Learning Objectives

After completing this unit, you should be able to:

- Summarize the four stages of the software development life cycle.
- Explain differences between the waterfall model and agile process.
- Use an integrated debugger (set breakpoints, step through code).
- Describe modularity, coupling, cohesion, and information hiding.
- Interpret structure charts, class diagrams, and sequence diagrams.
- Identify global variables, parameters, and local variables in Python.
- Explain the difference between functions and modules in Python.

Textbook Sections

7.1 The Software Engineering Discipline
7.2 The Software Life Cycle
7.3 Software Engineering Methodologies
7.4 Modularity
7.5 Tools of the Trade

Video Lectures

- CS Field Guide: Software Engineering
- Software Engineering
- UML 2.0 Tutorial

Assignments

Act09 Software Dev Life Cycles; Chapter 7 Problems
Lab09 Codecademy (3 & 4); Static analysis, debugging

Unit 9 Checklist: Oct 28 – Nov 03

Before Wednesday	**Date Completed**
FINISH models 1–3 of Software Development	
READ textbook 7.1 The Software Eng. Discipline (take notes) ANSWER question 4 in your notes	
READ textbook 7.2 The Software Life Cycle (take notes) ANSWER question 2 in your notes	
READ textbook 7.3 Software Eng. Methodologies (take notes) ANSWER question 1 in your notes	
WATCH video lecture: Software Engineering (take notes)	
START Lab09: Static analysis, debugging	(10 pts)
Before Friday	**Date Completed**
READ textbook 7.4 Modularity (take notes) ANSWER question 1 in your notes	
READ textbook 7.5 Tools of the Trade (take notes) ANSWER question 3 in your notes	
DO tutorial: Python Tutor (click “See example code...”)	
DO tutorial: Codecademy (4. Functions)	
START Act09 exercises (complete at least 75%)	(15 pts)
Before Monday	**Date Completed**
COMPARE your Lab09 and Act09 with the solutions in Canvas	
SUBMIT Quiz09 – 1st attempt closed: see what you don’t know	
STUDY your notes, ask questions on Piazza, meet with the TAs	
SUBMIT Quiz09 – 2nd attempt open: try to get the full 10 points	(10 pts)

TAKE Exam09	(40 pts)

Activity 9: Software Development

Software development activities are grouped into four main categories: ***analyze***, ***design***, ***code***, and ***test***. This activity explores ways to organize these categories into a software development life cycle (SDLC).

Model 1 Finding & Fixing Errors

Estimate how long (seconds, minutes, hours, days, weeks, months, or years) it typically takes to correct an error in software when it is found by:

a.	a **compiler**, seconds after the file was edited	seconds
b.	a **compiler**, later the same day or during a nightly build	hours/days
c.	a **pair programming** partner, seconds after the error was made	
d.	a **code review**, days or weeks after the file was edited	
e.	a **customer** or other user, months after the software is released	
f.	a **unit test**, minutes after the file was edited	
g.	a **unit test**, later the same day or during a nightly build	
h.	a **system test**, shortly before software is released (weeks or months after the file was edited)	

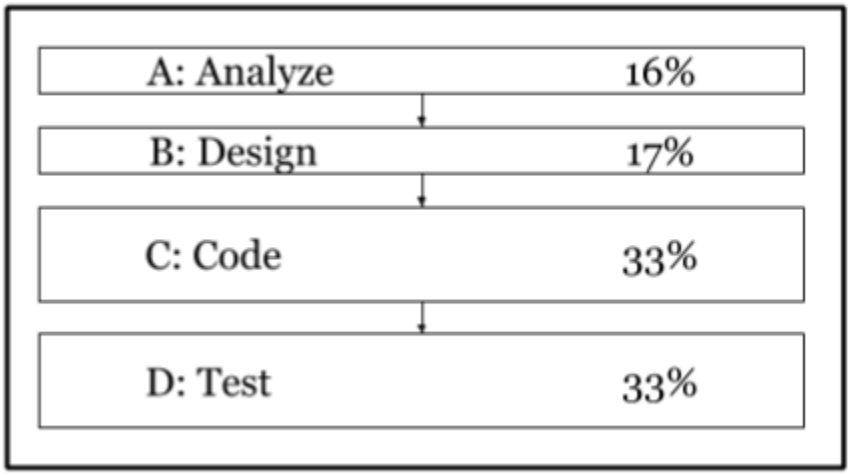
Questions (5 min) **Start time:** _____

1. Describe (or sketch a graph of) the relationship between the time to **find an error** and the time and cost to **repair an error**.

2. Explain why we should use an SDLC that finds and fixes errors as quickly as possible.

Model 2 The Waterfall Model

The following diagram shows the typical percentage of **total cost & effort** for each stage of software development. In practice, these percentages vary widely by project.



Questions (10 min) **Start time:** _____

3. Based on the Waterfall Model:

   a) How many stages are there?

   b) Which stage is 1st?

   c) Which stage(s) must be finished before **coding** starts?

