

Analysis of the stages in the development of a current information system using a...

[Business](#), [Strategy](#)



Overview: In this task I will use a traditional SDLC to outline the stages in the development of a current information system. I will be using the “ Waterfall” SLDC method to outline each of these stages. Assignment: Waterfall is an easy to understand SDLC method but is considered outdated because of its obtuse nature, meaning that none of the stages overlap with one another and you can only move onto the next stage when the previous stage has been completed as shown in the diagram below.

Stage 1: Planning and requirement analysis: The objective for this phase is to analyse what needs to be in the software. Alternative methods for achieving these objectives are also put forward. Costs and benefits are recognised and a preliminary plan is put forward. Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department and market research. In waterfall this would mean coming up with a general overview of what the product needs to function properly, e. g. For a video game the developers would need to define things like the mechanics, pickups, enemies, characters, health, plot and levels. It isnt necessary to define how these work in this stage but instead it is imperative to simply define what is needed. This would all need to be defined before they can move on to the second stage of waterfall.

Stage 2: Defining requirements: This step entails the process of defining project goals into defined functions and how the application will use them. This involves gathering facts, diagnosing problems and suggesting improvements. This stage also analyses user information needs and removes

any inconsistencies and incompleteness in these requirements. For waterfall, this would mean planning out how all of the mechanics and code in a game would work together, but also what parts wouldn't work together, why they wouldn't work together and what can be implemented to fix these inconsistencies.

Stage 3: Designing the product architecture: This stage describes the features to be implemented in great detail, including UI layout, UML diagrams and other documentation. UML diagrams are a great help for any SDLC method including waterfall, allowing developers to effectively show what needs to be done and the general path to accomplishing it. UML diagrams in a video game would depict things like an item shop for the character to purchase things at, which I have created and included below.

Stage4: Building/ developing the product: It is in this stage that actual development of code for the application begins and is carried out. This is the stage where all the previously made plans are put into production and actually built and/or coded. For the video game example I am using this is where the developers, artists and coders make all the assets required for the game and built them into a functional game in whatever game engine they choose such as Unity etc. Depending on the product being produced this stage can be either short or extremely long with better planning in the previous steps resulting in overall faster production times as opposed to a poorly thought out plan where the developers have to keep going back and re-writing it.

Stage 5: Product testing: All the pieces are brought together and put through rigorous testing in order to identify any problems or bugs that may have gone unnoticed thus far.

Stage 6: Market deployment and maintenance: Once the product has passed through the testing stage it can be deemed fit for market consumption and is released to the public. However, few pieces of modern software are released completely bug free and it is often necessary to release maintenance files containing updated code called “ patches” to remedy any bugs that have escaped being eradicated in the testing stage.