

CT5805701

營建資訊系統之軟體工程

Software Engineering in Construction Information Systems

BlackBoard: elearning.ntust.edu.tw

Time: 1:30AM – 4:20PM, Thursday

Classroom: IB-505

Lecturer: Yo-Ming Hsieh (謝佑明)

ymhsieh@mail.ntust.edu.tw

1

What is Software Engineering?

- Google: *define: software engineering*

1. The system of applying of an engineering discipline to the **design, implementation** and **maintenance** of software systems.
2. A collection of theories, techniques, and tools which enable fallible humans to design, construct and maintain **large software products** in a reliable and cost effective manner.

- Google: *define: system*

A system is an assemblage of inter-related elements comprising a unified whole. From the Latin and Greek, the term "system" meant to combine, to set up, to place together. A sub-system is a system which is part of another system. A system typically consists of components (or elements) which are connected together in order to facilitate the flow of information, matter or energy. ...

3

Discussions

- What is software ?
 - Coded instructions (programs) that make a computer do useful work.
 - The programs and instructions which direct a computer.
 - Programs written by programmers using computer languages.

2

Outline of this course

Part I. Development Methods

- Software development process
- Time and resource estimation

Part II. Information systems

- Software modeling
- Database technology
- Virtualization technology
- Information system architecture
 - 2-tier applications
 - 3-tier web applications
- Web services
- Internet security

4

Grading

- Assignments: 60 pts
 - The assignments will go through a complete cycle of software development.
 - The assignments will build *a platform (for selling or exchanging something)* with
 - 2-tier architecture (application + database)
 - 3-tier architecture (web client + web server + database)
 - Coding is minimal!!
 - We will create real applications by using modern technology, so most codes are generated for us.
 - Assignments late within a week get a 10% reduction in its grade.
 - Assignments late for more than a week will NOT be accepted.
- Final Exam: 30 pts
- Class participation: 10 pts

5

Lecture 1

Review of Programming Fundamentals

7

Reading Schedule

- Rapid Development: Taming Wild Software Schedules, Steve McConnell, Microsoft Press.
 - Week 1: Chap. 2 – 4.
 - Week 2: Chap. 6 – 7.
 - Week 3: Chap. 8 – 9.
- The first 2-3 lectures are based on the content of above chapters.
- You're encouraged to read the rest of this book. It is a fun reading.

6

Today's Topic

- The process of software development
- Large-scale software projects
- Rapid software development strategy
 - Four general strategies
 - Four dimensions of development

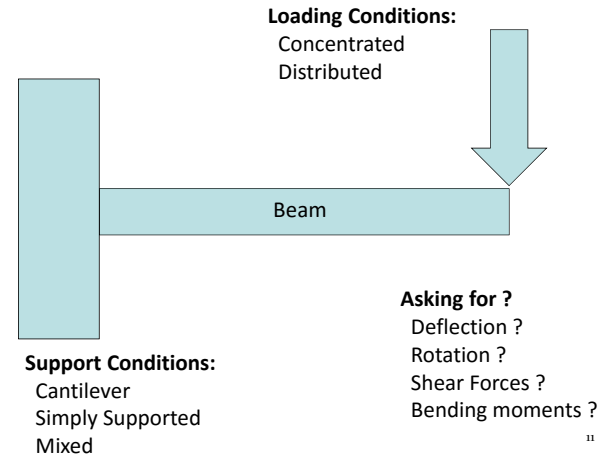
8

The process of small-scale software development

1. Requirement specification
2. Analysis
3. Design
4. Implementation
5. Testing and verification

It should be noted that the process of software development is almost the same as **problem-solving**.

9



The process of software development (1) Requirement Specification

- Understand **EXACTLY** what the problem is
- Understand what is needed to solve it
- What the solution should provide
- Constraints and special conditions

- How precisely we can define a problem depends on our degree of familiarity with the problem domain.
- If we are not familiar with the problem, we should either quickly acquire an education in it or contact people who are knowledgeable about it.
- **This step is key to success of software projects!**

10

What questions should you ask ?

Buy a meal at McDonald

Build a computer

Write a software

12

The process of software development (2) Analysis

- Identify the following:
 - Input
 - Output
 - Special constraints or conditions
 - e.g. the process from input to output must be less than 1ms.
 - e.g. the maximum RAM consumed must be less than 8kB.
 - Formulas and equations to be used

13

Output ?

Beam section size b & h

Formula ?

$$w(x) = -\frac{Px^2(3L-x)}{6EI} \quad I = \frac{1}{12}bh^3$$

Input ?

