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

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**.

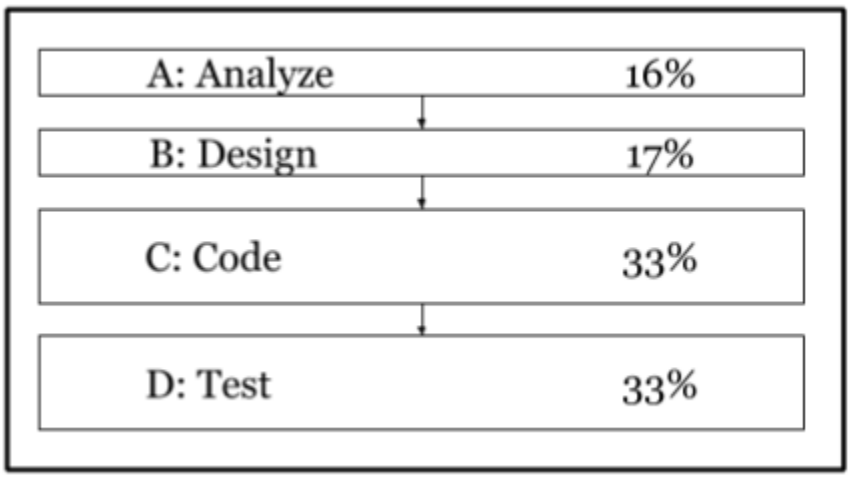
The longer it takes to find an error, the longer and more costly is is to repair.
NOTE: Ambler (2009) and Boehm (1978) contain graphs of this relationship.

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

The faster we find errors, the faster and less expensive it is to fix them.

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? 4

b) Which stage is 1st? A: Analyze

c) Which stage(s) must be finished before **coding** starts? A: Analyze, B: Design

4. Based on the Waterfall Model:

a) What % of total effort is in the **last stage**? 33%

b) What % of total effort is in the **first two stages**? 33%

c) When the project is 25% completed, what % of **analysis** is done? 100%

d) When the project is 25% completed, what % of **coding** is done? 0%

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

f) When the project is 50% completed, what % of **testing** is done? 0%

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? C: Code

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

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

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

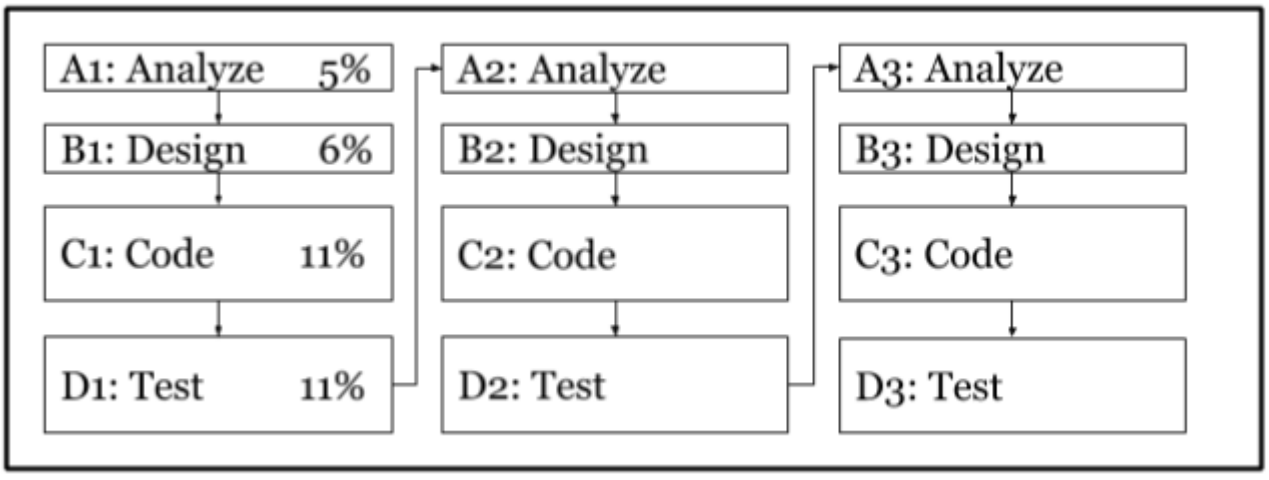
e) Which stage focuses most on **finding** errors? D: Test

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

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

Later stages must fix errors from earlier stages, and many errors are found late in the project during the Test stage.

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? 12

b) Which stage is 7th? C2: Code

c) Which stages involve design? B1, B2, B3

d) What % of total effort is for the **first four stages**? 33% A1+B1+C1+D1

e) What % of total effort is for **testing**? 33% D1+D2+D3

f) What % of total effort is for **analysis and design**? 33% A1+A2+A3 + B1+B2+B3

8. Based on the Iterative Model:

a) During what stage is the project 25% completed? D1

b) When the project is 25% completed, what % of **analysis** is done? 33% A1 only

c) When the project is 25% completed, what % of **coding** is done? 33% C1 only

d) When the project is 25% completed, what % of **testing** is done? About 9% (3%/33%)

e) During what stage is the project 50% completed? C2

f) When the project is 50% completed, what % of **analysis** is done? 67% A1 and A2

g) When the project is 50% completed, what % of **coding** is done? About 52% (17%/33%)

h) When the project is 50% completed, what % of **testing** is done? About 33% (11%/33%)

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? A1: Analyze

b) If **analysis** errors are found during **B1: Design**,
in which stage could they be fixed? A2: Analyze

c) If **coding** errors are found during **D2: Test**,
in which stage could they be fixed? C3: Code

d) If **analysis** errors are found during **B2: Design**,
in which stage could they be fixed? A3: Analyze

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

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

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

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

The sooner you find a defect, (1) the easier it is to fix, and (2) the few other defects it causes.

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

Iterative finds and fixes problems sooner, rather than waiting until the end of the life cycle.

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.