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

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

|  |
| --- |
| <https://github.com/LBruni98/Project-Lifecycles> |
| The link above takes the user to a repo regarding project lifecycles. In this repo contains a report of many software lifecycles. This is sufficient evidence because at least two of both iterative and sequential lifecycles are described in this document. |

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

|  |
| --- |
| <https://github.com/LBruni98/How-Risk-is-Managed-in-Spiral> |
| The link above directs the user to a lone repo, containing a simple report. This report talks about the spiral model, in depth. In this description, it is also explained how risk is managed in this lifecycle, how it helps with identifying risk earlier and what sort of project fits this development lifecycle |

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

|  |
| --- |
| <https://github.com/LBruni98/Feasibility-Reports#purpose> |
| Above is the link to a repo, containing a report on feasibility reports. This document describes what and explains their purpose. The report goes in depth with feasibility reports, explaining the purpose, how it is used and also goes a step further by detailing each of the components that go into a feasibility report. |

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

|  |
| --- |
| <https://github.com/LBruni98/High-Low-Card-Game#ide> |
| The link above leads the user to a micro project repo, the high low card game. In this document, underneath IDE, it describes two IDE used; repl.it, an online IDE used to construct the program for this project, and Notepad, which was used in a project before this. The section in the document, compares the two IDEs but also goes into detail in the features of common IDEs and their working environment, any factor that would be used when comparing technical solutions. |

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

|  |
| --- |
| <https://github.com/LBruni98/ZSL-The-Climate-Menace#specification> |
| The link above leads to the ZSL project repo, specifically the README document. The information is located underneath the “Specification” section and the project backlogs. These parts of the document outline the client’s requirements, what the client is exactly looking for and the backlogs detailing the development and working towards the requirements in regards to the software used and experience of the team. |

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

|  |
| --- |
| <https://github.com/LBruni98/ZSL-The-Climate-Menace#concept>  <https://github.com/LBruni98/ZSL-The-Climate-Menace/blob/master/Game%20HUD%20Mockup.png>  <https://github.com/LBruni98/ZSL-The-Climate-Menace/blob/master/Mockup%20App%20Menu.png>  <https://github.com/LBruni98/ZSL-The-Climate-Menace/blob/master/ZSL%20Concept%20Art.docx?raw=true> |
| The link above directs the user to the ZSL project repo. The evidence contained are in the README document and other files that can be accessed from the top of the repo and include concept art. These techniques were used to create the app entirely. The README document also contains details of the IDE used to create the app and goes in depth about how and why it was used. |

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

|  |
| --- |
| <https://github.com/LBruni98/Trace-Ball#development>  <https://github.com/LBruni98/High-Low-Card-Game#development>  <https://github.com/LBruni98/Anagram-Solver#development>  <https://github.com/LBruni98/ZSL-The-Climate-Menace#specification> |
| The links above are to mini projects and the last one directs to the ZSL project, all leading to the readme documents (automatically displayed when opened). The projects define the planning and the components chosen to create the program. The aims of each project were made by breaking down the project and creating user stories, treating them as individual tasks which would be used to help with the development of the project. |

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

|  |
| --- |
| <https://github.com/LBruni98/Project-Lifecycles> |
| Above is the link to a repo on Project Lifecycles, covering past criteria mentioned earlier. It contains several methodologies and details about them. Also, worth noting is that it also covers the benefits and drawbacks from them as well as examples; which company or project would best suit this type of methodology. |

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

|  |
| --- |
| <https://github.com/LBruni98/Feasibility-Reports#components> |
| Above is the link to a repo containing a simple report on Feasibility reports. The report is created using a README document and within contain what feasibility reports are and their impact. Listed inside also are the main components of a feasibility report, with explanation on what they are to make up the report. |

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

|  |
| --- |
| To be completed |
|  |

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

|  |
| --- |
| <https://github.com/LBruni98/Improving-Software-Quality#approaches-towards-improving-software-quality> |
| Above is the link to a repo on the subject of improving software quality. It explains what CMM is and refers to two methods on how to improve upon software quality, with how they can help upon the quality of the software being developed. |

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

|  |
| --- |
| To be completed |
|  |

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

|  |
| --- |
| To be completed |
|  |

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

|  |
| --- |
| <https://github.com/LBruni98/Project-Lifecycles#waterfall> |
| Above is the link to a repo on Project Lifecycles, covering past criteria mentioned earlier. It contains several methodologies and details about them. The waterfall methodology is listed and the README document explains the advantages and disadvantages to using the methodology in the workplace. |

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

|  |
| --- |
| <https://github.com/LBruni98/Feasibility-Reports#impact-of-feasibility> |
| The link above leads to a repo containing a simple report on Feasibility reports. The importance and components are listed in the README document upon opening the repo. Different feasibility criteria are covered, including details on the impact of each. These being technical, economic and organisational feasibility. |

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

|  |
| --- |
| To be completed |
|  |

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

|  |
| --- |
| To be completed |
|  |