To develop our ATM system, we chose to follow the evolving prototype approach. The key advantages for us which helped us decide to use the evolving prototype approach is that it allows for a quick mock-up of the software system, it does not require rigid specifications and requirements and these can evolve over time, and it is a quick iterative process which tightly integrates developer and client feedback. Using the evolving prototype approach, we are required to develop a textual ATM system. Due to the foreign combination of a computer terminal and an ATM machine, the requirements of all the features we wanted to implement and how we wanted to present the final product were unknown and evolved over time as we continually re-evaluated our system. Because of this, the evolving prototype approach is suitable as it does not require rigid specifications and allows for evolving requirements over time, as we take in feedback from each other and our teacher. Furthermore, using the evolving prototype approach allows us to develop a quick mock-up of the ATM system. Anthony has developed previous projects which implement textual terminal GUIs, and he can use these to quickly create a prototype for an initial ATM system which we can use as a starting point to visualise and iterate over the features and design of the ATM. As well as this, the timeframe of the project is relatively short, being only 3 weeks, and thus the development approach chosen needs to allow for quick iterations, where we gain feedback from each other and our teacher to ensure we are on the right track, and needs to be suitable for small scale projects. Due to this the evolving prototype is the perfect approach as it allows for quick iterations and is suitable for a short time scale. Adding to this, we didn’t use the structured or agile approach as we evaluated them to be too large scale for this type of project. We did not have enough resources in terms of time to effectively utilise the structured approach, and the agile approach would be ineffective as there isn’t enough time to plan out, test and iterate through each cycle multiple times, which can be done more efficiently with the evolving prototype approach. We chose not to use the RAD (rapid application development) approach as a key feature of RAD is the use of little to no code, and therefore as we plan to implement all of the features in code ourselves, this approach is not suitable. We also cannot use the End User Approach as the project is not internal and the ATM system needs to be robust and secure. Finally, the evolving prototype approach is suitable to our own personal development preferences. When developing small-scale projects like the ATM system, we prefer to implement the code by quickly prototyping a feature and testing it out to see if it is suitable to the design and requirements of the system. Then, we like to ask for feedback from peers and mentors which we can use to refine the system being developed, which we can then implement in the next prototype cycle. Thus, the evolving prototype approach is the perfect software development approach for the development of the ATM system.