Umaru Musa Yar’adua University Katsina

Faculty of Natural & Applied Science

Department of Computer Science

CSC4311

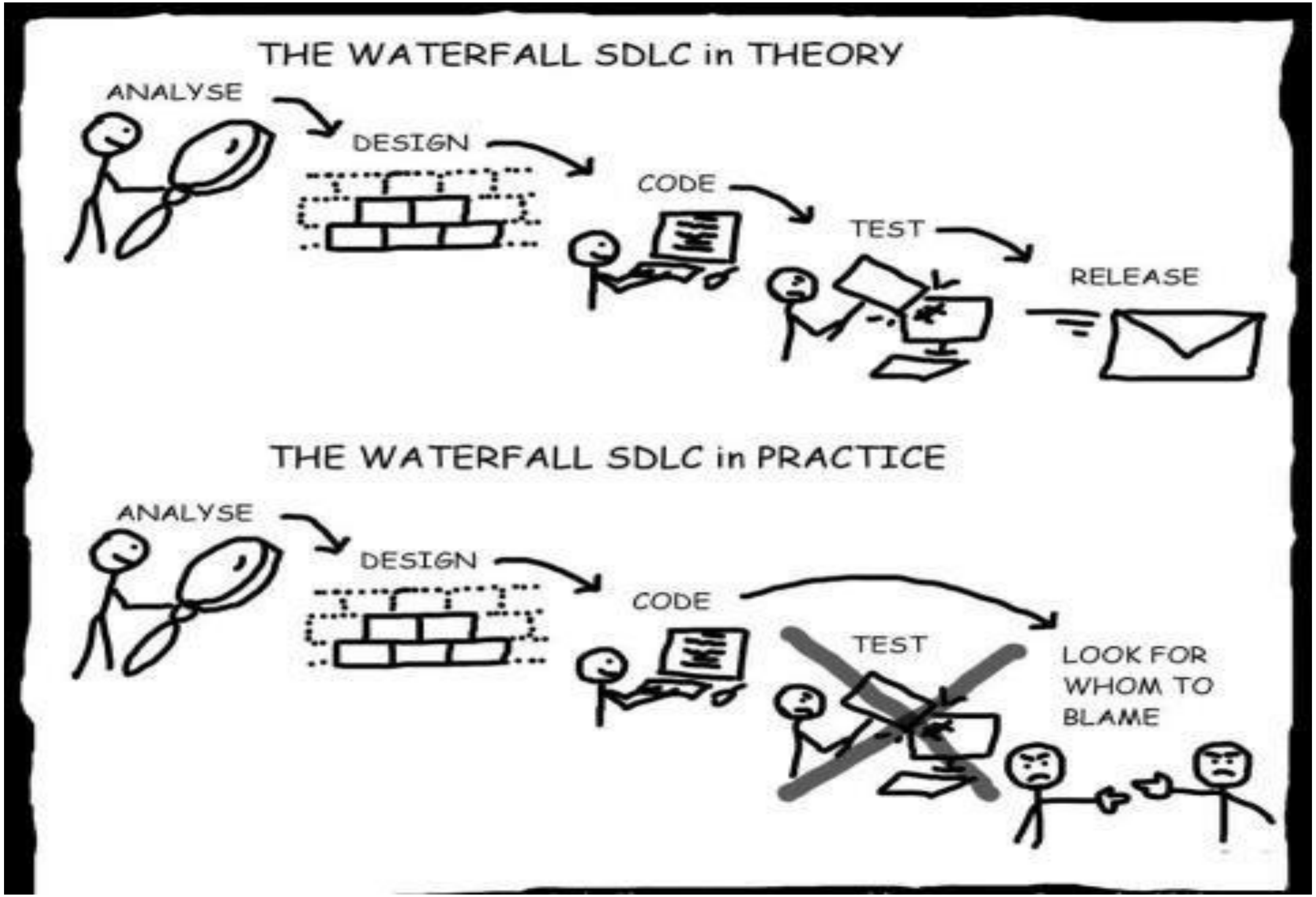
SOFTWARE ENGINEERING

LECTURE ONE

LECTURER: DR. AMINU AMINU MUAZU

COURSE INFORMATION

Where we are…


THE WATERFALL SDLC in THEORY
ANALYSE
DESIGN
CODE
TEST
RELEASE
THE WATERFALL SDLC in PRACTICE
ANALYSE
DESIGN
CODE
TEST
LOOK FOR WHOM TO BLAME

