




CS-489
Software Engineering

Prof.: Dr. Jeff Blessing
Office: CC-27a
Phone: 277-7194
Email: blessing@msoe.edu
<http://www.msoe.edu/~blessing/cs489>




1




Logistics

- ❖ Schedule
 - ❖ Office hours
- ❖ Text
 - ❖ Applying UML & Patterns
- ❖ Language
 - ❖ C++/MFC/Java
- ❖ Tools
 - ❖ ROSE, MS Project, Together Control Ctr.
- ❖ Grading
 - ❖ Weekly Quiz (Fri.)
- ❖ Design project (teams)
 - ❖ Formal reports
 - ❖ Reviews
- ❖ Course web page
- ❖ Homework!
- ❖ Questions?

2




Course Objectives



- ❖ Learning Software Engineering concepts
 - ❖ Analysis, Design, Impl., Testing, Maint.
- ❖ Industry Standards for SW development
 - ❖ CMM, PSP, etc.
- ❖ Rational ROSE, Together (UML)
 - ❖ Tutorial: online and in the package itself
 - ❖ <http://www.rational.com/products/rose/>
 - ❖ <http://www.rational.com/uml/>
 - ❖ <http://www.togethersoft.com/>


3



Team Project

- ❖ Ideally, 4-5 person project teams
 - ❖ Assigned based on distributing skills
- ❖ Two passes through the Lifecycle
 - ❖ Textbook organized in this fashion!
- ❖ Software Tool will be implemented
 - ❖ Source code formatter, KLOC, diff, ...
- ❖ Each team presents milestones in lab
 - ❖ Rotate presenters within each group


4



Job Assignments

- ❖ Project Leader
 - ❖ Coordinates all group activities
- ❖ MFC/JFC Specialist
 - ❖ Leads GUI effort and educates others
- ❖ ROSE/Together Specialist
 - ❖ Leads CASE effort and educates others
- ❖ SW Librarian & Tester
 - ❖ Leads effort in SQA area
- ❖ All members are developers!

5



**What is
Software
Engineering
?**

6

Introductory Exercise

- ❖ What does “Software Engineering” mean to you?
- ❖ What knowledge do you expect to gain from this course?
- ❖ What skills do you expect to develop or enhance as a result of taking this course?

7

The Software Crisis

- ❖ Software is:
 - ❖ Always late
 - ❖ Poor quality
- ❖ Software development is:
 - ❖ Impossible to manage
 - ❖ Too expensive

Cautionary Tales

- ❖ Software system failures
- ❖ Documented in:
 - ❖ IEEE Computer, July 1993, p. 18-41
 - ❖ Therac-25
 - ❖ Computer-Related Risks
 - ❖ Peter G. Neumann
 - ❖ ACM Press, 1995

Therac-25

Neumann, p. 68

- ❖ Safety-critical software
 - ❖ Computerized radiation therapy
 - ❖ Six known accidents (1985-87)
 - ❖ Massive overdoses
 - ❖ Deaths and serious injuries
 - ❖ Poor software and interface design

Denver Airport

- ❖ Computer baggage-handling
 - ❖ Real-time software
 - ❖ Individual robotic carts
- ❖ Airport unable to open
 - ❖ Airport cost = \$4.1B
 - ❖ Financing costs ≈ \$1M/day
 - ❖ Finally fixed?

Telephone System

Neumann, p. 14

- ❖ AT&T long distance
 - ❖ January 15, 1990
- ❖ Error in recovery software
 - ❖ Switch crashed due to hardware fault
 - ❖ Software could not handle recovery
 - ❖ Neighboring switches crashed next
 - ❖ Spread to all 114 #4 ESS switches

Telephone System

- ❖ Nationwide blockade: 9 hours
 - ❖ Manual load reduction to stabilize
 - ❖ About 5 million calls blocked
- ❖ Cause
 - ❖ C program with `switch` clause
 - ❖ Embedded `break` inside an `if`
 - ❖ Violated expected practice

More Failures

- ❖ Patriot missile clock drift (34)
- ❖ Military aviation (38)
- ❖ Lufthansa Airbus A320 (46)
- ❖ Budd Company robot (67)
- ❖ Medical databases (71)
- ❖ Emergency dispatch (72)

And More

- ❖ ATM transaction error (190)
- ❖ Auto insurance: age 101 (87)
- ❖ Fifth-grade repetition (192)
- ❖ Molten steel & DS time (89)
- ❖ BC&BS (217)
- ❖ IRS \$4B modernization

Engineering?

