The various System Development Life Cycle which we are studying have a variety of aspects in which they excel at and some aspects in which they struggle with. The waterfall model is a incremental model as each phase provides provides documentation of what the phase achieved and documentation to review the previous work to ensure that the work is correct along with what work is required for the next phase. This incremental process doesn’t give the client a clear end goal of what the product will look like until the product is well under development which can cause high amounts of risks within the project as the user requirements for the project have to be well defined prior to the work commencing on the product, as a major change in the requirements could cause a major increase in time and cost or may even shut the project down. Another SDLC model which is similar to Waterfall is the Incremental Model, both Waterfall and Incremental work in phases but Incremental is less expensive and less detrimental if a client requests a change in a requirement as Incremental works on groups of functions and features at a time instead of working on the whole project at once like in the Waterfall Model. As Incremental Model divides the project in various different builds where each build goes through the design, implementation and testing phases incrementally until the product is completed, it allows for a working version of software for the product to be provided earlier on in product development, compared to The Waterfall Model this involves the client more and respond to the results that the team provided to them, as the client can see a working product early on, it can reduce the risk of changes in the requirements as the client can get a clearer image of the end goal for the product. Due to the incremental model’s flexibility versus the Waterfall Model the cost of changing the requirements is less than that of the Waterfall Model, but the overall cost of Waterfall is cheaper than that of the incremental model.

The V-Shaped Model which stands for Verification and Validation. The V-Shaped is similar to the previous two SDLC models in which it works as a sequential model in which each phase must be completed before another phase can commence. Where the V-Shaped Model differs from The Waterfall Model and The Incremental Model is that for each phase that is completed a testing phase is planned in parallel with that phase. As the product is tested thoroughly throughout, it leaves room for less errors and if an error or a problem in the system it can be found and dealt with earlier than in the Waterfall Model as this model has a downward flow, the defects could be carried throughout the project until the product is near completion in which the testing phase may begin. As the defects can be found early on in the V-Shaped model it could reduce the cost of the project drastically and increase the chances of success of the project as the defects can be resolved earlier than that of the Waterfall Model. Where the Incremental Model differs from The V-Shaped Model in which the coding doesn’t commence until late in the project development there is no opportunity for a prototype to be produced. The main problem involved with the V-Shaped model is that like the Waterfall Model, if a change is required to be made then all the documentation involved in the project is required to be changed for the project to continue, as testing within the V-Shaped Model occurs throughout the project there is less of a risk involved when it comes to an error or change occurring in the project. Like the Waterfall Model and the Incremental Model, the V-Shaped Model is best suited to be used on projects which are smaller and have clearly defined requirements and goals which must be achieved upon completion of the project.

Unlike the Waterfall Model and the V-Shaped Model, the Prototype Model concentrates more on the development of working software versions of the product instead of focusing on documenting each phase, this could allow the product to be released earlier than that of the other models discussed. As there is more focus on the working prototypes of the project, the involvement of the client is required more than that of the Waterfall Model and the V-Shaped Model. As the client has more involvement in the project they can provide clearer and complete requirements and feedback it can reduce the chance of requirements changing as the client can get a clear view of what their product will look like. As the client can get a glimpse of what the product may look like upon completion early on, through the creation of prototypes, the users can provide clearer goals for which they want to product to look like early on in the project. The prototype model attempts to counteract the disadvantages of the Waterfall Model by creating prototypes instead of creating large volumes of documentation. Due to less documentation being required in the Prototyping model in comparison to the previous SDLC models discussed, the prototyping model attempted to accommodate the shortcomings of the client in which they may be unclear of certain requirements involved within the project.

***1. Critically compare and contrast the various Systems Development Life Cycle models (Waterfall, Incremental, V-shape, Spiral and Prototyping).***

***2. Give realistic examples of where you would use each model type.***