Course Evaluation

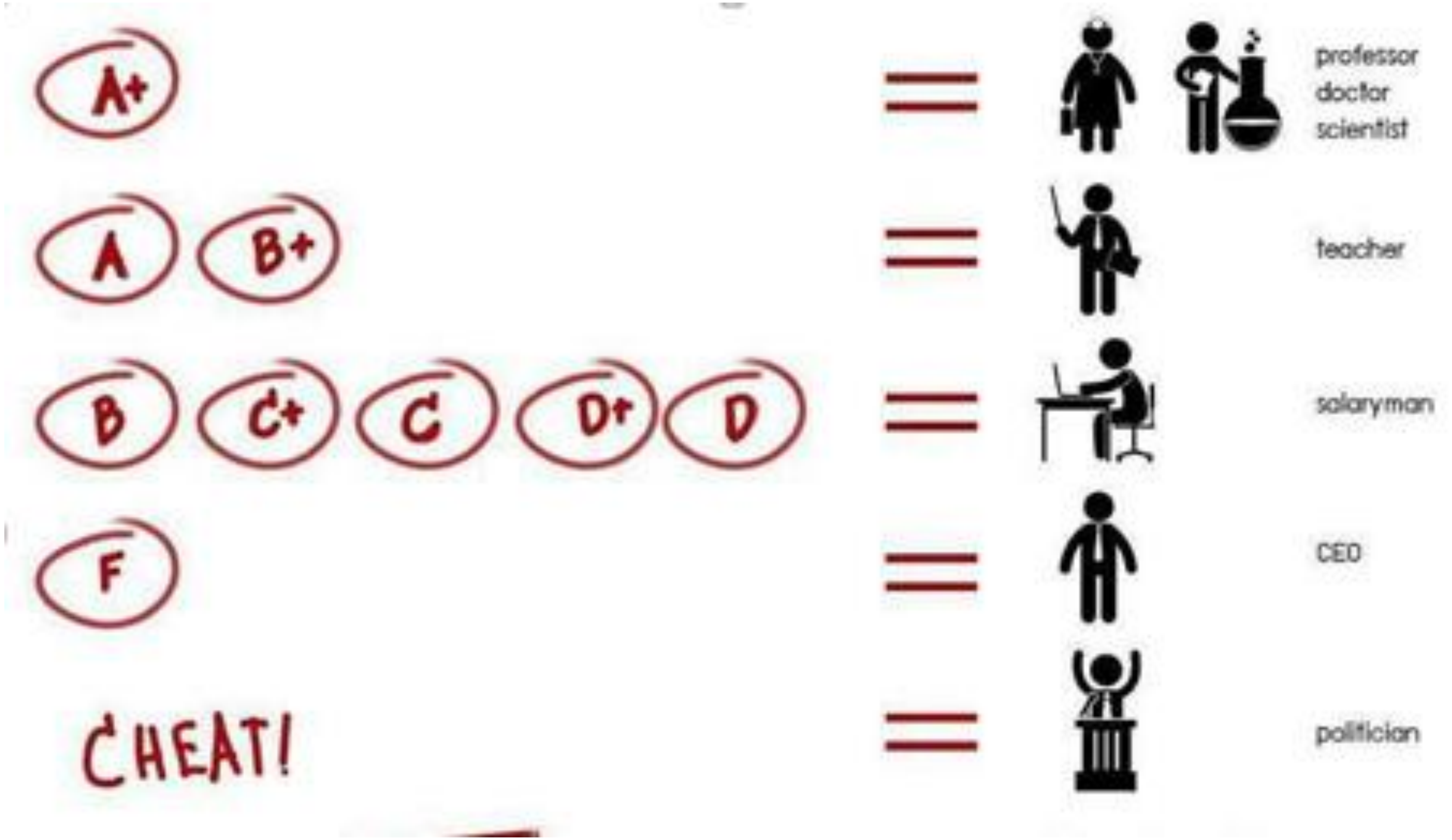
Contact Hours: 2 - 3 hours lecture

Final Exam: 60%

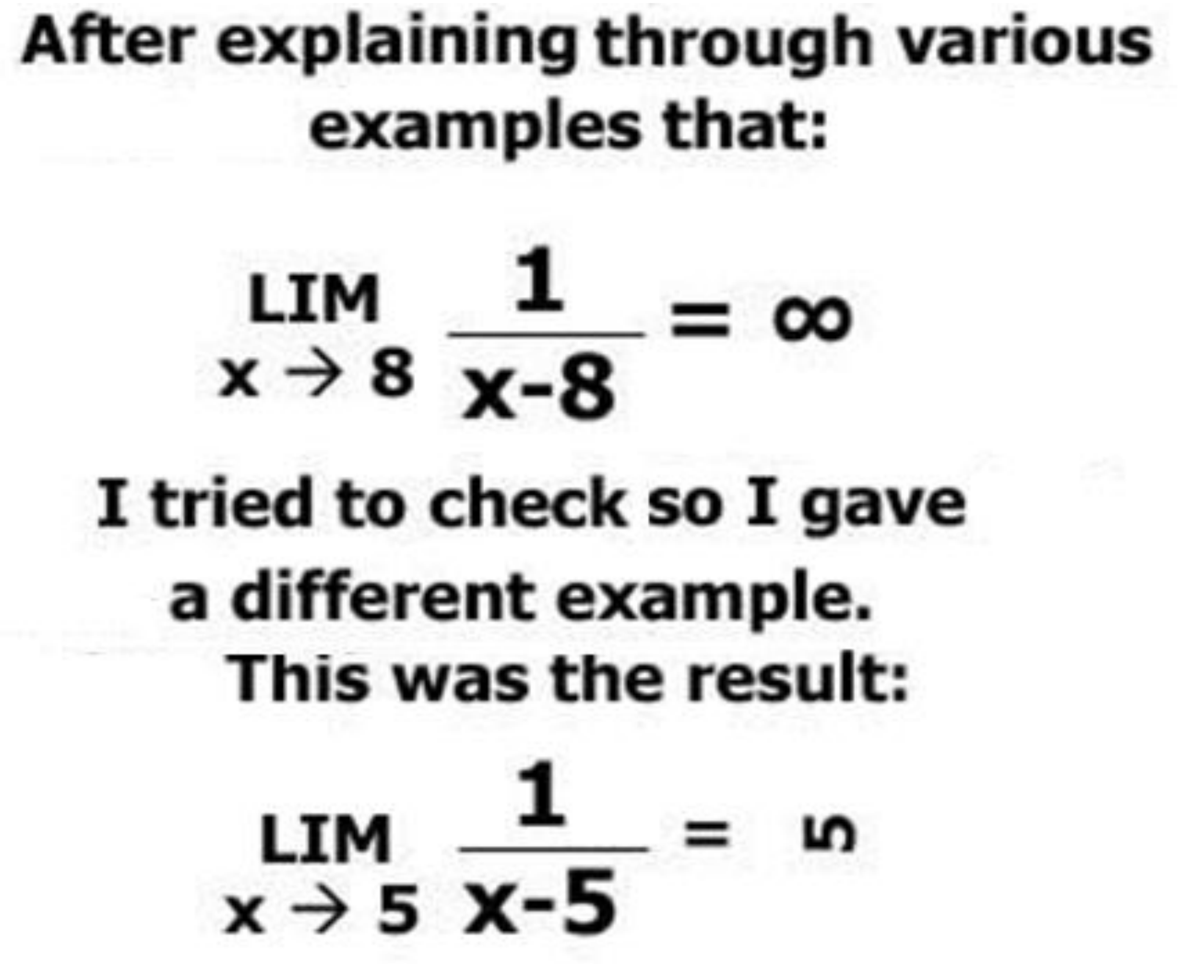
Course Work: 40%

Methods	**Weighting**
Attendance	10%
C.A Test	20%
Group Project	10%
Final Exam	60%
Total	100%

What grade do you want in this class?



