

INF 111 / CSE 121: Software Tools and Methods

Lecture Notes for Summer, 2008
Michele Rousseau

Lecture Notes 1 – Administrative / Intro to Software Eng.

Announcements

1. **Brush up on your Java**
2. **Read Van Vliet Chs. 1, 3 and 15**
3. **Lab 1 & 2 Due on Thursday**
4. **WELCOME TO CLASS ☺**

Lecture Notes 1

2

Today's Lecture

- **Administrative details**
- **Review of Software Engineering**

Lecture Notes 1

3

Introductions

• Instructor

- Michele Rousseau
- Email: michele@ics.uci.edu
 - Please put INF111 in the Subject
- Office Hours: by appointment
- Office: DBH-5204 (for now)

Lecture Notes 1

4

Introductions

• Teaching Assistant

- Kristina Winbladh
- Email: awinblad@ics.uci.edu
- Office Hours: Thursday Noon
 - In Lab

Lecture Notes 1

5

Pre-requisites

- **Informatics 43 OR ICS 52 OR CSE90**
 - with a grade of C or better
- **ICS 6D / Mathematics 6D**
- **ICS 6B / Mathematics 6B**
- **Mathematics 6G or 3A**
- **Lower-division writing**

- **Please let me know if you have not satisfied these requirements**

Lecture Notes 1

6

Class Information

Website

- www.ics.uci.edu/~michele/Teaching/INF111-Sum08
- Can access from my home page
 - ▣ www.ics.uci.edu/~michele

Lecture Notes 1

7

Course Materials

Required textbooks

- Van Vliet, Hans
Software Engineering: Principles and Practice.
3rd edition.
- Brooks, Fredrick
The Mythical Man-Month.
Anniversary edition.
Will be announced on the website and in lecture

Recommended Readings

- Humphrey, Watts
The Personal Software Process
- Will be announced on the website and in lecture

Lecture Notes 1

8

Course Mechanics (1)

Lecture

- TTh: 9a – 11:50a (ELH 110)

Discussion

- Tue: 1p - 1:50p (DBH 1200)

Lab

- Thu: 1p - 1:50p (ICS 192)

Lecture Notes 1

9

Discussion is for...

- Discussing Details about the Assignments / Labs
- Presenting Tools
- Discussing Readings
- Reviewing Material
- Preparing for Tests
- Reviewing Tests & Assignments

Lecture Notes 1

10

How to be successful (1)

Attend class

- Much is covered that is not in the textbook
 - ▣ Material is core part of the exams
 - ▣ What is said in class supercedes all else
- Official place for announcements

Attend Discussion Section & Lab

Visit course Web site on a regular basis

- Assignments
- Labs
- Lecture Slides

Lecture Notes 1

11

How to be successful (2)

- Check EEE
- Do the Assignments / Labs
- Do the Readings
- Use Office Hours
- Ask Questions

Lecture Notes 1

12

Grading

Assignments	40%
Labs	15%
Quizzes	20%
Final	25%

- Will scale only if necessary

Lecture Notes 1

13

Assignments (1)

○ 3 Assignments

- Will have at least 1 week to complete each Assignment

○ Format of the assignments

- Questions on Effort Estimation
- Use a software tool to complete a task
- Reflect on and document your experience

Lecture Notes 1

14

Assignments (2)

○ Package properly

- Every assignment...
 - ...has your student ID
 - ...has your name
 - ...has a cover page with class title, Name, student ID and assignment #
 - ...is properly stapled
- Electronic Submissions through Checkmate

○ Quality over quantity

○ Express yourself clearly

- Be concise

Lecture Notes 1

15

Assignments (3)

○ A new forum will be created on the Noteboard for each assignment

- Post your questions there
- Ta, other students, and I will post answers, suggestions and additional information

○ Don't wait until the last minute to post (The TA may not be awake or available)

○ Please turn in on time

- Do not wait until the last minute

○ No late assignments

Lecture Notes 1

16

Exceptions for being late

○ At the TA/Instructor's discretion

- Contact the instructor/TA as soon as possible
- Preferably before you are late

○ Valid reasons

- Serious illness, accident, family emergency, etc.

