

Software Engineering 101: Exercises

Bernardo Paulsen

March 16, 2021

1 Introduction

1.1 Welcome to the Course!

2 Software Lifecycle

2.1 Why Learn Software Engineering/Development?

Investment.

2.2 Typical Software Development Lifecycle

1. Requirements
 - What is going to be built?
2. Design
 - How is it going to be built?
3. Implementation
 - Building.
4. Verification
 - Testing.
5. Maintenance
 - Fixing.

2.3 Software Development Lifecycle Example

1. Requirements
 - Collect email address and message.
 - Send to and store in a database.

- Prevent user from bad input.
2. Design
 - Use HTML and CSS for building the framework of the form.
 - Use Javascript for verification of the input.
 - Use JQuery and MYSQL for contacting backend.
 3. Implementation
 - Conde and document the work.
 4. Verification
 - Does the form collect information?
 - Does the form send that information to a database?
 - Does the form prevent bad user input?
 5. Maintence
 - Create lifecycle plan, fix any bugs.

3 Requirements and Specifications

3.1 Requirements Definition

A way of figuring out the exact specifications of what the software should do.

3.2 Requirements vs Specifications

- Requirements
 - A non-technical definition of something the user requires from the system.
- Specifications
 - A technical definition of what is required from the system.

3.3 Functional vs Non-Functional Requirements

- Functional
 - Requirements and specifications pertaining to the function of the program.
- Non-functional
 - Requirements and specifications on what goals should be met.

- Non-functional
 - Product Requirements
 - * Must have of the product itself. Talk about behavior.
 - Organizational Requirements
 - * Company policies, standards, styles, rules.
 - External Requirements
 - * External laws, regulations, trends etc.

3.4 WRSPM Model Introduction

- World
 - The world assumptions which are used to develop the system.
- Requirements
 - Defining the problem at hand in terms of a user of the system.
- Specifications
 - Defining the technical requirements of the system. Here we are linking together the idea of the solution, to the system itself.
- program
 - The program and code itself.
- Machine
 - Hardware specifications.

3.5 WRSPM Visual Model

- Environment
 - World
 - Requirements
- Interface
 - Specifications
- system
 - Program
 - Machine

3.6 WRSPM Variables

- Environment
 - Eh
 - * Elements of the environment which are hidden from the system.
- Interface
 - Ev
 - * Elements of the environment which are visible from the system.
 - Sv
 - * Elements of the system which are visible from the environment.
- System
 - Sh
 - * Elements from the system which are hidden from the environment.

3.7 WRSPM - World Example

Example: photo machine.

- World Assumptions
 - There will be readily available electricity.
 - There will be internet.
 - In a normal, controlled, temperature.
 - People will have standardized photo formats to print.
 - People will be able to read the prompts and tutorial.
 - Stores will re-fill the materials frequently.
 - People will have debit cards that work with standard debit card tech.
 - Debit/credit cards are widely used.

3.8 WRSPM - Requirements Example

Example: photo machine.

- Requirements
 - A user can connect a smartphone to the system.
 - The user can edit photos.
 - The smartphone can transfer pictures over.
 - The photos can be printed out of the machine.
 - The kiosk can collect money.
 - The user can change the language.

3.9 WRSPM - Specifications Example

Example: photo machine.

- Specifications

—

3.10 Requirements Example