Is this what we want from this course?



After explaining through various examples that:

$$\lim_{x \to 8} \frac{1}{x-8} = \infty$$

I tried to check so I gave a different example.

This was the result:

$$\lim_{x \to 5} \frac{1}{x-5} = 5$$

Course Topics

- Introduction to Software Engineering
- Software process
- Requirement analysis
- Design
- Coding
- Testing
- Software Configuration Management

INTRODUCTION TO SOFTWARE ENGINEERING

Overview

- What is software?
- Software in our lives
- Hardware vs. Software
- What is **software engineering**?
- Software engineering - precis of a short history
- Software myths
- Summary

What is software?

- Computer programs and associated documentation

- Software products may be developed for a particular customer or may be developed for a general market.

- Software products may be:
  - Generic - developed to be sold to a range of different customers
  - Custom - developed for a single customer according to their specification

Software is ubiquitous

- System software
  - OS, compilers, device drivers
- Business software
  - Payroll, accounting
- Engineering/scientific software
  - Computer-aided design, simulation
- Embedded software
  - GPS navigation, Flight control
- Product-line software (PC-like based)
  - Spreadsheets, word processing, games
- Web-based software
  - Gmail, Facebook, YouTube
- Artificial intelligence software
  - Robotics, artificial neural networks

Hardware vs Software

Hardware

- Well established (recognized)
- The price often decreased
- Standard procedures (actions)
- Need spare parts

Software

- Relatively new field
- The price is constantly rising
- No standard procedures
- Need modification

What is the core business of Bill Gates, one of the richest man in the world ?

Software Crisis?

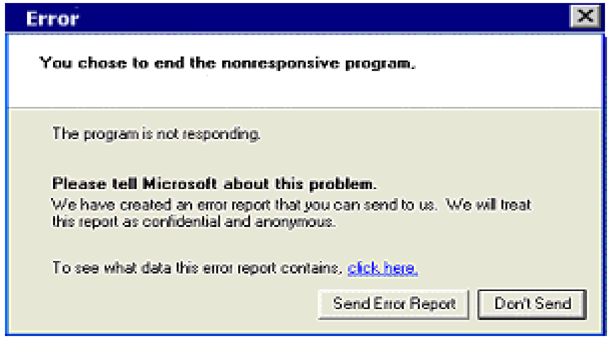
Are we doing the right software?
Are we doing the software right?

- Projects running over-budget
- Projects running over-time
- Software was very inefficient (useless)
- Software was of low quality
- Software often did not meet requirements
- Projects were unmanageable and code difficult to maintain
- Software was never delivered

Software Crisis => Disaster

- Software failures can lead to terrible consequences

- Loss of data .....

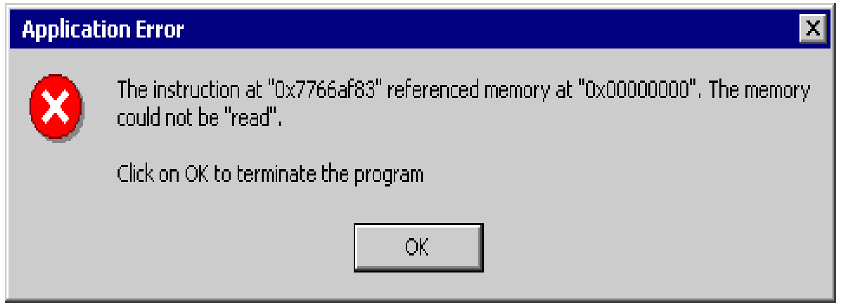


You chose to end the nonresponsive program.

