CSE 1310 - Introduction to Computers & Programming

Software Development Life Cycle

Alex Dillhoff

University of Texas at Arlington

Introduction

Software Development Life Cycle (SDLC) is a general framework for designing, developing, testing, and maintaining software.

It is a useful tool in both academia and industry for managing development of high quality projects subject to deadlines and cost constraints.

Introduction

Because SDLC is a general framework, there are specific variations that implement key concepts from the general overview.

- Waterfall Model
- Iterative Model
- Spiral Model
- Agile Model
- etc.

SDLC Stages

1. Planning and Requirement Analysis
2. Defining Requirements
3. Design
4. Development
5. Integration and Testing
6. Deployment
7. Maintenance

Planning and Requirements Analysis

During the **Planning and Requirements Analysis** stage, a preliminary overview and plan of the project is created.

This is arguably the most import part of SDLC as the major objectives and requirements are established.

Planning and Requirements Analysis

Common tasks during this stage are:

- Collect requirements from customers
- Identify potential risks (budget, timeline, etc.)
- Feasibility study (are the requirements possible?)
- Cost benefit analysis (can we re-use existing components?)

Defining Requirements

In the **Defining Requirements** stage, the high level requirements collected in the previous stage are translated into specific technical requirements.

It is important to maintain communication with all stakeholders during this stage to ensure that the overall goals of the project are fulfilled even with the possibility of technical limitations.

Defining Requirements

The primary focus of this stage is to define the exact technical requirements. Other tasks include:

- Translate general requirements into exact requirements
- If existing components exist, determine integration strategies
- Review technical requirements and identify shortcomings
- Create **Software Requirement Specification** document

Design

At this stage, the product is designed in great detail, including UI mockups use-case scenarios, and process diagrams.

The final design should be cleared with all parties involved in the project before moving forward.

Design

Common tasks for the **Design** stage are:

- Create the **Design Document Specification** document
- Design based on the requirements from the previous stage
- Sometimes propose multiple design alternatives
- Double check design with clients
- Double check design against time and cost constraints

Development

This stage involves actual coding and development of the system.

The development team takes the exact requirements and translates them into code.

Integration and Testing

In this phase, testing is done of the currently developed system to determine if it is successfully meeting all requirements without bugs.

The testing involved at this point goes beyond unit testing. The major components and use-case scenarios are evaluated.

Deployment

An initial working version of the system is deployed during this stage.

Any unresolved bugs or deficiencies that were unaccounted for will become apparent.

Maintenance

Unintended use-cases and unforeseen circumstances will surely appear during the life of the system.

Some amount of effort will be needed to maintain the overall functionality of the system.

Maintenance

Additionally, the requirements may change over time.

Along with adding new features, some existing components may be modified or removed.

SDLC

Specific implementations will be studied in later classes, but let's attempt to solve problems using this general framework.

Sales System

We are tasked with creating a system that manages sales of general goods for an online vendor.

The first step is to collect the general requirements of the system.

Planning

The customer has given some general requirements. The system should...

- Manage a product inventory
- Track sales
- Provide summary statistics

Define Requirements

Based on the general requirements, the planning team has come up with a few core features:

- Store product inventory (ID, price, name, amount)
- List all products (with filter options)
  - All available
  - By price threshold
  - Out of stock items
- Log a sale
- View all sales (with filters)
  - By price threshold
  - By item ID

Define Requirements

The data will be processed as Comma Separated Values. Eventually, we will process all data through files.

For now, we should design and implement our functions as though each item is processed as CSV.

Inventory Item

For an inventory item we have the following properties:

- Product ID
- Name
- Price
- Current stock

Example Item

```
"1,Intel Processor,399.99,10"
```

Sales Log

For tracking sales, we need the following information.

- Product ID
- Amount sold
- Customer ID
- Seller ID

Example Item `"1,2,10000,20000"`

Design

The design team has decided that the UI will be implemented ASCII since the customer will be running the system on very outdated hardware.

Development

We will begin development on some core functionality.

The first task will be to add items and write them to a file.

Integration, Testing, etc.

We will discuss the rest of the steps of our project as we get to them.

For now, let's start developing...