○ Not-so-valid reasons

- "No ink in my printer", "didn't know it was due today", "my computer crashed", "couldn't find parking", etc.

Lecture Notes 1

17

Laboratories

○ We will drop the lowest lab (6 Labs)

- Work needs to be completed by the END of the lab session
- No late labs will be accepted
- Usually will require work to be done by the end of Lab
- Sometimes will need to submit a write-up afterwards
- Electronic submissions through Checkmate

Lecture Notes 1

18

Quizzes and Final Exam

- **Quizzes**
 - Weekly
 - Primarily based on **lectures** and **readings**
 - **Must be in pen** if you want it re-graded
- **No Make-up Quizzes**
- **Final will**
 - Primarily based on **lecture** and **readings**
 - May include some elements from **assignments, discussion** and **labs**

Lecture Notes 1

19

Grading

- **Performed by the TA**
- **Disputes**
 - Let me know **ASAP** (by the next class)
 - Please don't play the "points-game"
 - I have limited time
 - **Check your grading thoroughly** and ASAP
 - Include a **coversheet** with your name, student ID, and a detailed description of the error
- **Re-grading**
 - Will only accept re-grades at the beginning of the class following the date they were returned
 - Must be accompanied with a **clear explanation** of what needs to be reconsidered and why
 - **Entire assignment** will be considered

Lecture Notes 1

20

Questions

When in doubt

- **Ask Me or the TA**
 - Open door policy
 - Attend Office Hours
 - Attend Discussion
- **Check the Noteboard (eee)**
- **Email me**
 - If I think the whole class could benefit I'll forward it
 - let me know if you specifically don't want it forwarded
- **Ask your friends**

Lecture Notes 1

21

Academic Dishonesty (ugh)

- **Please don't Cheat**
 - Know the academic dishonesty **policies** (for ICS & UCI)
 - ICS: <http://www.ics.uci.edu/ugrad/policies/>
 - UCI: <http://www.editor.uci.edu/catalogue/appx/appx.2.htm>
- **If you do...**
 - **Final grade** is an "F", irrespective of partial grades
 - Assignments, Quizzes, Labs, or Final
 - **Letter in your UCI file**
- **Anything copied from a book or website needs to be quoted and the source provided**

22

Help each other but don't share work

- **Your TA is your friend, but your friend is not your TA**
 - Your friend's help may be cheating
- **To avoid being a cheater**
 - Always do your work by yourself
 - **It is okay to...**
 - ... ask your friends about **how solve/approach** a problem
 - ... **discuss** an assignment
 - **It is not okay to...**
 - ... ask for the **answer/solution**
 - ... **copy work**
 - ... have them **do it for you!**
 - ... put your work on the **Web**
 - ... **borrow** or **lend** work!
 - **When in doubt – ask me!**
- **Use good Judgment**

23

Add/Drop/Change of Grade Policy

- **Adding or Dropping the Class**
 - Check with Summer Sessions
 - Check with the Department
 - If they are good with it – so am I
- **Changing Grade to P/NP option**
 - Check with Summer Sessions
 - Check with the Department
 - If they are good with it – so am I
- **Please bring **completed** Add/Drop Cards (2 cards)**
 - **In Pen PLEASE ☺**

24

Other Policies

- Please use your **UCI or ICS account**
 - This is for your privacy
 - Needs to be activated if you are a new student
 - include **INF111** in the subject of all email
- Questions of general interest will be forwarded to the board
 - if you don't want it forwarded for some reason please state that
- If you need accommodations due to a disability, talk to me

Lecture Notes 1

25

Miscellaneous

- **INF 111 / CSE 121 mimics the real-world**
 - Package your assignments/labs properly
 - Use proper language and proper form
- You get out of this class what you put into it
 - Attend Class, Discussion , & Labs
 - Participate
 - Follow instructions
 - Read and study the textbook and slides
 - Help is available, do not be afraid to ask questions

Lecture Notes 1

26

Course Objective

- To teach you basic tools & methods used in software engineering development

Lecture Notes 1

27

Now to the fun part...