- ❖ Predictable?
- ❖ Scientific basis?
- ❖ Standards of practice?
- ❖ A profession?
- ❖ A secure job?

Can we do better?
Are we doing better?

What Can Be Done?

- ❖ No “silver bullet”
- ❖ Requires:
 - ❖ Knowledge
 - ❖ Hard work
 - ❖ Investment
 - ❖ Process reform

Lifecycle Model

- ❖ Requirements Analysis
- ❖ Specifications
- ❖ Design
- ❖ Implementation
- ❖ Testing
- ❖ Maintenance

Requirements Analysis

- ❖ Process of determining WHAT the system is to do.
- ❖ Produces written *descriptions* and working *diagrams*
- ❖ Includes a *feasibility study*
- ❖ Involves *planning* the development process
- ❖ Involves OOA methodology

19

Specifications

- ❖ Sometimes lumped into Analysis
- ❖ Formal documentation of the problem to be addressed by the system to be developed
- ❖ Must conform to standards
 - ❖ company stds. (CMM)
 - ❖ ISO-9000, FDA, FAA, FCC, etc.
- ❖ Functional Spec., User's Guide, Test Plan

20

Design

- ❖ Huge Topic! (volumes have been written)
- ❖ O-O Designs are more accessible designs
 - ❖ also more flexible, maintainable, etc.
- ❖ OOA, OOD, & OOP work together
 - ❖ tools (like ROSE) are built on this idea
- ❖ Design Patterns have emerged
 - ❖ Gamma, et. al., (the "gang of four")
- ❖ Component Design (COTS)

21

Implementation

- ❖ Commonly called "coding"
- ❖ Usually rushed into by "real" programmers!
- ❖ Should naturally be a translation of the design into an appropriate p'gming lang.
- ❖ Should only take about 10% - 20% of the total time spent on the project
- ❖ Also has guidelines
 - ❖ rules about use of header files, recursion, etc.

