**Software Development Lifecycles (Advocate: Thiago Viana)**

**Describe two iterative and two sequential software lifecycle models.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  - Waterfall  Waterfall development has specific goals for each phase of development. Each phase has to be completed before moving on to the next. At the end of each phase, a review is taken place to determine if the project is on the right path and whether we should continue or discard the whole thing. There are 6 phases of the Waterfall model and they are requirements, which are where the potential requirements of the application are methodically analysed and written down in a specification document that serves as the basis for all future development. Secondly, is, Analysis and during this phase the system is reviewed/analysed so we can properly generate the business logic and models that will be used later in the actual application. Next is the design phase this stage covers technical design requirements such as data layers, services and programming languages. Coding is next where all of the core coding is written implementing all models, business logic and service integrations that were specified in the prior stages. The next phase is the testing phase where beta testers and all other testers systematically discover and report any issues they find with the application. Lastly, is the Operations phase, the operations stage entails not just the deployment of the application, but also subsequent support and maintenance that may be required to keep it functional and up-to-date. Some advantages of Waterfall are that it is very simple and easy to understand and use, phases do not overlap as each one is completed before moving onto the next, it is one of the best models to use for smaller projects where requirements are very well understood and finally it is easy to manage because of the rigidity of the model. A few disadvantages of the Waterfall model is it is not good for projects that are complex and object-oriented projects, it is a model that contains high amounts of risk and uncertainty, one the project/application is in testing stage it is very hard to go back and change something that was not well-thought out in the concept stage, it is not a good model for projects that are long and ongoing and lastly, there is no working software produced until one of the last phases within the life cycle.  - Prototyping  The main idea of a Prototype model is to create a throwaway prototype to understand the requirements of the project. The prototype is developing based on the requirements we know. By using this model, the developers can get a ‘feel’ of the system; since the interactions with prototype can enable the client to better understand the requirements of the desired system. The advantages of Prototyping are users remain actively involved in the development process, it has quicker user feedbacks than most models leading to better solutions and lastly, confusing or difficult functions can be easily identified. The disadvantages of Prototyping are: practically this methodology might increase the difficulty of the system; as the scope of the system may expand beyond original plans, incomplete or inadequate problem analysis and an incomplete application may cause the application no to be used as the fully system was designed. Some good examples of when you should use the Prototype model is when the desired system needs to have a lot of interaction with the ends users. Additionally, online system and web interfaces have a very high amount of interaction with end users and these are best suited for Prototype model. Lastly, Prototyping makes sure that the end users constantly work with the system to provide feedback, which is incorporated in the prototype to result in a usable system.  - Spiral  The Spiral model emphasises the importance of risk analysis. There are four main phases of The Spiral model, which include: the planning phase things like Business Requirement Specifications and System Requirement specifications are gathered to make sure you know the ins and outs and what needs to be done/happen within a project. The risk analysis stage, which for the Spiral model is one of the most important phases. Within this phase, a process is done to identify certain risks you may occur during a project and alternate solutions you can take to avoid these risks or make them easier to deal with. Towards the end of the Risk Analysis phase a prototype will then be produced to help further with finding any bugs, problems and to see how the prototype looks/feels. Thirdly, we have the engineering phase, which is where software is developed, and a lot of testing is done within the software to make sure it properly works. Finally, is the evaluation phase where the customer will evaluate the output of the project to date before the project continues to the next stage. There are a few advantages of using the Spiral Model; one being the high amount of risk analysis pays off greatly when it comes to the performance and the quality of the project, it is good for large and mission-critical projects, additional functionality can be added later in time, there is a strong approval and documentation control and lastly software is produced early in the software life cycle. With advantages come disadvantages, these being: it can be a very costly model to use, the high amount of risk analysis involved requires a highly specific expertise, not a good model for smaller projects and lastly the project’s success is highly dependent on how the risk analysis phase goes. A few examples of when you should use the Spiral Model, would be when cost and risk evaluation is important, for medium to high risk projects, when the user is unsure of the needs they want, when they requirements are complex and finally for a new product line.  - Rapid Application Development (RAD)  Rapid application development (RAD) heavily emphasizes rapid prototyping and iterative delivery; because of this, it is good alternative to the Waterfall development model. There are many advantages of RAD but these are just a few crucial ones you should take into consideration. Firstly, it is easy to measure progress as there are so many components, iterations, and prototypes coming down the pipe, and so much information and feedback being given back everything can be maintained and evaluated to make sure you are keeping schedules and budgets. A large percentage of active software developers will switch to the multi-discipline roles, a Rapid application development methodology allows skilled team members to quickly produce prototypes and code that works to illustrate examples that using a slower development technique could take weeks. While most models wait until the end to begin integrations, an application will be integrated instantly. While developing software is a fairly malleable form, since code can be altered that can change the entire system. It is an advantage to the developers to use this flexibility early on and use it a lot; by iterating concepts or ideas while developing. On the other hand, there are a few disadvantages of RAD, one being that even though RAD gives the developers a lot of flexibility throughout the product's lifecycle; it reduces the control and restrictions, which in reality is not all bad, but on large projects, it can be difficult. Finally, when using RAD the team will need to consist of a very skilled group of individuals that can adapt to things within the project evolving.  - Traditional X Agile  The Traditional x Agile model is based upon adaptability of changing product requirements and also enhancing customer satisfaction by rapidly delivering working product features and evolving the client. One of its methods is to break up the entire product into easily developable features, developed through a cycle knows as a sprint. Lastly, there are a few frameworks within the Agile model, but they all boil down to the same idea of breaking down tasks, and responsibilities into smaller tranches. The advantages of the Agile model are the project is continuously improving and so are the team. The project can evolve a lot within this model so the team will adapt and overcome the obstacles they are faced with, improving the project. Additionally, if you the team you are dealing with is experienced change can be catered very well for even though Traditional X Agile has a short planning cycle. A few disadvantages of the Traditional X Agile model is that the planning of a target date consisting of x can be sometimes dubious and not secure. The team will need to consist of highly skilled members on the Agile framework chosen and cross skilled in competencies, as the team can be very small. Lastly, time and effort are continually required from the products resources, this is essential to the cycle planning and success.  - Formal / Light Formal  The Formal Methods are more of the complex models to use as they use rigorously specified mathematical models to build its software and hardware systems. Additionally, to ensure correct behaviour of the system, Formal methods will use mathematical proof as a complement to system testing and as safety becomes a more important issue and as systems become a lot more complicated, the Formal approach offers another level of insurance. A few advantages of Formal methods are, because of their rigour they require an engineer to think about his design in a more thorough way; this thorough approach can help identify faulty reasoning far sooner than other traditional designs. Another advantage is the clarity that comes with Formal methods. Formal methods require very clear and defined goals, objectives and approaches, which gives the team a very good idea of what they want to achieve and how they want to go about doing it. For example when in a safety critical system, ambiguity can be a huge threat to the system as a whole, which is why using a Formal method is better as it eliminates ambiguity. A disadvantage of a Formal approach would be the cost. As there is far more rigour in Formal methods it c makes the approach a lot more expensive. Formal methods involve a large initial cost followed by less consumption as the project advances; this is the complete opposite of traditional models. Finally, most formal methods introduce some form of computational model, which restricts the efficiency of the operations allowed to make the notation elegant and the system provable or work well. These design limitations are usually considered unacceptable from a developer's perspective as it restricts their creativity, and skills. |

