

# INF 111 / CSE 121: Software Tools and Methods

Lecture Notes for Fall Quarter, 2007  
Michele Rousseau  
Set 2

(Some slides adapted from Susan E. Sim)

---

---

---

---

---

---

---

---

## Today's Lecture

- More Review of S/W Engineering
- Overview of Software Tools
  - Why we need them and what they are

Topic 2

2

---

---

---

---

---

---

---

---

## From Programming to Engineering

- **People**
  - Who else would do the work?
  - Range from novice to very experienced
- **Processes**
  - To organize and manage the efforts of individuals
  - Range from informal to very formal
- **Tools**
  - To support the people and the processes
  - Range from simple to very advanced

Topic 2

*People + Processes + Tools ⇒ Product*

3

---

---

---


---

---

---

---

---



## People

- **The single most important factor in the success/failure of a product**
- **Scarce resource**
  - Quality
  - Suitability
  - Cost
- **Many different kinds of people**
  - Managers
  - Programmers
  - Technical writers
  - Testers

Topic 2 4

---

---

---


---

---

---

---

---



## Processes

- **Essential to achieve a quality product**
- **(Time is a) scarce resource**
  - Quality
  - Suitability
  - Cost
- **Many different kinds of processes**
  - Bug tracking
  - Change approval
  - Quality assurance

Topic 2 5

*Focus of ICS 52*

---

---

---


---

---

---

---

---



## Tools

- **Needed to support people and processes**
- **Scarce resource**
  - Quality
  - Suitability
  - Cost
- **Many different kinds of tools**
  - Drawing
  - Analysis
  - Project management
  - Source code management

} people support

} process support

Topic 2 6

*Focus of This Class*

---

---

---

---

---

---

---

---

## Product

- **Result of applying people, processes, and tools**
- **Consists of many deliverables**
  - Software
  - Documentation
  - User manuals
  - Test cases
  - Design documents
- **Intrinsic qualities**
  - Safety
  - Reliability
  - User friendliness

Topic 2 7

---

---

---

---

---

---

---

---

## People, Processes, Tools, Products

- **Products are always the eventual goal**
  - Selling products creates revenue
  - Selling good products creates lots of revenue
  - Selling bad products creates little revenue
- **People, processes, and tools are retained by organization**
  - Build a reputation through the quality of products
  - Create organizational culture
  - Important to keep the team intact

Topic 2 *People + Processes + Tools ⇒ Product* 8

---

---

---

---







---

---

---

---

## Choose the Right Solution

	↔	Hawaii	
	↔	Tijuana	
	↔	Europe	

Topic 2 9

---

---

---

---

---

---

---

---

## Matters of Scale

When orders-of-magnitude improvement are required, new technology may be necessary

- Choose appropriate technique for problem
  - elephant gun to kill a fly?
  - fly-swatter to ward off an elephant?

Topic 2 10

---

---

---

---

---

---

---

---

## What is a Software Lifecycle Model?

- “A *software life cycle model* is either a descriptive or prescriptive characterization of how software is or should be developed.” [scacchi]
- “abstract representation of a process” [sommerville]

