Software Development Process

Michael Nowak

Texas A&M University

Overview

Overview

Overview

Analysis

- Figure out what should be done (requirements/specifications)
    - What are the possible problems that need to be solved?
    - What is your current understanding of those problems?
    - What is the process that you must go through to solve these problems?
    - Are there any edge cases that must be considered / are there any constraints that must be acknowledged?

Overview

Design

- Create an overall structure for the system
  - How does the program flow?
  - Which parts should the system have?
  - How should those parts communicate?
  - Can any libraries help you solve the problem?

Design

- Focused on problem solving, *not* on implementation/coding details
- Documentation of the design should be language agnostic
- In this course, we will capture design details using
  - Flowcharts
  - Pseudocode

Overview

Implementation

- Consists of three stages
  - Writing code
  - Debugging the code that we've written
  - Testing the code to ensure that it actually does what it is supposed to do

Overview

Repeat

- When we start a program, we rarely know the problem(s) well
  - We frequently think that we do, but often we don't
- Only a combination of thinking about the problem (analysis) and experimentation (design and implementation) give us the solid understanding that we need to write good programs

Repeat

- We frequently build a small, limited version of our programs first
  - This helps bring out problems in our understanding, ideas, and tools
  - It also helps us see if details of the problem statement need changing to make the problem managable
  - It is rare to find that we had anticipated everything when we analyzed the problem and made the initial design
  - So we frequently have multiple iterations through the analysis, design, and implementation steps

Overview

References

- Stroustrup, B. (2014). *Programming: principles and practice using C++* (2nd ed.). Pearson Education.