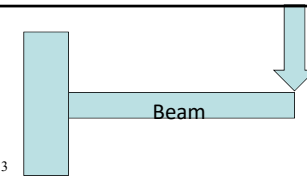
x, P, L, E

Special Constraints and Conditions?

(The equation applies for elastic material behavior and homogeneous property) !!

The maximum deflection be less than 1mm.

The material has to be iron



15

Output ?

Deflection at end point

Formula ?

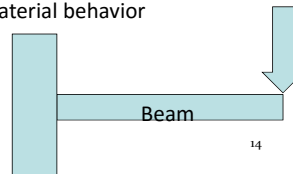
$$w(x) = -\frac{Px^2(3L-x)}{6EI} \quad I = \frac{1}{12}bh^3$$

Input ?

x, P, L, E, I, b, h

Special Constraints and Conditions?

(The equation applies for elastic material behavior and homogeneous property) !!



14

The process of software development (3) Design

- Algorithm: the method of solution
 - An algorithm is a sequence of a finite number of steps arranged in a specific logical order, which, when executed, produce the solution for a problem.
 - An algorithm must satisfy some requirements:
 - Unambiguousness
 - Generality
 - Correctness
 - Finiteness
- Presentation of algorithms: “pseudo coding” or flowcharting

16

Output ?
Beam section size b & h

Formula ?

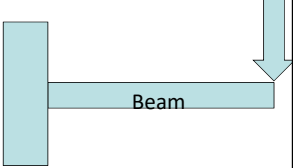
$$w(x) = \frac{PL^3(3L-x)}{3EI6EI} \quad I = \frac{1}{12}bh^3$$

Input ?
 $P, L, E \quad P = 100, L = 20$

Special Constraints and Conditions?
 The maximum deflection be less than 3mm.
 The material has to be iron

For E in (all possible E values for iron)
 Computed required I using ...
 For b in all possible width
 Compute h using ...
 Output b, h, and E
 End For
 End For

$I = \frac{PL^3}{3 \times E \times 0.003}$
 $h^3 = \frac{12I}{b}$



17

Pseudocode

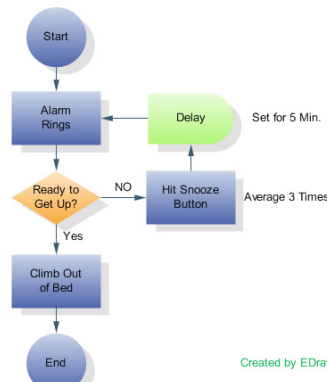
- Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.
- It uses the structural conventions of a programming language, but is **intended for human reading** rather than machine reading.
- Pseudocode typically **omits details** that are essential for machine understanding of the algorithm, such as variable declarations, system-specific code and some subroutines.
- The programming language is augmented with natural language description details, where convenient, or with compact mathematical notation.
- The purpose of using pseudocode is that it is easier **for people to understand** than conventional programming language code, and that it is an efficient and **environment-independent** description of the key principles of an algorithm.

19

<https://en.wikipedia.org/wiki/Pseudocode>

Flowchart

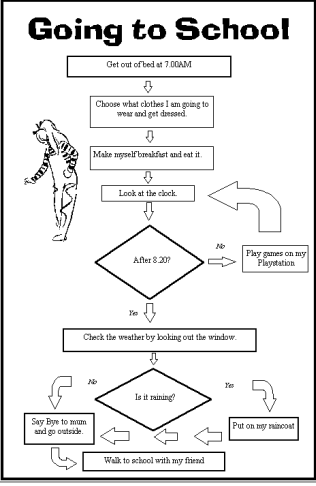
- Three basic symbols
 - Square: activities or tasks
 - Diamond: decision points
 - Arrows: flow of control
- Basic principles
 - Tasks have only one exit.
 - Tasks may have one or more entries.
 - Decisions have two or more exits. Each exit arrow should label the criterion.



Created by EDraw

<http://www.edrawsoft.com/images/examples/Process-Flowchart.png>

Going to School



20

http://english.unitecnology.ac.nz/resources/units/pathways/flow_chart.gif

Pseudo Code Example (1/2)

Main Procedure Monopoly_Game

Hand out each player's initial money.

Decide which player goes first.

Repeat

 Call Procedure Monopoly_Move for next player.

 Decide if this player must drop out.

Until all players except one have dropped out.

Declare the surviving player to be the winner.

21

<http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

The process of software development (4) Implementation

- Translate each step of the algorithm into statements in that particular language and end up with a computer program.
- Programming errors
 - design errors
 - syntax errors
 - run-time errors

23

Pseudo Code Example (2/2)

Procedure Monopoly_Move

Begin one's move.

Throw the dice.

