

Question 1 (5 marks):

What differences have the Web and Cloud Computing made to software engineering? Explain at least 8 of them and provide one example.

- Companies do not have to invest a large initial sum for projects. Cloud services such as Microsoft Azure have services that can host a company's application for them or provide storage services for the length of use.
- It makes things faster as now many things can be skipped such as hosting a database server or having a firewall. Cloud services have database servers and firewalls for you.
- Cloud computing has flexible scalability. This means a company can use a certain amount of a resource and if they need more or less of this resource, they can easily switch to the amount they need.
- Web services allow an application to be hosted all over the world. An application will have a reasonable response time because there is a datacenter near you.
- A company does not always have to need to use a service. With cloud computing you can choose when you need to service active, so you only must pay for what you use.
- A web service allows you to not be concerned with how something is done. This reduces the complexity of a project.
- Cloud computing services are reliable to companies. They can depend on the service.
- Virtualization of apps for more efficient use of servers

Example: A small bank wants to create a new service that is a casino for your customers. They do not have the infrastructure to created it. Instead, they use Amazon Web Services for databases and hosting.

Question 2 (7.5 marks):

- Suggest the most appropriate generic software process model.
- Propose a software architecture approach/ pattern that might be used as a basis for managing the development of the following systems:

- a) I think this fits the incremental process model because the required features can be implemented within different increments. Increment 1 – collect and store data from public online resources, increment 2 – Take the data and generate visualizations, etc.

Architecture: Collect publicly available data using API's from google, facebook, etc. -> process the data -> store the data -> AI visualization -> Retrieve data -> User search

- b) This one includes implementation of low latency sensor devices and would include embedded systems. Therefore, I think a plan-driven model would work best. The devices to be implemented need to have a predefined function. Also, safety and security are a major concern because if the product fails, life could be at risk.

Architecture: Inputs from devices (radar, motion sensor, etc) -> interpret data -> send data over network

- c) I think the spiral model works here because the requirements are very short and vague. Just do a little bit at a time and get feedback and more communication.

Architecture: Ground terminal -> Communication (Data, Commands) <- Spacecraft terminal

Question 3 (7.5 marks):

Explain how the principles underlying agile methods lead to the accelerated development and deployment of software compared with other methods. Provide one sample scenario and discuss the advantages and disadvantages of agile methods.

Agile methods accelerate development because it focuses on one feature at a time. It plans for that feature then implements it and tests it. Now you have that feature without planning every single feature first. It also focuses on the most important features first. These important features make up 80% of the product and are worked on for 20% of the time of development. So agile gets these important core features done first. This allows you to have a working product before every feature is thought up. You can continuously integrate not as important features after the fact. In comparison to the plan-driven model, you must have everything planned out ahead of time which delays the core product.

Example Scenario: Mobile banking app – Advantages: Can be developed quicker once core features implemented, nice to have features will be dealt with after such as 2FA or mortgage calculator, if a new trend among rich people is to send \$100 to 1000 random people, the bank can react fast to implementing that as a feature.

Question 4 (15 marks):

The pandemic has forced many companies to go through a remote work experience. Assume you are based in Windsor and are working on a multinational software development team. You and colleagues in Toronto, Vancouver, Tokyo, and Berlin must analyze a 150-page requirements specification for a commercial e-commerce system and deliver a report in a week and develop and test the software in 4 months.

- a)
 - 1. Time zone: When working will take place and how everyone will communicate, especially under critical incidents.
 - 2. Government regulations: You will need to deal with the government regulations of multiple countries.
 - 3. Time-limit: The time constraints for planning the whole project in a week. How long will communication with stakeholders/customers last? 150 requirements?
 - 4. Requirements list: Being given a list of 150 requirements is a very static and a lot. I think that an agile approach needs to be adopted and it will be a challenge to get a way shorter list of core requirements and nice-to-haves.
 - 5. Security and many servers: This is a project that must be very secure as it is dealing with people's money. Breaches in security are more impactful than other software's. Especially true since it will have servers all over the world.

- b)
 - 1. Time zone: Propose flexible working schedules for everyone with synchronized meetings that works for everyone.
 - 2. Government Regulations: Have these regulations part of the core requirements
 - 3. Time-limit: Have an agile development approach without handoffs so that everyone can do everyone's job. Handoffs slow down the team effort. This way everyone can work on the report, and everyone can work on development.
 - 4. Implement the core features first then the nice to have features. Figure out from customers what the core features should be.
 - 5. Use cloud services such as AWS. This allows for use of their servers all over the world with guaranteed security.

- c)
 - 1. Source control platform: For code versioning making sure everyone's changes to codes merge to existing project.
 - 2. Video Conferencing: To meet with colleagues and stakeholders/customers to discuss objectives and updates.
 - 3. Way to see database tables: Not everyone is familiar with databases, way to show data to others without querying databases (export into excel or something)
 - 4. Slide Show: Show progress and report/ideas to colleagues and customers
 - 5. Display of tasks: A list of tasks people can take on to follow an agile style

Question 5: (15 marks):

Assume your team has fully developed a mobile application and presented it to the customer, but the customer is not satisfied with the final product and does not accept it. (15 marks)

- a)
 - 1. It is more costly to change planning after implementation depending on the approach used. Agile is less costly, Waterfall is more costly.
 - 2. Depending on the issues starting over is possible.
 - 3. Team members might be scheduled for a different project after affecting those other projects.
 - 4. Loss of jobs

- b)
 - 1. Requirements not understood by team
 - 2. Wrong approach used for development
 - 3. Did not do core features first
 - 4. Absence of customer representative (Like a product owner)
 - 5. Unclear/ambiguous requirements
 - 6. Skipping software development life cycle steps
 - 7. Implementing too many features
 - 8. Lack of interviews with customer

- c)
 - 1. Use agile approach which implements a feature you can show to stakeholders every couple of weeks
 - 2. Have a customer to own the product
 - 3. Ask the customer for user stories to interpret requirements
 - 4. Construct prototypes to show to a customer for feedback prior to completion
 - 5. Spend more time on communication