**Explain how risk is managed in the Spiral lifecycle model.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  The Spiral model is based a lot upon risk analysis and making sure every risk is taken into account when moving forward. The progress of the project depends a lot on the risk analysis phase and all ideas are risk analysed to see if it is beneficial to the project's progress. A process is undertaken to identify risks and other solutions that can be used to prevent/lower the chance of failure and risks. A prototype produced at the end of the risk analysis phase to help see what the finished product would look like after all of the points from the risk analysis have been taken into account. If a risk or problem is found during the risk analysis there will be a brainstorm of solutions and ideas that could either help reduce this risk or completely eliminate it. |

**Explain the purpose of a feasibility report.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  A feasibility report is an analysis of the viability of any idea; it is taken very seriously and is a huge part of any new project or existing project. It is a process of thinking of the logical start and the logical end of the idea in immense detail. A feasibility report acts as security measure as it looks over every idea to make sure it is not going to negatively affect your project, or going to waste your time starting a project that won’t become successful or work to benefit you. The purpose of a feasibility report is to know the different variables involved with taking your project or business venture onto the business market, whom the target audience will consist of and if they will accept it. |

**Describe how technical solutions can be compared.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 4 in my sessions:  **Simple report,** research and describe how technical solutions can be compared.  Also, you can use your ZSL project and explain how did you choose your IDEs, the programming language, and other stuff that you have used in this project. How to compare those options? |

**Undertake a software investigation to meet a business need.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  You did this in the ZSL project. You need to describe how and when you met with the client. How you wrote the requirements, how and why you changed your project specifications and add a link to your ZSL repo. |

**Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  You did this in the ZSL project. You need to describe how and when you met with the client. How you wrote the requirements, how and why you changed your project specifications and add a link to your ZSL repo.  In this project you probably used these tools: Text Editor, Presentation Editor (slides), Image Editor (to the assets and prototypes in your project), UNIT (and others IDEs), other tools.  Also, in this project you probably used these techniques: Interviews with the client, Prototyping (for your APP). You should write and add links to your ZSL documentation. |

**Explain how user and software requirements have been addressed.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  In each project you are addressing the requirements using EPICS, USER STORIES and creating documentation to your projects. You should state this here and add the links to all your projects documentation. |

**Describe, with an example, why a particular lifecycle model is selected for a development environment.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 1 in my sessions:  **Week 1:**   * **Simple report** - Research each of the lifecycles given:   1. - Waterfall   2. - Evolutionary   3. - Prototyping   4. - Spiral   5. - Rapid Application Development (RAD)   6. - Traditional X Agile   7. - Formal / Light Formal   Note how each model works, their advantages and disadvantages and give some examples of software that could benefit from each specific lifecycle. |

**Discuss the components of a feasibility report.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 2 in my sessions:  Explain the purpose of a feasibility report. |

**Analyse how software requirements can be traced throughout the software lifecycle.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  TO DO (you can leave it blank now, we are going to address this un future sessions) |

**Discuss two approaches to improving software quality.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 5 in my sessions:  **Simple report,** research and discuss two approaches to improving software quality.  You should research the CMM quality model and explain 2 approaches that they suggest to develop mature softwares. |

**Suggest two software behavioural specification methods and illustrate their use with an example.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  TO DO (you can leave it blank now, we are going to address this un future sessions) |

**Differentiate between a finite state machine (FSM) and an extended- FSM, providing an application for both.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  TO DO (you can leave it blank now, we are going to address this un future sessions) |

**Assess the merits of applying the Waterfall lifecycle model to a large software development project.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 1 in my sessions:  **Week 1:**   * **Simple report** - Research each of the lifecycles given:   1. - Waterfall   2. - Evolutionary   3. - Prototyping   4. - Spiral   5. - Rapid Application Development (RAD)   6. - Traditional X Agile   7. - Formal / Light Formal   Note how each model works, their advantages and disadvantages and give some examples of software that could benefit from each specific lifecycle. |

**Assess the impact of different feasibility criteria on a software investigation.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  Link to your TASK 2 in my sessions:   * - Assess the impact of different feasibility criteria on a software investigation. |

**Critically evaluate how the use of the function design paradigm in the software development lifecycle can improve software quality.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  TO DO (you can leave it blank now, we are going to address this un future sessions) |

**Present justifications of how data driven software can improve the reliability and effectiveness of software.**

|  |
| --- |
| Please use this section to provide all appropriate, valid and checked http Links that point to your evidence; use multiple lines to separate multiple links |
| Please provide a short (between 3 to 8 well considered, fully proofread and reflected sentences) explanation that justifies why the evidence/links you have provided is suitable as evidence of this requirement  TO DO (you can leave it blank now, we are going to address this un future sessions) |