- Brief review of S/W concepts
- Overview of Software Tools and Methods

Lecture Notes 1

28

What is Software Engineering?

- "A discipline that deals with the *building of software systems* which are *so large* that they are built by a *team* or teams of engineers." [Ghezzi, Jazayeri, Mandrioli]
- "A discipline whose *aim* is the production of *fault-free software, delivered on-time and within budget, that satisfies the user's needs*. Furthermore, the software must be *easy to modify* when the user's needs change." [Schach]
- "Software engineering is concerned with *theories, methods and tools for professional software development*" [Sommerville]

Lecture Notes 1

29

Best practices for Software Engineers

- Software engineers should adopt a **systematic and organised approach** to their work and use **appropriate tools and techniques** depending on the **problem** to be solved, the development **constraints** and the **resources** available.

Lecture Notes 1

30

Why Software Engineering?

- Software Crisis – Software is typically
 - Late
 - Over Budget
 - Wrong product
- OBJECTIVE:
 - Reduce costs and increase quality

Lecture Notes 1

31

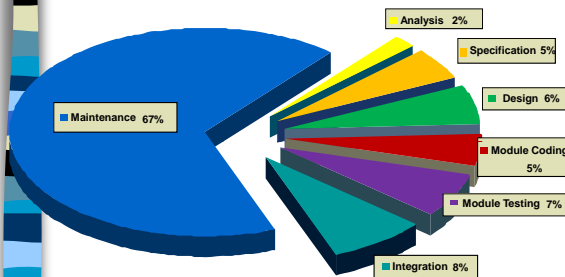
Software Costs

- Software costs often dominate system costs.
 - In other words..
 - The costs of software 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

Lecture Notes 1

32

High Cost



Lecture Notes 1

[Schach]

33

What do we mean by HUGE?

- Thousands of programmers
- Millions of lines of code
- Daily builds and regression testing
- More testers than programmers
- Needs to work on many machines
 - Cross platform
- Needs to be backwards compatible

Lecture Notes 1

34

The Five P's of Software Engineering

- People – those who develop, manage, and run the software
- Product - the software itself
- Project - the activity of creating the software
- Process - the manner in which the project proceeds
- Professionalism - the attitude of all involved

Lecture Notes 1

35

Software Engineering is not programming

Small project	Huge project
You	Teams
Build what you want	Build what they want
One product	Family of products
Few sequential changes	Many parallel changes
Short-lived	Long-lived
Cheap	Costly
Small consequences	Large consequences



Lecture Notes 1

36

Take a break!

- Stretch, Relax
- Get some water, Use the restroom
- Get to know your classmates...
- Etc.....

When we return...

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

Lecture Notes 1 37

From Programming to Engineering

- **People**
 - Someone has get it done (generally many someones)
 - 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

People + Processes + Tools ⇒ Product

Lecture Notes 1 38

People

- The **single** most important factor in the **success/failure** of a product
 - Quality
 - Suitability
 - Cost
- People with these characteristics can be scarce
- Many different kinds of people
 - Managers
 - Programmers
 - Technical writers
 - Testers

Lecture Notes 1 39

Processes

- **Essential** to achieve a **quality** product
- **Need these qualities (not always available)**
 - Quality
 - Suitability
 - Cost
- **Time is also scarce resource**
- **Many different kinds of processes**
 - Bug tracking
 - Change approval
 - Quality assurance

