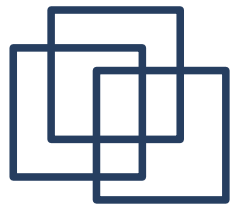


# CMSC 128

---

## Introduction to Software Engineering Second Semester AY 2007-2008

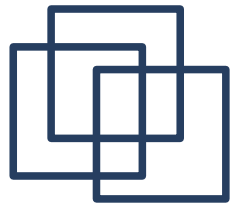
[jachermocilla@uplb.edu.ph](mailto:jachermocilla@uplb.edu.ph)



# Software Project Planning

---

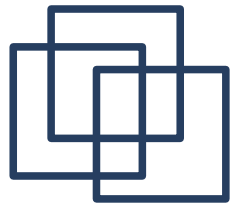
- Project planning marks the start of the project management process
- Estimation is the first activity in project planning
- When estimates are made, we look into the future and accept some degree of uncertainty
- Estimation is both science and art
  - Need not be conducted haphazardly



# Software Project Planning

---

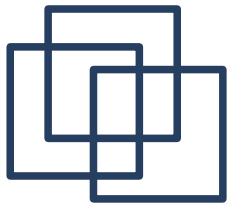
- What to estimate?
  - resources, cost, and schedule
- Factors affecting estimation
  - Project complexity
  - Project size
  - Structural uncertainty
  - Availability of historical data



# Software Project Planning

---

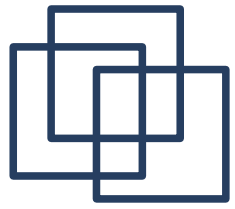
- Objective
  - Provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule within a limited time frame at the beginning of the project.
  - Define best-case and worst-case scenarios so that outcomes can be bounded



# Software Scope

---

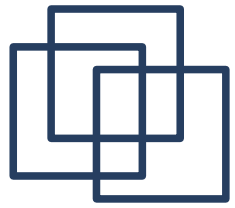
- Function and performance allocated to software during system engineering must be assessed to establish a project scope
- Software scope describes function, performance, constraints, interfaces, and reliability
- Software scope must be unambiguous and understandable at management and technical levels



# Obtaining Info for Scope

---

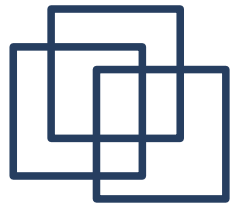
- Must communicate with customer
- First set of context-free questions
  - Who is behind the request for this work?
  - Who will use the solution?
  - What will be the economic benefit of a successful solution?
  - Is there another source for the solution?



# Obtaining Info for Scope

---

- Second set of context-free questions
  - How would you characterize good output of the successful solution?
  - What problems will this solution address?
  - Can you show me the environment in which the solution will be used?
  - Are there special performance issues that will affect the way solution is approached?

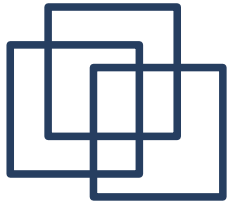


# Obtaining Info for Scope

---

- Third set of context-free questions
  - Are your answers official?
  - Are my questions relevant to the problem that you have?
  - Am I asking too many questions?
  - Is there anyone else who can provide additional information?
  - Is there anything else that I should be asking you?

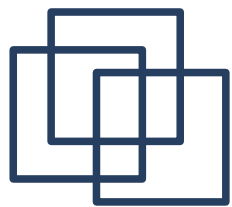




# Resources

---

- After defining the scope, resources must be estimated
- Each resource is characterized by
  - Description
  - Statement of availability
  - Chronological time that the resource will be required
  - Duration of time that the resource will be applied

