

Course Topics

•OO analysis & design

- UML, use cases, design patterns
- CASE tools

•Implementation

- Code, review, test, document

•Project management

- Team roles
- Effort and quality tracking

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

SEI CMM Model

- **Capability Maturity Model**
 - Software Engineering Institute (CMU)

- 1 Initial (chaotic, unpredictable)
- 2 Repeatable (disciplined process)
- 3 Defined (standard, consistent)
- 4 Managed (predictable)
- 5 Optimizing (continuously improving)

Competition

- Many organizations at level 1
- But many now at levels 3-5
 - Motorola (most groups at level 3)
 - Aerospace (many at levels 4-5)
 - Winning contracts on this basis
- Compete on quality, reliability

CS-489

- Methods
 - Object-oriented process examples
 - Design patterns
 - CASE tool usage
- Project experience
 - Compressed schedule
 - Two implementation cycles
 - Apply process