Lecture Notes 1 40

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

Lecture Notes 1 41

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

Lecture Notes 1 42

People, Processes, Tools, Products

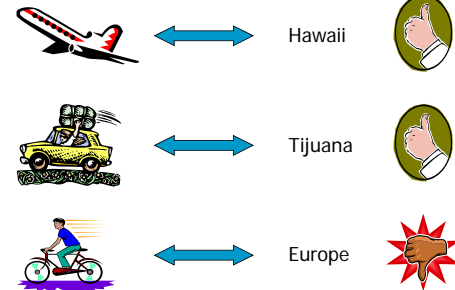
- **Products are always the eventual goal**
 - Selling products creates revenue
 - Selling good products creates lots of revenue
 - Selling bad products creates little revenue
- **People, processes, and tools are retained by organization**
 - Build a reputation through the quality of products
 - Create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

People + Processes + Tools ⇒ Product

Lecture Notes 1

43

Choose the Right Solution

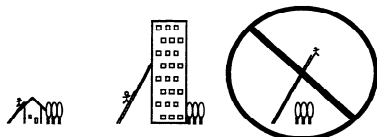


Lecture Notes 1

44

Matters of Scale

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



- **Need to choose the appropriate technique/tool for the job.**

- elephant gun to kill a fly?
- fly-swatter to ward off an elephant?

Lecture Notes 1

45

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]**

Lecture Notes 1

46

Typical Activities in the Lifecycle

- **Requirements**
 - The User's Needs
 - Eliciting requirements from the Client
 - Interviews, Surveys
 - Prototyping
 - Usability testing... etc..
- **Specification**
 - Defining “What the system will do”
 - Articulation of the requirements
 - Informal → Formal
- **Design**
 - Abstract Representation of the system
 - Defining the structure of the system
 - Architecture (High Level) → Components (Low Level)
 - How do the components connect?
 - Design Patterns

47

Typical Activities in the Lifecycle

- **Testing**
 - Verification: “Have we built the system right”
 - Validation: “Have we built the right system”
 - Perpetual Testing
 - Different Levels of Testing
 - System
 - Integration
 - Unit
 - Regression
- **Maintenance**
 - Configuration management / Change Management

48

Software Life Cycle Models

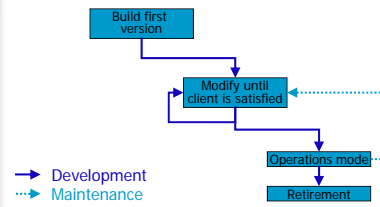
- Build-and-fix
- Waterfall
- Rapid prototyping
- Incremental
- Spiral

A software life cycle model is a high-level process

Lecture Notes 1

49

Build-and-Fix

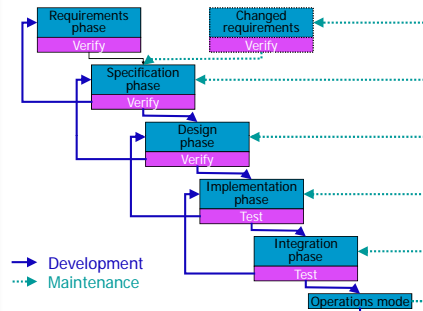


Some argue if this is really a lifecycle model... it is a bad one at best. What about our basic activities?

Lecture Notes 1

50

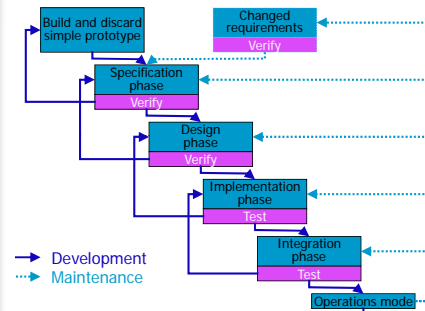
Waterfall [Royce , 1970s]



