

Design Document

Unit 4: Assignment 2

Contents

User Requirements	2
Software Development Lifecycle	3
Design of the System	4

User Requirements

The computer program that I've been asked to design and produce will have to be made in a simple way so kids can understand it better. Since the target audience is for 6-9 year olds, I will need to make an educational program which focuses on multiplications.

How this will work is that it will generate a series of 10 random multiplication questions that will use the numbers from 1 to 10, while the answers must be integers between from 1 to 100. A child that will be using the program, will receive a question that's generated, so then they would need to input their answer to the generated question. The program will then check for an error in that answer, if the answer is wrong, the program will output that it's wrong (which would be an error), and if the answer is right, the program will output that it's right. They will need to do this 10 times as the program gives them questions that's generated 10 times. After these 10 generated questions, the program will give them their mark out of 10 by checking which ones they've got right and which ones they've got wrong, so say if the child that's using the program got 6 questions right and 4 questions wrong, the program would output "Well done you've got 6 questions out of 10 correct", then the program would need to have a supportive message coded that will help the child know how to improve their performance and knowledge.

Software Development Lifecycle

The software development lifecycle (which in short is SDLC) is the process that effectively costs less time and money development teams use in order to build and design software with high quality. The goal of this process is to minimise project risks by planning forwardly so that the software meets what customer is expecting during their production and beyond. However, they outline several tasks that are required to build a software application as it works by going through several stages as developers will be adding new features and fix bugs in their software.

- Plan – this involves tasks such as analysing, scheduling, estimating and allocating resources which will all be done in a way that will benefit costs. The development team collects requirements from customers, internal and external experts, and managers to create a software requirement specification document.
- Design – software engineers analyse requirements and identify best solutions to creating software. For example, they may consider integrating modules that already exist, making technological choices, and identifying developmental tools. They'll research the best way on how to integrate new software into existing IT infrastructure the organisation may have.
- Implement – the development team codes the program. They analyse requirements that's there to identify smaller coding tasks they can do daily to achieve the final result.
- Test – they combine automation and manual testing to check the software for bugs. Quality analysis includes testing software for errors and checking if it meets customer requirements. Because many development teams test the code they write immediately, the testing phase often runs parallel to the development phase.
- Deploy – when teams develop software, they code and test on a different copy of the software than the one that users have access to. The software customers use is called production, while other copies are in the build or testing environment. Having separate build and production environments ensures that customers can continue to use the software even while it's being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environmental configuration and installation.

- Maintain – the development team fixes bugs, resolves customer issues and manages software changes. Additionally, they monitor overall system performance, security and user experience to identify new ways to improve existing software.

Design of the System