Software Development Life Cycle (SDLC)

Alston

1 Introduction

Notes on SDLC for the 9618 CAIE Computer Science course.

2 Basics

The five most crucial steps are: **Analysis, Design, Implementation, Evaluation, Maintenance**.

2.1 Analysis

Define the problem. Do the research and collect data.

2.2 Design

Outline the program process: structure charts, what each module does, what parameters are being passed. Decide what hardware is required. Also outline the UI and UX

2.3 Implementation/programming

Software developer writes and debugs each module

2.4 Implementation/testing

Testing Type	**Description**
White-box Testing	Tests the internal structure and logic of the program; requires knowledge of the source code.
Black-box Testing	Tests the program's functionality without knowing the internal code; focuses on inputs and expected outputs.
Unit Testing	Tests individual modules or components of the system in isolation.
Integration Testing	Tests how different modules work together and communicate correctly.
Alpha Testing	Conducted by the developers or in-house testers before release, to identify and fix major issues.
Beta Testing	Conducted by a limited group of end users in a real environment before final release.
Acceptance Testing	Determines whether the system meets the agreed requirements and is ready for deployment; often done by the client.

2.5 Evaluation

Normally carried out 3-6 months after the software is online. End user feedback is collected about UI and UX and functionality.

2.6 Maintenance

Software upgrades and bug fixes. Corrective, adaptive, perfective maintenance. **Corrective**: fixes bugs that were not found in previous stages of testing. **Adaptive**: development of changes that were necessary for the system. Program is changed to do something it's not orginally designed to do. **Perfective**: Quality of life improvements

3 Waterfall Life Cycle Model

Each step follows from each other linearly. Not a flexible model. It's good for the manufacturing industry, cuz changes after the implementation stage will icnrease the cost, so we must take great care before moving on.

4 Spiral and Agile

Spiral model just runs the fives stage process over and over again for many times. After each run, a prototype might be generated and improved upon. Best for long term projects.

Agile model is non-linear, so the developer makes some changes and gets feedback from the users. Best for short term projects.