4. Based on the Waterfall Model:

   a) What % of total effort is in the **last stage**?

   b) What % of total effort is in the **first two stages**?

   c) When the project is 25% completed, what % of **analysis** is done?

   d) When the project is 25% completed, what % of **coding** is done?

   e) When the project is 50% completed, what % of **coding** is done?

   f) When the project is 50% completed, what % of **testing** is done?

5. It is important to find and fix errors in software.

a) If **coding** errors are found during **C: Code**,
in which stage should they be fixed?

b) If **coding** errors are found during **D: Test**,
in which stage should they be fixed?

c) If **analysis** errors are found during **B: Design**,
in which stage should they be fixed?

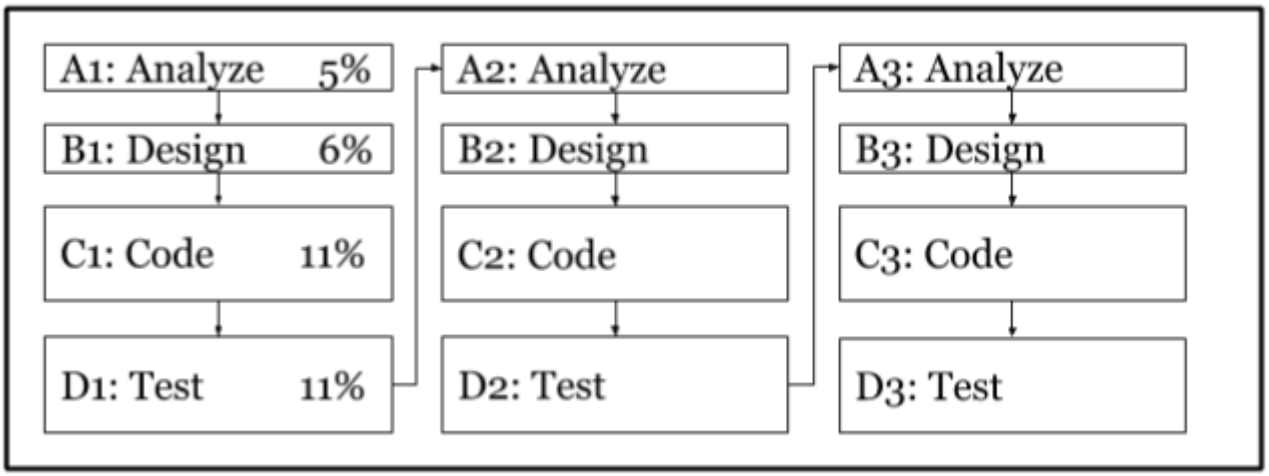
d) If **analysis** errors are found during **D: Test**,
in which stage should they be fixed?

e) Which stage focuses most on **finding** errors?

f) Are major errors in analysis and design more likely
when the project is **similar** to past projects, or **different**?

6. Later stages often take more time, effort, and money than expected. Explain why based on your answers to the previous questions.

Model 3 The Iterative Model



Assume that the total cost & effort is the same for Model 2 and Model 3. They differ only in how the SDLC is organized.

Questions (15 min) Start time: _____

7. Based on the Iterative Model:

   a) How many stages are there?

   b) Which stage is 7th?

   c) Which stages involve design?

   d) What % of total effort is for the **first four stages**?

   e) What % of total effort is for **testing**?

   f) What % of total effort is for **analysis and design**?

8. Based on the Iterative Model:

   a) During what stage is the project 25% completed?

   b) When the project is 25% completed, what % of **analysis** is done?

   c) When the project is 25% completed, what % of **coding** is done?

   d) When the project is 25% completed, what % of **testing** is done?

   e) During what stage is the project 50% completed?

   f) When the project is 50% completed, what % of **analysis** is done?

   g) When the project is 50% completed, what % of **coding** is done?

   h) When the project is 50% completed, what % of **testing** is done?

9. It is important to find and fix errors in software.

   a) If **analysis** errors are found during **A1: Analyze**,
   in which stage could they be fixed?

   b) If **analysis** errors are found during **B1: Design**,
   in which stage could they be fixed?

c) If **coding** errors are found during **D2: Test**,
in which stage could they be fixed?

d) If **analysis** errors are found during **B2: Design**,
in which stage could they be fixed?

e) Are **analysis** errors likely to cause **design** errors?

f) Are **design** errors likely to cause **coding** errors?

g) Is it better to have **one try** or **several tries**
to remove all errors from the project?

10. Explain why each test stage should try to find as many errors as possible.

11. Explain why **Iterative** is less likely then **Waterfall** to run into projects later in the project.

NOTE: The iterative model does not necessarily repeat exactly three times. The key idea is that it repeats each stage multiple times, for the reasons you have identified.

Chapter 7: Software Engineering

Complete the following Chapter Review Problems on pages 354–355.

#10 (coupling vs cohesion)

#14 (why global variables are bad)

#17 (how to read a structure chart) – *see Figure 7.6 on Page 332*

a)

b)

c)

d)

e)

f)

#33 (diagram vocabulary matching)

a)

b)

c)

#20 (class diagram for magazines) – *see Figure 7.10 on Page 339*

#22 (use case diagram for library of books) – *see Figure 7.9 on Page 339*

#23 (sequence diagram for utility company) – *see Figure 7.5 on page 331*