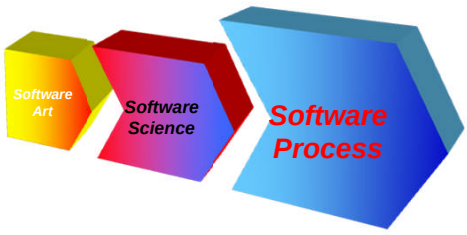
Comp 330 – Software Engineering

Dr. William L. Honig
Associate Professor
Department of Computer Science

Evolution of Software Creation

1960 → 1990



COMP 330 Software Engineering

Objective
- Introduce students to process-based software development
- Experience the dynamics of team software development

Approach
- Learn activities of software engineering
- Work as a team for full semester
- Experience full life cycle(s)
- Develop single project for full semester

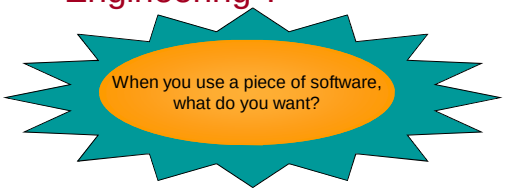
Software engineering

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development
- Software engineering expenditure represents a significant fraction of GNP in all developed countries

Software costs

- Software costs often dominate system costs. The costs of software on a PC are often greater than the hardware cost
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs
- Software engineering is concerned with cost-effective software development

What is Software Engineering ?



What are some words that come to mind?

-
-
-

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

About Me…

- 33 years experience telecommunications
  - Product development, R&D
  - Software, Hardware, Systems, Marketing
- Accomplishments
  - Implemented telephone switches, packet networks, satellite systems, mobile phone infrastructure
  - Managed R&D organizations of 200+ people
  - Worked on two grand failures
- Lived around the country and world
- Hired over 200 new college graduates

What is a software process?

- A set of activities whose goal is the development or evolution of software
- Generic activities in all software processes are:
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands

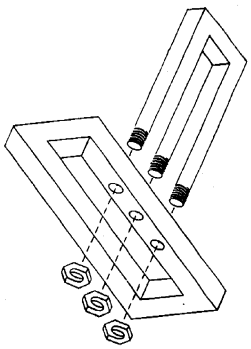
Objectives of the Class

- Appreciate Software Engineering:
  - **Build complex software systems in the context of frequent change**
- Understand how to
  - **produce a high quality software system within time**
  - **while dealing with complexity and change**
- Acquire technical knowledge (main emphasis)
- Acquire managerial knowledge

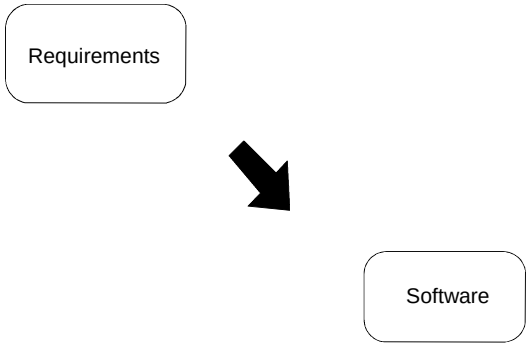
Acquire Managerial Knowledge

- Understand the Software Lifecycle
  - **Process vs Product**
  - **Learn about different software lifecycles**
  - **Greenfield Engineering, Interface Engineering, Reengineering**

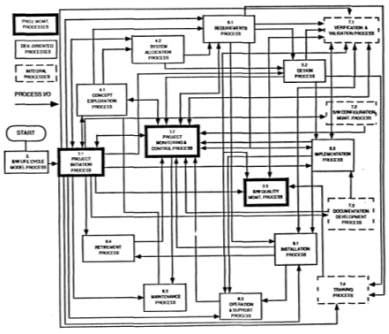
Can you develop this?



Limitations of Non-engineered Software



What is this?



1. Abstraction

- Inherent human limitation to deal with complexity
  - **The 7 +- 2 phenomena**
- Chunking: Group collection of objects
- Ignore unessential details: => Models

Software Production has a Poor Track Record Example: Space Shuttle Software

- Cost: $10 Billion, millions of dollars more than planned
- Time: 3 years late
- Quality: First launch of Columbia was cancelled because of a synchronization problem with the Shuttle's 5 onboard computers.
  - **Error was traced back to a change made 2 years earlier when a programmer changed a delay factor in an interrupt handler from 50 to 80 milliseconds.**
  - **The likelihood of the error was small enough, that the error caused no harm during thousands of hours of testing.**
- Substantial errors still exist.
  - **Astronauts are supplied with a book of known software problems "Program Notes and Waivers".**

Software Engineering: Definition

Software Engineering is a collection of techniques, methodologies and tools that help with the production of

- a high quality software system
- with a given budget
- before a given deadline

while change occurs.

