SE Week 2 Assignment

1. Can you compare and contrast the Waterfall model with the Agile model in terms of flexibility, risk management, and client involvement? When would it be more appropriate to use one model over the other?

   The Waterfall Model:

   - Is more rigid due to how it flows linearly and sequentially in a single direction where the distinct phases cannot overlap just like a waterfall which keeps on flowing.
   - Risk management in this model is easier to be maintained due to its smaller-scaled nature with low complexity, stable and well-understood requirements, and regulatory environments where documentation and process standardization are critical. However, the risks may only be identifiable after all phases are complete which means a bigger quality check is very much required.

- Client management is less maintained in this model due to how less flexible the phases are so that customer feedback can only be received once every phase has been completed. A single revision means starting from scratch.

The Agile Model:

- Is more flexible and adaptable due to its prioritization of collaboration, rapid prototyping, and continuous improvement. Working in short, manageable iterations instead of bigger phases makes it easier to backtrack into an iteration if revision is required.
- With regular assessments and feedback loops, risks can be identified earlier than in the Waterfall Model.
- Has a customer-centric approach which helps ensure that the product aligns with the client's needs and improves user experience.

2. What are the typical challenges faced when transitioning from a traditional SDLC model like Waterfall to an Agile methodology? How can these challenges be addressed?

   The challenges may typically come in the form of collaboration between members and time constraints. Communication is way more important in the Agile methodology due to faster processes meaning that bigger parts have to be handled by more than a single person, and there may be a possibility documentation is not possible in such a short time frame that those members working in different aspects of the same part should be able to find ways to share their progress via Agile tools which they must also learn to use in advance to improve efficiency.

3. Make comparisons between all the SDLC models mentioned in slide 4

Model Name	Steps	Flexibility	Risk Management	Customer Management	When to Use,
Waterfall	Requirement Analysis > System Design > Implementation > Testing > Deployment > Maintenance > Repeat	Low	High	Low	Requirements are very well known, product definition is stable, technology is understood, new version of an existing product, porting an existing product to a new platform.
Incremental SDLC (Software Development Life Cycle)	Incremented versions of full software development (Design > Code > Test > Deploy)	Low	Mid (only manageable after each increment is finished)	Mid (only manageable after each increment is finished)	Staffing is not available by deadline, software can be broken down into increments and each represents a solution. Need to get basic functionality to the market early, lengthy development schedules, on projects with new technology.
RAD (Rapid Application Development)	Incremented prototypes with each containing	Mid (Project is time-box	High (System are modularized minimizing	High (User involved throughout life cycle)	Reasonably well-known requirements, user involvement

	steps (Business Modeling > Data Modeling > Process Modeling > Application Generation > Testing and Turnover) that after finished, would be carried over by an Automation Tool to deliver the Application Code. Involves incremental amounts of teams with each being in charge of their own modeling and construction before deployment.	ed but requires further team communication for input)	risks via easier quality controls per module)		throughout the life cycle, projects can be time-boxed, functionality delivered in increments, high performance not a requirement, low technical risks, system can be modularized.
Agile	Conceptualization > Inception > Construction > Release > Production and Maintenance	High (teams adapt and evolve requirements based on	High (regular reviews and feedbacks)	High (customers are allowed to see progress and deliver feedback via delivered working	Shorter development schedules, customer involvement, cooperative teams, integrated working

	> Retirement	market condition s)		softwares)	environment with a proper managerial scrum master.