The program is not responding.

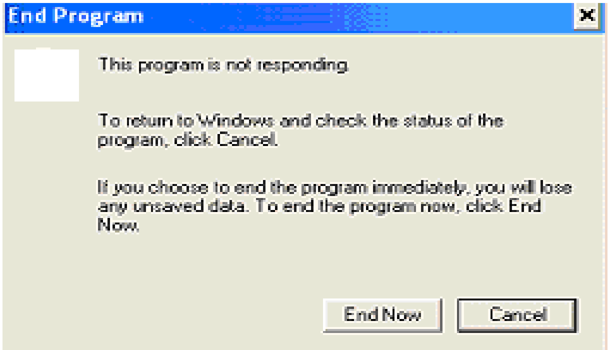
Please tell Microsoft about this problem.
We have created an error report that you can send to us. We will treat this report as confidential and anonymous.

To see what data this error report contains, click here.



The instruction at "0x7766af83" referenced memory at "0x00000000". The memory could not be "read".

Click on OK to terminate the program



This program is not responding.

To return to Windows and check the status of the program, click Cancel.

If you choose to end the program immediately, you will lose any unsaved data. To end the program now, click End Now.

- Loss of fortune .....

- Loss of lives .....

What makes software special?

- ✓ It is difficult for a customer to specify requirements completely.
- ✓ It is difficult for the developer to understand fully the customer needs.
- ✓ In defining and understanding requirements, especially changing requirements, large quantities of information need to be communicated and assimilated continuously.
- ✓ Software is seemingly easy to change (i.e. *malleable).*
- ✓ Software is primarily intangible; much of the process of creating software is also intangible, involving experience, thought and imagination.
- ✓ It is difficult to test software exhaustively

Why does software fail?

- ✓ Terminated for convenience/ non-performance of contract.
- ✓ Completed but the system is not deployed as users cannot or will not use it.
- ✓ Completed but the system does not meet the originally promised cost.
- ✓ Completed but the system does not meet the originally promised schedule.
- ✓ Completed but the system does not meet the originally promised quality.
- ✓ Completed but the system does not meet the originally promised capability.
- ✓ Completed but the system could not be evolved in a cost-effective manner

Best practice

A Solution –Software Engineering

What is software engineering?

Pressman’s book:

A discipline that encompasses

- process of software development
- methods for software analysis, design, construction, testing, and maintenance
- tools that support the process and the methods

Process, Methods, Tools

- Various tasks required to build and maintain software
  - e.g. design, testing, etc.

- SE process: the organization and management of these tasks
  - Using various process models
- SE methods: ways to perform the tasks
- SE tools: assist in perform the tasks
  - UML tools, IDEs, issue tracking tools

History of Software Engineering

- Nato Conference 1968, Garmisch, Germany
  - ✓Led to the discipline called Software Engineering
  - ✓Discusses issues related to software crisis, some of which is still relevant today.

Software Myths

Management Myths

- “If we get behind schedule, we can just add more people and catch up”
- Fact: Adding people to a late project makes it even later
  - The people working now must spend time educating the newcomers

Customer Myths

- “A general statement of objectives (requirements) is enough to start programming”
- Fact: An ambiguous (unclear) statement of objectives leads to project failures
  - Unambiguous requirements need effective and continuous communication between customer and developer

- “Changes in requirements are easy to deal with because software is flexible”
- Fact: Changes are hard and expensive

Practitioner’s Myths

- “Once we get the program running, we are done”
- Fact: 60-80% effort comes after the software is delivered for the first time
  - Bug fixes, feature enhancements, software reengineering, migration

- “The only deliverable work product is the running program”
- Fact: Need the entire configuration
  - Documentation of system requirements, design, programming, and usage

Summary: Software Engineering

- Software is complex, expensive, late, low-quality, hard to maintain
- Goal: approach these problems using software engineering
- Key message: the field is very young –The term “SE” was introduced in 1968