Move the number of spaces on the board shown on the dice.

If the token landed on "Go to Jail,"

 then go there immediately.

Else if the token landed on "Chance" or "Community Chest,"

 then draw a card and follow its instructions.

Else

 follow the usual rules for the square (buying property,

 paying rent, collecting \$200 for passing "Go", etc.).

End one's move.

22

<http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/pseudo.htm>

The process of software development (5) Testing and Verification

- Testing
 - the process of executing a program to demonstrate its **correctness**
- Verification
 - the process of ensuring that a program **meets user requirements**

24

Summary

The process of software development / problem solving

Large-scale software projects

Rapid software development strategy (for large-scale software projects)

Four general strategies

Four dimensions of development

25

A good software project

- Correct
 - Meets user requirements
- On schedule
- Cost efficient

27

Large-scale software projects

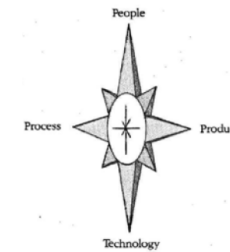
- Windows NT 4.0
 - <http://www.cnn.com/2004/TECH/biztech/02/13/microsoft.source/>
 - 28 millions lines of code, 95103 files
- Windows 2000
 - <http://www.dwheeler.com/sloc/>
 - In 2001: 35 million lines of code
 - Windows XP: 40 million lines of code
 - Windows Vista: 50 million lines of code
 - Windows 7: 40 million lines of code (team size: 2000 – 2500)
- Redhat Linux 7.1
 - <http://www.dwheeler.com/sloc/>
 - >30 million lines of code

<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>
<http://arstechnica.com/information-technology/2008/08/2500-windows-7-employees-broken-up-into-25-feature-teams/>

26

Four Dimensions of Development

People
Process
Product
Technology



28

People Peopleware

- Peopleware may have more impact on software **productivity** and **quality**
 - Productivity with similar levels of experience varies by a factor of at least **10 to 1**.
 - The performance of entire teams varies on the order of **3, 4, or 5 to 1**.
 - “**technology is not the answer**; the most effective practices are those that leverage the human potential of their developers” ~ Basili et al. 1995
- Look into: staff selection, motivation, teamwork, and training

29

People Staff Selection (2/2)

- **Team balance** – select people who will complement and harmonize with each other
- **Misfit elimination** – eliminate and replace problem team members as quickly as possible
- **Other factors** – design ability, programming ability, programming-language experience, machine and environment experience, and application-area experience

31

People Staff Selection (1/2)

- **Top talent** – use better and fewer people
- **Job matching** – fit the task to the skills and motivation of the people available
- **Career progression** – help people to self-actualize than forcing them to work where they have the most experience or where they are most needed

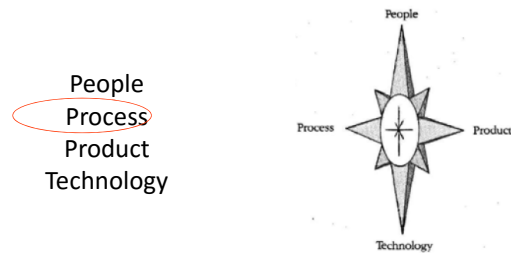
30

People Team Organization & Motivation

- Tailor teams to match
 - Project size
 - Schedule goals
 - Product attributes
- Motivation
 - No factor other than **motivation** will cause a person to forsake evenings and weekends without being asked to do so.

32

Four Dimensions of Development



Process (2/3)

- Rework avoidance
 - Avoid requirement change at late stages
- Quality Assurance (Q/A)
 - Assure the product has an acceptable level of quality
 - Detect errors at the stage when they are least time-consuming (and least costly) to correct. → catch errors as close as possible to the time that they are introduced
- Development fundamentals
 - Analysis → design → construction → integration → testing will not product lightning-fast schedules, but they **prevent disasters**
 - Half of the challenge of rapid development is **avoiding disaster**.³⁵

Process (1/3)

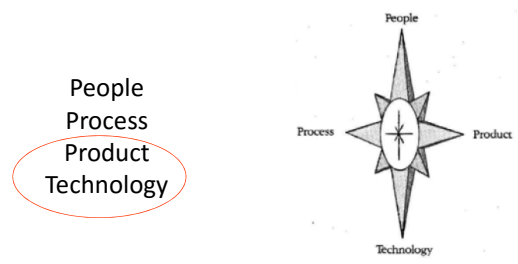
- Management and Technical Methodologies
- Process represents an area of **high leverage** in improving development speed
- Hughes aircraft, Lockheed, Motorola, NASA, Raytheon, and Xerox find by focusing on improving their development process
 - Cut time-to-market by about one-half
 - Reduced cost and defects by factors of 3 to 10