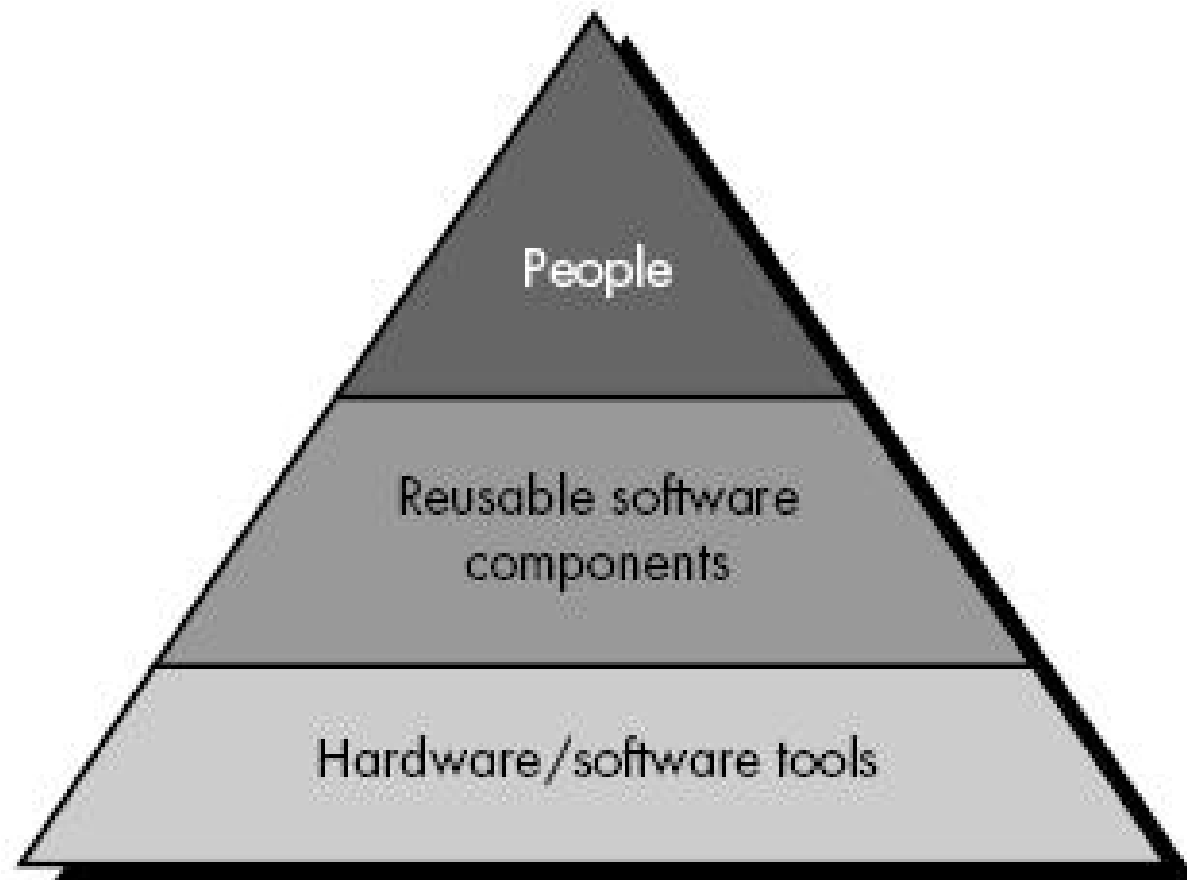


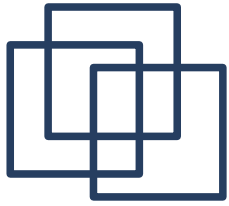
# Resources

---

**FIGURE 5.2**

Project  
resources

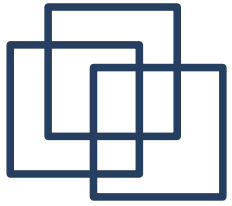




# Resources

---

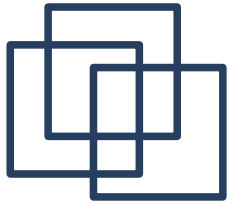
- Human Resources
  - Skills required are identified
  - Both organizational(ex. manager) and specialty(ex. DBA) positions are specified
- Reusable Resources
  - Off-the-shelf components
  - Full-experience components
  - Partial-experience components
  - New components



# Resources

---

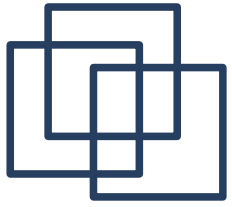
- Environmental Resources
  - Supports the project: both hardware and software
  - Software Engineering Environment
  - Development Environment



# Estimation

---

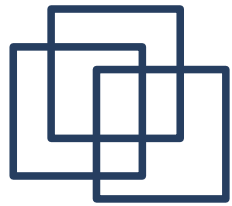
- Options to obtain reliable estimates
  - Delay estimation until late in the project
  - Base estimates on similar projects that have already been completed
  - Use decomposition techniques
  - Use one or more empirical models for software cost and effort estimation



# Decomposition

---

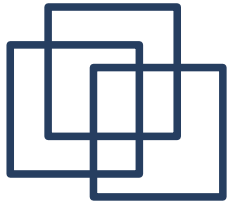
- Can use problem or process decomposition
- Understand the scope and generate an estimate of 'size'



# Software Sizing

---

- Factors affecting project estimates
  - Estimate of the size of product
  - Ability to translate size estimate into effort, time, and cost
  - Degree to which project plan reflects the ability of the team
  - Stability of requirements and SEE
- In project planning, size refers to the quantifiable outcome of the software project (LOC or FP)

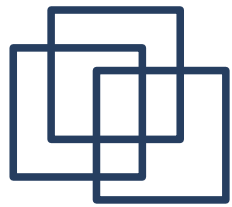


# Software Sizing

---

- Approaches
  - 'Fuzzy-logic' sizing
  - Function point Sizing
  - Standard component sizing
    - Some software have 'standard components'
      - Modules, reports, screens, etc
  - Change sizing
    - When using existing software that must be modified
    - Estimates the number and type of modifications(adding, deleting, reuse, etc.)