Topic 2 11

---

---

---

---

---

---

---

---

## Software Life Cycle Models

- Build-and-fix
- Waterfall
- Rapid prototyping
- Incremental
- Spiral

*A software life cycle model is a high-level process*

Topic 2 12

---

---

---

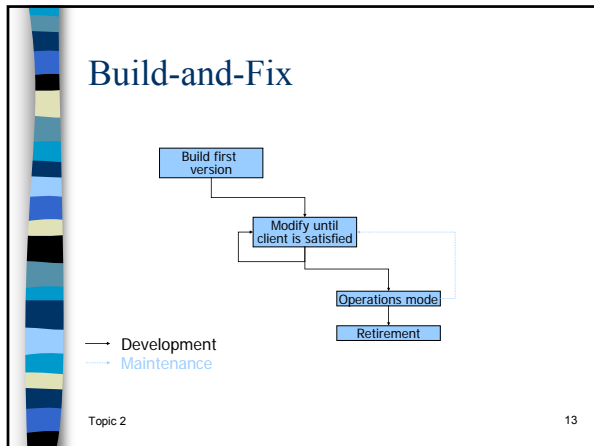
---

---

---

---

---




---

---

---

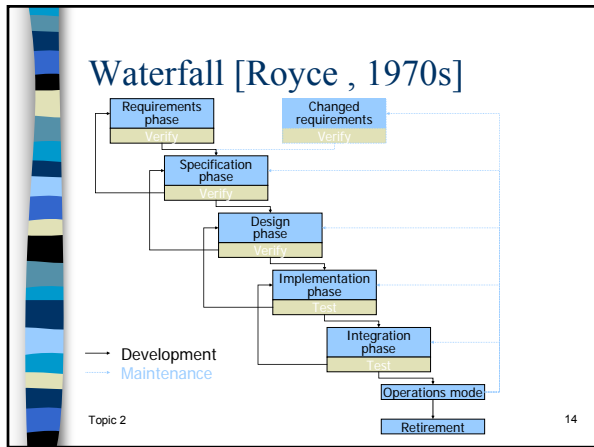
---

---

---

---

---




---

---

---

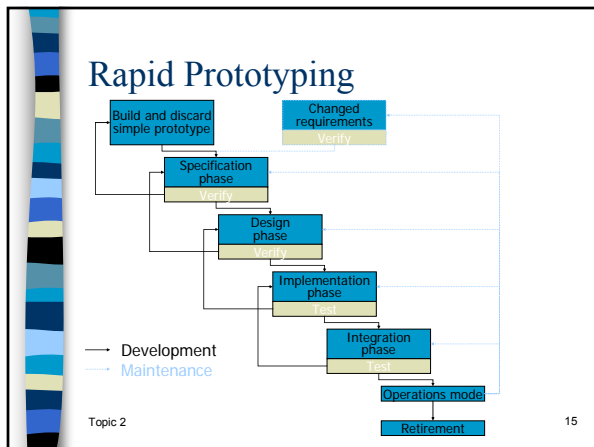
---

---

---

---

---




---

---

---

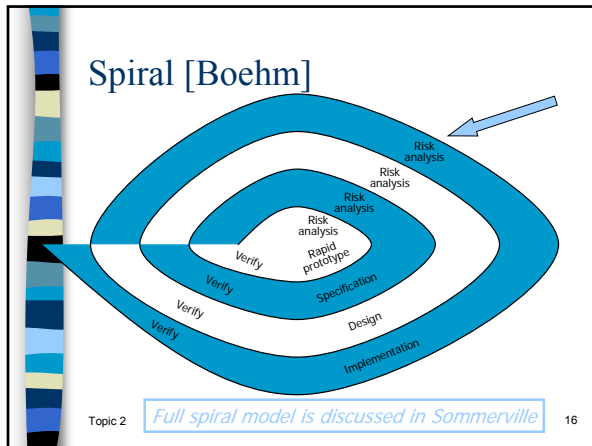
---

---

---

---

---




---

---

---

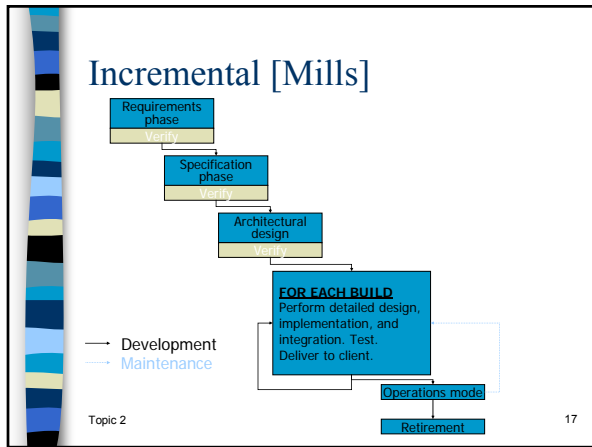
---

---

---

---

---




---

---

---

---

---

---

---

---

### Moving on...

- o **Methods and Tools**

Topic 2 18

---

---

---


---

---

---

---

---



## Tools Students Have Typically used

- **Text editors**
- **Maybe some have used IDE**  
(Integrated Development Environments like Netbeans)
- **File system to manage projects**
- **Print statement for debugging**

