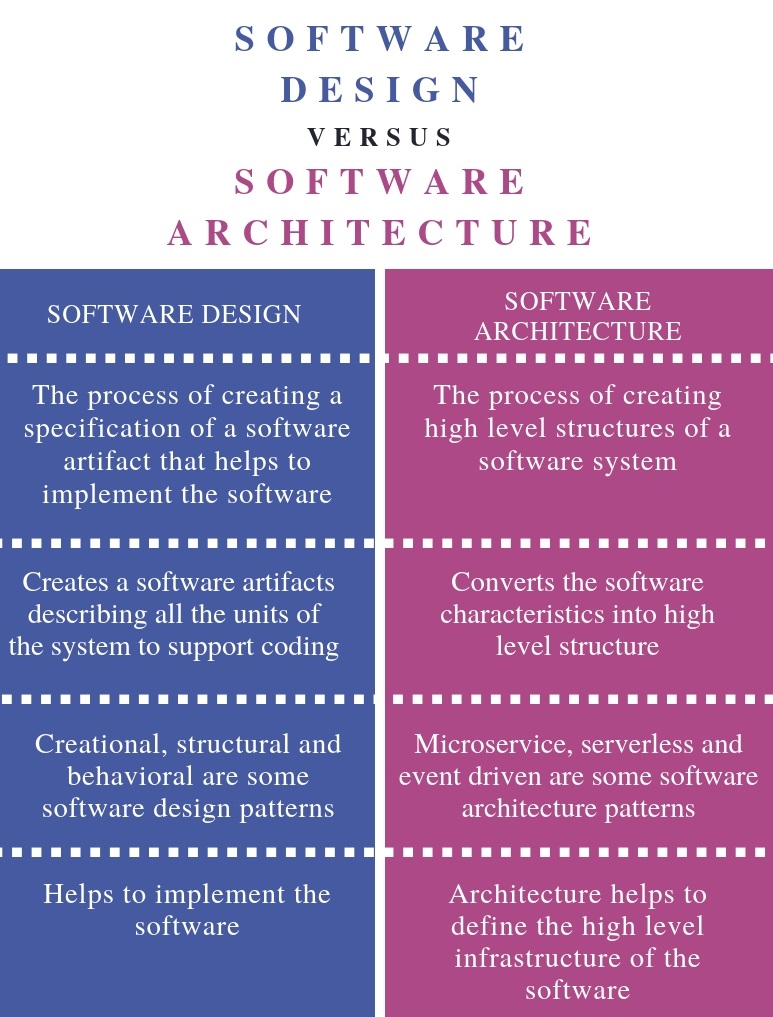
1. **“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”  
     
    This principle is very powerful to implement, these days every one is online, and preferring to be online, but with face-face conversation many things can be discussed regarding the software development.**

****

**Source:** [**Face-to-Face Communication**](http://articles-junction.blogspot.com/2013/08/advantages-and-disadvantages-of-face-to.html)

**As you can see above, there are many advantages of face to face communication, when in face-face communication, the conversation is very effective, they are willing to listen carefully, and they have your attention. Feedback is instant, when compared to online, which can delay. Feedback is informal, and review is very direct, when it comes to delicate situations, can create more solutions. Not to forget, face to face communication is very educational when it comes to interviews.**

**3. Watch the following video as well as read "whoNeedsArchitect.pdf" article and explain software architecture and software design and their relationships (15 points). Relate what you read to your project and explain whether you have a particular architecture? How about any specific design model (15 points)? (Total = 30 points)**

**Software Architecture: Defines how the computer is divided into components, and how well the components interact with themselves with the help of interfaces. Much more, these components are made of smaller components, but all this architecture is included with components and interfaces, and only the developers are known with this knowledge. Another way of putting this would be, sets of design decisions that must be made early in a project. First would be designing the components, and second would be coding platform or the language the software is designed. That coding language will be the architecture for most of the project.When it comes to software design, the role is to come up with a problem-solving mentality for a software related-problem, and is responsible for that role. Whenever there is a problem in the software architecture, the software designer is responsible for solving it with the help of the person who came up with the software architecture’s component design.**

**When it comes to their relationships, the designer helps to implement/advance the software, whereas the architect helps to define the structure of the software. Above all, if one of them has a problem, the other helps them out. When it comes our project, we share some similar principles and structure. The components in our project are work we do, planning, brainstorming. The software design are two of our groupmates, who know how to handle software problems, whereas the software architects are the remaining of our group, but we shift our roles, and help each other around.**

**Specific design would be Architectural Design: Helps us define the relationship between major components of the software design, primarily about decomposing the system so there are lot of interaction within the website.**

**Source:** [**Design Modeling in Software Engineering | Basics & Principles of DMSE (educba.com)**](https://www.educba.com/design-modeling-in-software-engineering/)