34

Process (3/3)

- Risk management
- Resource Targeting
 - Get the most bang for your buck
- Lifecycle planning
 - Several lifecycle models to be introduced later
- Customer orientation
 - Develop software to its spec. is half job done
 - The other half is help the customer figure out what the product should be (Thus, requirement specification is very important).³⁶

Four Dimensions of Development



37

Technology

- Platform, operation system, software components, development tools, ...
- From assembly → high-level languages was one of the most influential changes in software-development history
- Integrated Development Environment, IDE
 - includes Source code editor, compiler, debugger, profiler, ...
 - Microsoft Visual Studio, Eclipse, Borland C++ builder, ...
 - helps manage the complexity of software projects when there are many classes and methods.
- Visual Programming
 - programming can be done by drag-and-drop without writing a single line of code
 - e.g. Scratch: <http://scratch.mit.edu>

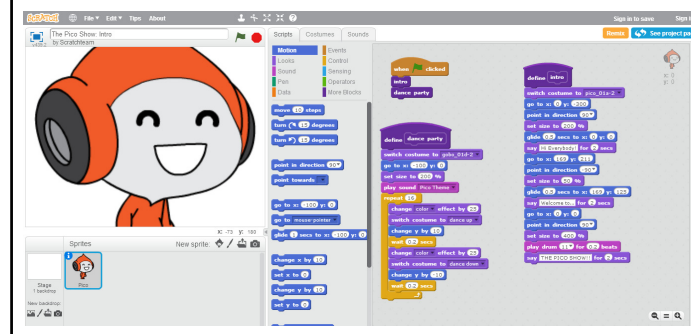
39

Product

- The most tangible dimension
 - product size
 - 80/20 rule
 - Additional features require additional specification, design, construction, testing, and integration
 - 1/2 produce size → 2/3 effort saving
 - product characteristics
 - Performance, memory footprint, robustness, reliability
 - Don't insist on too many priorities at once!

38

Scratch



Example: [the Pico show](#)

40

Summary for 4 dimensions

People

Process

Product

Technology

41

Summary

- The process of small-scale software development
- Large-scale software projects
- Rapid software development strategy
 - Four general strategies
 - Four dimensions of development

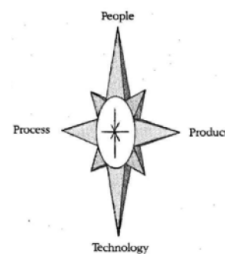
43

Which dimension matters the most

- Different projects have different needs
 - Accept the limitations on the dimensions you cannot change
 - Emphasize the other dimensions to get the rest of the schedule benefit you need

Examples

- Fuel-injection system for a car
- In-house business program
- A feature-driven shrink-wrap market



42

Assignment #0

- This is one and the only assignment on individual basis.
- You're about to write a program for calculating taxi fare in Taipei.
 - Minimum fare: NT \$70, and it covers the first 1.25km. Afterwards, NT \$5 for every additional 250m above the initial 1.25km.
 - Lag charges: NT \$5 for every 100 seconds that the car speed goes below 5 km/hour.
 - Nighttime service charge: NT \$20 additional if you get in the car between 11:00pm and 6:00am
 - Lunar New Year service charge: NT \$20 additional if you travel between two days before the Lunar New Year and the end of the Lunar New Year.

44

<http://www.pto.gov.taipei/ct.asp?xItem=1065628&ctNode=12593&mp=117041>

Assignment #0

- Here are some examples (which can be seen as test cases):
 - Travelling 15km with 80 seconds lag time during day time during non-Lunar New Year period: NT \$345.
 - Travelling 15km with 240 seconds lag time during day time during non-Lunar New Year period: NT \$355.
 - Travelling 15km with 120 seconds lag time during nighttime during non-Lunar New Year period: NT \$370.
 - Travelling 15km with 360 seconds lag time during nighttime during Lunar New Year period: NT \$400.
- In this assignment, you're required to document the following steps in your software development process: 1) require specification, 2) analysis, 3) design (draw flowcharts or write pseudocode).
- You can continue to 4) implementation (using your preferred computer language) and 5) testing & verification to get **extra credit** for this assignment.

45

Assignment #0

- Rapid Development: Taming Wild Software Schedules, Steve McConnell, Microsoft Press.
 - Week 1: Chap. 2 – 4.
 - Week 2: Chap. 6 – 7.
 - Week 3: Chap. 8 – 9.
- Nothing to turn in, but content of these chapters may appear in the final exam.

46

47

48