Topic 2 19

---

---

---


---

---

---

---

---



## Problem – Approach Doesn't Scale

- **May work for 1 or 2 students**
- **Doesn't work with..**
  - More people
  - Bigger code
  - Different versions
  - Multiple Platforms

Topic 2 20

---

---

---


---

---

---

---

---



## Why Do We Need Tools and Methods?

- **Scaling Problem**
- **Ex: Longhorn project**
  - 50+ Million lines of code
  - Daily Builds and Regression testing
    - Takes 3 days from the time you submission to executable
  - Needs to be backwards compatible
  - Installation needs to work on millions of machines
  - 4000 Programmers
  - ~1.7 Testers for each programmer
  - \$2 billion
  - 6 years in development

Topic 2 21

---

---

---


---

---

---

---

---



## Challenges

- **Logistics**
  - How do you design a process that will allow thousands of people to work together at the same time?
  - How do you test?  
So many platforms so little time
- **Design**
  - How do you design a system with 50+ mill lines of code?
  - How do you maintain conceptual (architectural) integrity?

Topic 2 22

---

---

---

---

---

---

---

---



## Software Technology

- **Types of Software Technology**
  - Tools
  - Methods
  - Notations
- **How do they help?**
  - Automate tasks
  - Help people to do complex tasks
  - Improve s/w quality
  - Increase productivity
  - Permit verification and conformance checking
  - Project tracking
  - Establish procedures

Topic 2 23

---

---

---


---

---

---

---

---



## Tools

- **IDE (Integrated Development Environment)**
- **Compiler**
- **Debugger**
- **Diagram Tools**
- **Automated testing**
- **Static checking**

Topic 2 24

---

---

---

---

---

---

---

---



Methods

- Process Models
- Unit / System Testing
- Pair Programming
- Test driven development
- Class-Responsibility-Collaboration Cards

Topic 2 25

---

---

---

---

---

---

---

---

Notations

- Programming languages
  - Java, C, C++
- Data formats
  - HTML, XML, Comma Delimited
- Logic Notation
  - Propositional or Predicate Logic
- Design Notations
  - UML
- Requirements Notations
  - Z notation, Larch, Nitpick, SCR, Natural Language

Topic 2 26

---

---

---

---


---

---

---

---

Example: Guitar



- Strum
- Pick
- ... etc

Topic 2 27

---

---

---

---


---

---

---

---





## Conflict: Practice and Problems

- **Why is there a gap between actual practice and state of the art?**
  - Both for students and industry
- **Focus is on the end product – not the process**
  - Don't have time for a new tool or method
  - Learning Curve

Topic 2 31

---

---

---


---

---

---

---

---



## First time using a tool?

- **You know it might exist**
- **Have to find it**
  - Decide which one to download
- **Download / Buy it**
- **Install it**
- **Figure it out**
  - (hmm what does that button do?)
  - How do I get it to help me
  - Hmm what does this error message mean....
  - Sheesh -- this is just slowing me down
- **Uncertain payoff**

Topic 2 32

---

---

---


---

---

---

---

---



## Learning Curve

- **It takes  $\geq 10,000$  hours to become and expert**
- **A s/w eng needs to know:**
  - Programming languages
  - Programming techniques
  - Tools
  - Methods
  - Notations
  - Domain knowledge

...and how & when to apply each of these

Topic 2 33

---

---

---

---

---

---

---

---

## Summary

- Personal Practices and S/W Eng GAP
- Focus tends towards *What and not How*
- Steep learning curve and unclear payoffs dissuades users

This course will help with all of these

Topic 2

34

---

---

---

---

---

---

---

---