22

Testing

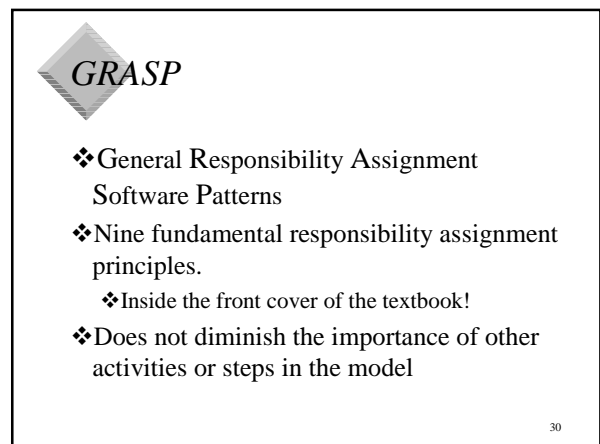
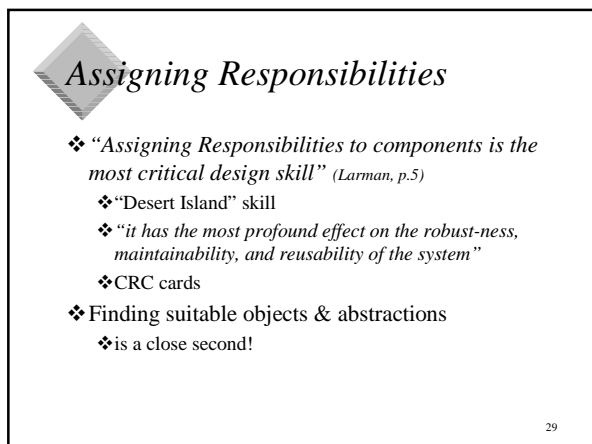
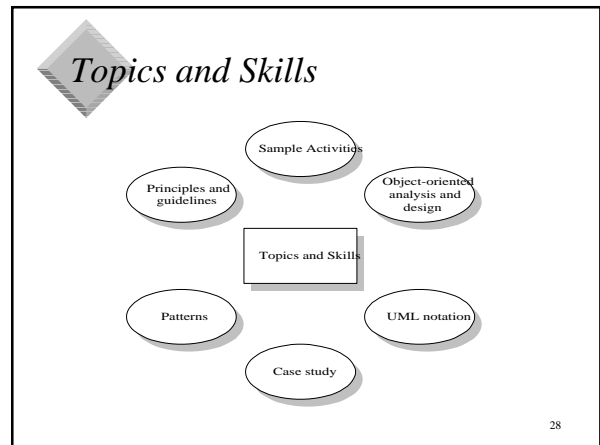
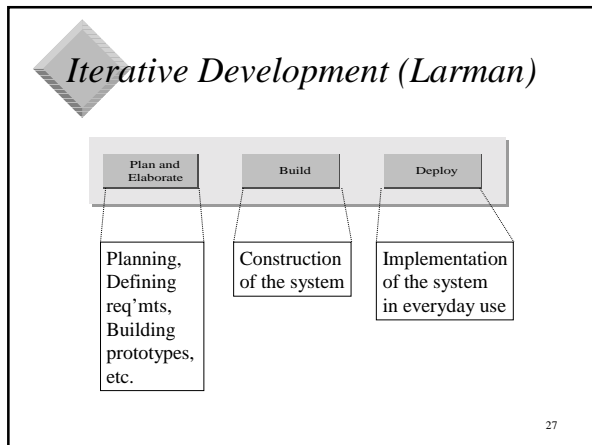
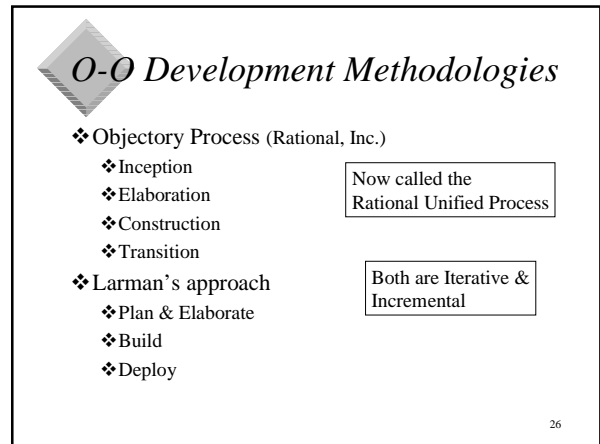
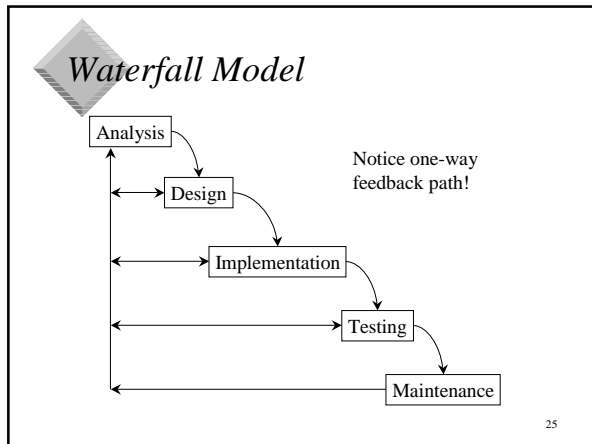
- ❖ Also called Verification & Validation
- ❖ Verification
 - ❖ conforming to specification
- ❖ Validation
 - ❖ correctness of the software
- ❖ Done in stages:
 - ❖ Unit, Integration, Regression, Beta, Acceptance
- ❖ Not to be confused with debugging!

23

Maintenance

- ❖ Too often associated with fixing "bugs"
 - ❖ Points out a weakness in testing!
- ❖ Activities which happen after release of the software system
- ❖ Customization to suit a client
- ❖ Feature enhancements
 - ❖ Further development

24



What is Analysis & Design

- ❖ Analysis
 - ❖ Investigates the problem, rather than finds a solution
 - ❖ Concentrates on answering the "What" questions
 - ❖ also the "Who" questions
- ❖ Design
 - ❖ Focuses on finding a suitable solution
 - ❖ Concentrates on answering the "How" questions
 - ❖ also the "Where" and "When" questions.

31

What is O-O Analysis & Design

Analysis

Investigation of the problem

Design

Logical solution

Construction

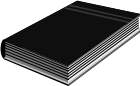
Code

The emphasis on activities is the same, but from the point of view of objects in the problem domain.

things, concepts, or entities

32

domain concept



representation in analysis of concept


Book

title

representation in an object-oriented programming language

```
public class Book
{
  public void print()
  private String title
}
```

33



34