Software Construction Guided Notes

Instructions:

Fill in the missing blanks in the notes as you watch the course videos for this learning module.

Software construction is like building a house. You must:

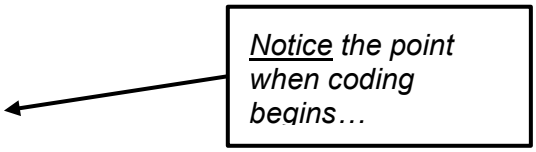
- plan first
- construct in the proper order
- reconstruction is always more expensive

> *"Weeks of coding can save you hours of planning!"*

Software Construction Steps

Software construction involves the following steps:

- Defining the problem
- Identifying requirements
- High-level system design
- Detailed design (of the components)
- Coding and debugging (implementation)
- Testing phase (unit, integration, system, acceptance)
- Deployment
- Maintenance



Purpose of Planning

The bigger the project, the longer the time you need to spend planning.

Finding and correcting errors before construction is 10 to 100 times less expensive than after coding begins.

Prerequisites: Problem Definition

Problem definitions should be	Problem definitions should not be
simply stated written in user language written from the user's point of view	overly complex filled with technical jargon specify a solution

Prerequisites: Finding Requirements

Requirements describe what the software should do, not how the software should do it.

Explicit requirements create a dialogue between you and the client. They also help diffuse any future arguments.

Prerequisites: Architecture

Architecture quality impacts software quality.

Making construction, architecture changes are expensive.

Dose of Reality

Requirements are never completely stable!

You can reduce the amount of requirements changes before construction by:

- making sure you start with good requirements
- having a change-control procedure to help throttle change requests
- using iterative development approaches

You can manage changing requirements during construction by:

- reminding clients how expensive a change will be
- reminding developers (including yourself) of the business case

If all else fails, you may have to consider scrapping the project.

The time spent in planning depends on:

- the needs and size of the project
- your experience with the project's domain and development tools

You should know where you are on the software landscape:

- business system, mission-critical system, or embedded life-critical system?
- sequential or iterative development?
- where on the technology wave?

Summary

Summarize your notes in 2-4 sentences. Identify the key points and the main take-away message.

When starting a project, many software developers are inclined to jump right into coding. However, developers should begin by planning to ensure that code is constructed in the proper order. Unforeseen events may alter your plan, but preparing before coding will save time, efforts, and money.