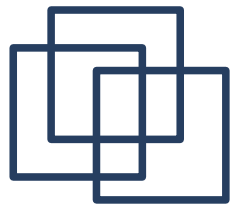




# Problem-based Estimation

---

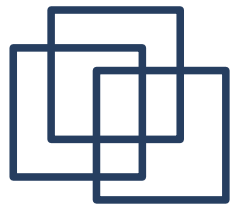
- Use of LOC and FP in estimation
  - *Estimation variable* that is used to 'size' each element of software
  - *Baseline metrics* collected from past projects and used together with estimation variables to develop cost and effort projections
- Planner begins with bounded statement of software scope, then decompose software into problem functions that can each be estimated individually using LOC or FP



# Problem-based Estimation

---

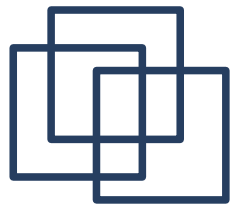
- Baseline productivity metrics (LOC/pm or FP/pm) are applied to the estimation variable and cost or effort for the function is derived
- In using LOC as estimation variable, decomposition is essential to obtain more accurate estimates
- In using FP as estimation variable, countable information domain and complexity adjustment values are used



# Problem-based Estimation

---

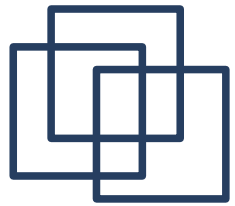
- When estimating (based on historical data or intuition), specify optimistic, pessimistic, and most likely size
- Three point value or expected value is computed
  - $EV = (opt + 4*most\_likely + pess)/6$



# Process-based Estimation

---

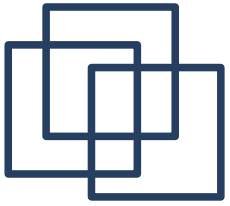
- Identify framework activities and tasks for the selected process then estimate the effort required for each
  - Unit of effort is person-month, the amount of work that a person can do in one month
- Decompose software into problem functions
- Meld the product and the process



# Empirical Models

---

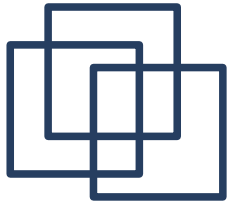
- Use empirically derived formulas to predict *effort* as a function of LOC or FP
- General structure
  - $\text{Effort} = A + B \times (\text{estimation var})^C$ , where A, B, and C are empirically derived constants
- Examples
  - Walston-Felix,  $E = 5.2 \times \text{KLOC}^{0.91}$
  - Bailey-Basili,  $E = 5.5 \times 0.73 \times \text{KLOC}^{1.16}$



# COCOMO

---

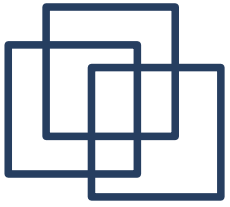
- COnstructive COst MOdel, by Boehm
- Hierarchy of models
  - Model 1 - basic
    - Effort(and cost) as function of size (LOC)
  - Model 2 - intermediate
    - Same as Model 1 but includes 'cost drivers'
  - Model 3 – advanced
    - Same as Model 2 but includes assessment of cost drivers impact on each step



# COCOMO

---

- Three classes of software projects
  - Organic
    - Relatively small, small teams, less rigid requirements
  - Semi-detached
    - Intermediate in size and complexity
    - Teams with mixed experience
  - Embedded
    - Tight constraints

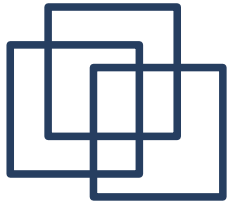


# COCOMO

---

- Basic
  - $E = a * KLOC^b$ ,  $D = c * E^d$
- Semi-detached
  - $E = a * KLOC^b \times \text{Effort\_Adjustment\_Factor}$

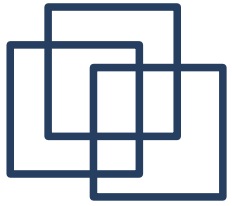




# Software Equation

---

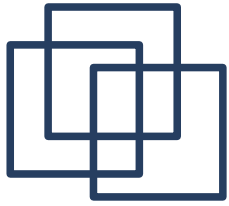
- $E = [\text{LOC} \times B^{0.333} / P]^3 \times (1/t^4)$ 
  - E – effort
  - T – duration in years
  - B – special skills factor, min=0.16, max=0.39
  - P – productivity parameter



# Make or Buy?

---

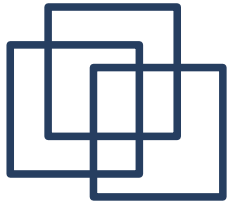
- Guidelines
  - Develop specification of function and performance
  - Estimate internal cost to develop and delivery date
  - Select three or four candidates applications
  - Select reusable components
  - Develop comparison matrix
  - Evaluate
  - Contact other users for opinion



# Summary

---

- Planner must estimate time, effort, and resources (hardware, people, resource)
- Estimation techniques: decomposition and empirical
- Automated tools can be used
- Estimates derived from one technique must be reconciled with other techniques



# Reference

---

- Roger S. Pressman. Software Engineering: A Practitioner's Approach, 4th Ed. McGraw-Hill, 1997. Chapter 5