Lecture Notes 1

51

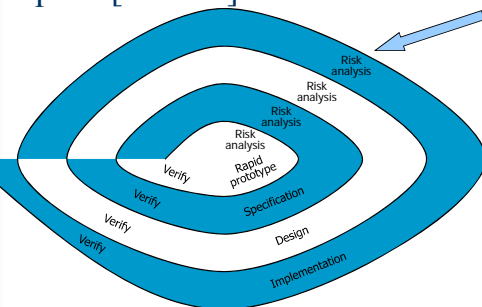
Rapid Prototyping



Lecture Notes 1

52

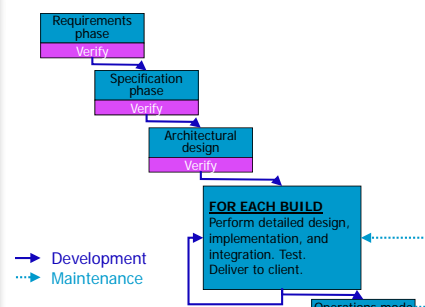
Spiral [Boehm]



Lecture Notes 1

53

Incremental [Mills]



Lecture Notes 1

54

Moving on...

- **Methods and Tools**

Lecture Notes 1

55

Tools Most Students Have Used:

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

Lecture Notes 1

56

Problem – Approach Doesn't Scale

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

Lecture Notes 1

57

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

Lecture Notes 1

58

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?

Lecture Notes 1

59

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

Lecture Notes 1

60

Tools

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

Lecture Notes 1

61

Methods

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

Lecture Notes 1

62

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

Lecture Notes 1

63

Example: Guitar



- Strum
- Pick ... etc

Lecture Notes 1

64

```
#1          #2
3/4 G   Am7  G   4/4 G
E|-----|-----|-----|-----|
B|0-----1-----3-----|12-----12-----12-----12-----|
G|0-----0-----0-----|0-----0-----0-----0-----|
D|-----|-----|-----|-----|
A|0-----0-----2-----|10-----10-----10-----10-----|
E|-3-----|-----|-----|-----|
+ . . . + . . . + . . . + . . .
```

```
3/4 G   Am7  G   4/4 G
E|-----|-----|-----|-----|
B|0-----1-----3-----|12-----12-----12-----12-----|
G|0-----0-----0-----|0-----0-----0-----0-----|
D|-----|-----|-----|-----|
A|0-----0-----2-----|10-----10-----10-----10-----|
E|-3-----|-----|-----|-----|
+ . . . + . . . + . . . + . . .
1. Blackbird singing in the dead of night
2. Blackbird singing in the dead of night
```

Lecture Notes 1

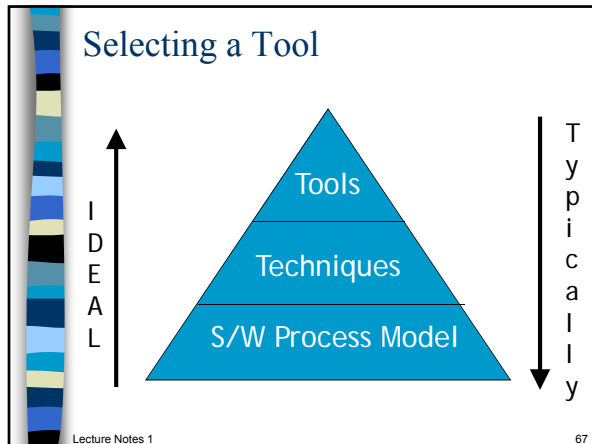
65

Analogy

- Guitar => Tool
- How To Play => Method
- Tabs => Notation

Lecture Notes 1

66



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

Lecture Notes 1 68

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
 - (hmmm what does that button do?)
 - How do I get it to help me
 - Hmmm what does this error message mean....
 - Sheesh --- this is just slowing me down
- Uncertain payoff

Lecture Notes 1 69

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

Lecture Notes 1 70

Personal Practices & 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

Lecture Notes 1 71