Scientist vs Engineer

- Computer Scientist
  - **Proves theorems about algorithms, designs languages, defines knowledge representation schemes**
  - **Has infinite time…**
- Engineer
  - **Develops a solution for an application-specific problem for a client**
  - **Uses computers & languages, tools, techniques and methods**
- Software Engineer
  - **Works in multiple application domains**
  - **Has only 3 months...**
  - **…while changes occurs in requirements and available technology**

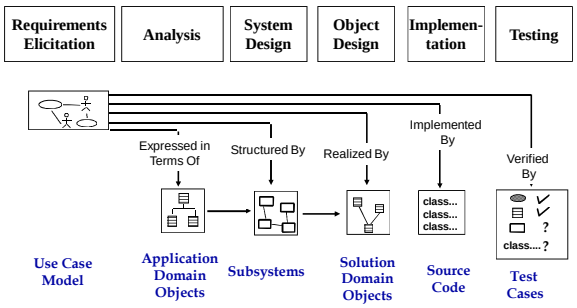
Factors affecting the quality of a software system

- **Complexity:**
  - The system is so complex that no single programmer can understand it anymore
  - The introduction of one bug fix causes another bug
- **Change:**
  - The "Entropy" of a software system increases with each change: Each implemented change erodes the structure of the system which makes the next change even more expensive ("Second Law of Software Dynamics").
  - As time goes on, the cost to implement a change will be too high, and the system will then be unable to support its intended task. This is true of all systems, independent of their application domain or technological base.

Why are software systems so complex?

- The problem domain is difficult
- The development process is very difficult to manage
- Software offers extreme flexibility
- Software is a discrete system
  - **Continuous systems have no hidden surprises (Parnas)**
  - **Discrete systems have!**

Software Lifecycle Activities ...and their models



Software Lifecycle Definition

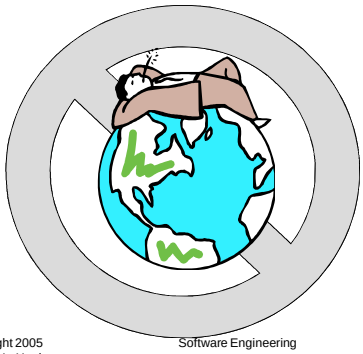
- Software lifecycle:
  - **Set of activities and their relationships to each other to support the development of a software system**
- Typical Lifecycle questions:
  - **Which activities should I select for the software project?**
  - **What are the dependencies between activities?**
  - **How should I schedule the activities?**

Summary

- Software engineering is a problem solving activity
  - **Developing quality software for a complex problem within a limited time while things are changing**
- There are many ways to deal with complexity
  - **Modeling, decomposition, abstraction, hierarchy**
  - **Issue models: Show the negotiation aspects**
  - **System models: Show the technical aspects**
  - **Task models: Show the project management aspects**
  - **Use Patterns: Reduce complexity even further**
- Many ways to do deal with change
  - **Tailor the software lifecycle to deal with changing project conditions**
  - **Use a nonlinear software lifecycle to deal with changing requirements or changing technology**
  - **Provide configuration management to deal with changing entities**

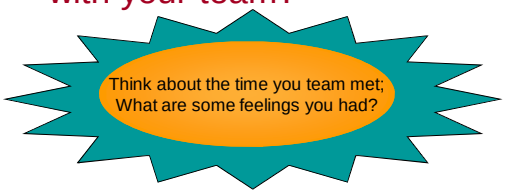
One Team Role NO ONE Has

LOYOLA UNIVERSITY CHICAGO



- It will not be possible to rely on others to do your work!

What is it like working with your team?



What are some things you remember feeling in the meeting?

-
-
-

Team Building – Also Part of the Job

- Some "Natural" Team Building Activities
  - Assign, discuss, agree on team roles
  - Confirm schedules, planned meetings
  - Agree on attendance requirements
- Also some "Special" Team Building Activities
  - Not specifically job related
  - Take time AWAY from job to be done
  - Goal: Improve team work and team understanding
  - Pay attention to TEAM before PROBLEMS arise
    - ➢Most Critical at Start of Team
    - ➢Speed up the Ability of the Team to Work Together

Team Building – A Simple Team Building Activity

- Before team meeting, each member write down the answer to these three questions:
  1. What is the last book (not a textbook) I read?
     Write down author and title
  2. Name of some unusual place I have visited?
     Write down the name and location of the place
  3. Something about my family?
     Write down something about your family members, family activities, etc.
- At first team meeting
  - Put all papers in the middle of table
  - Someone picks a paper and reads it out loud
  - All team members try to guess who it is (take 2 minutes max)