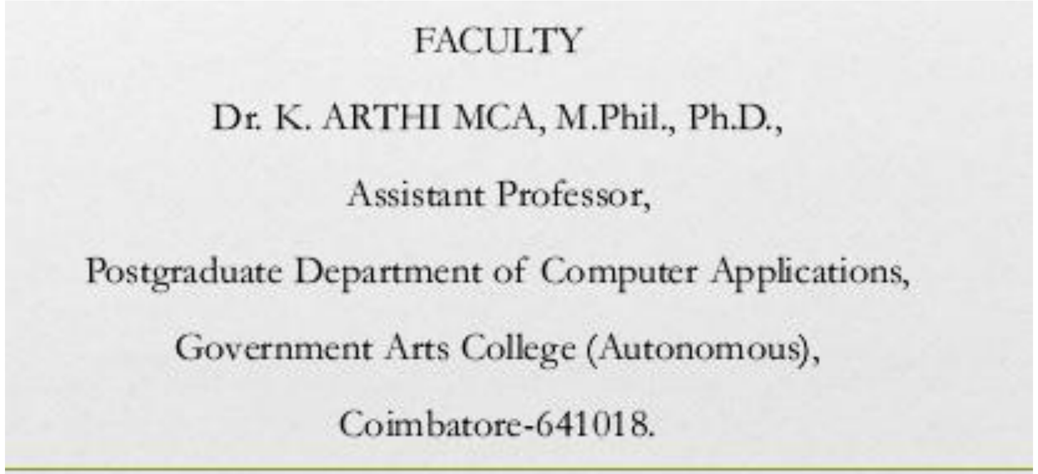
18MCA44E SOFTWARE TESTING

UNIT III – SOFTWARE MAINTENANCE



FACULTY

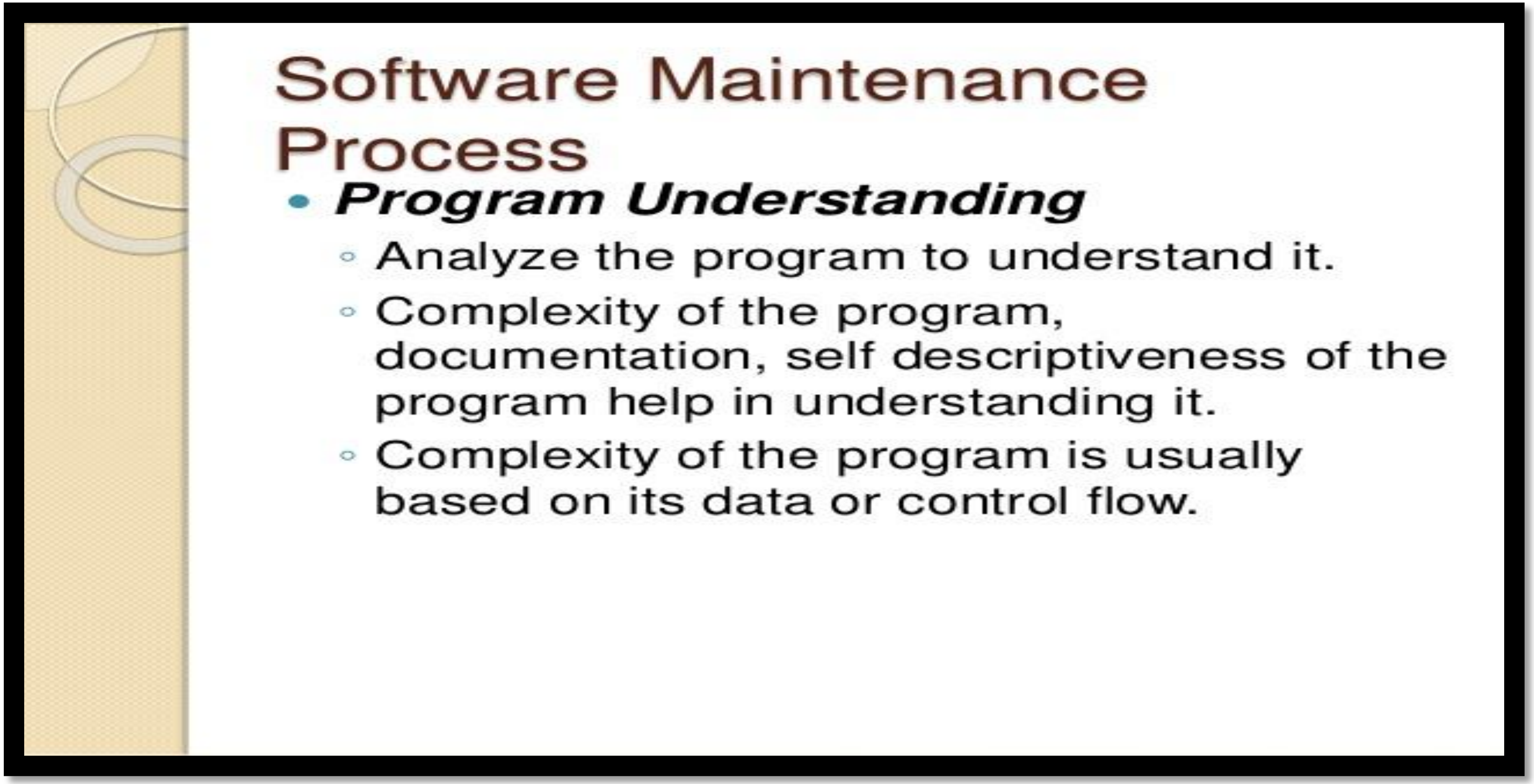
Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

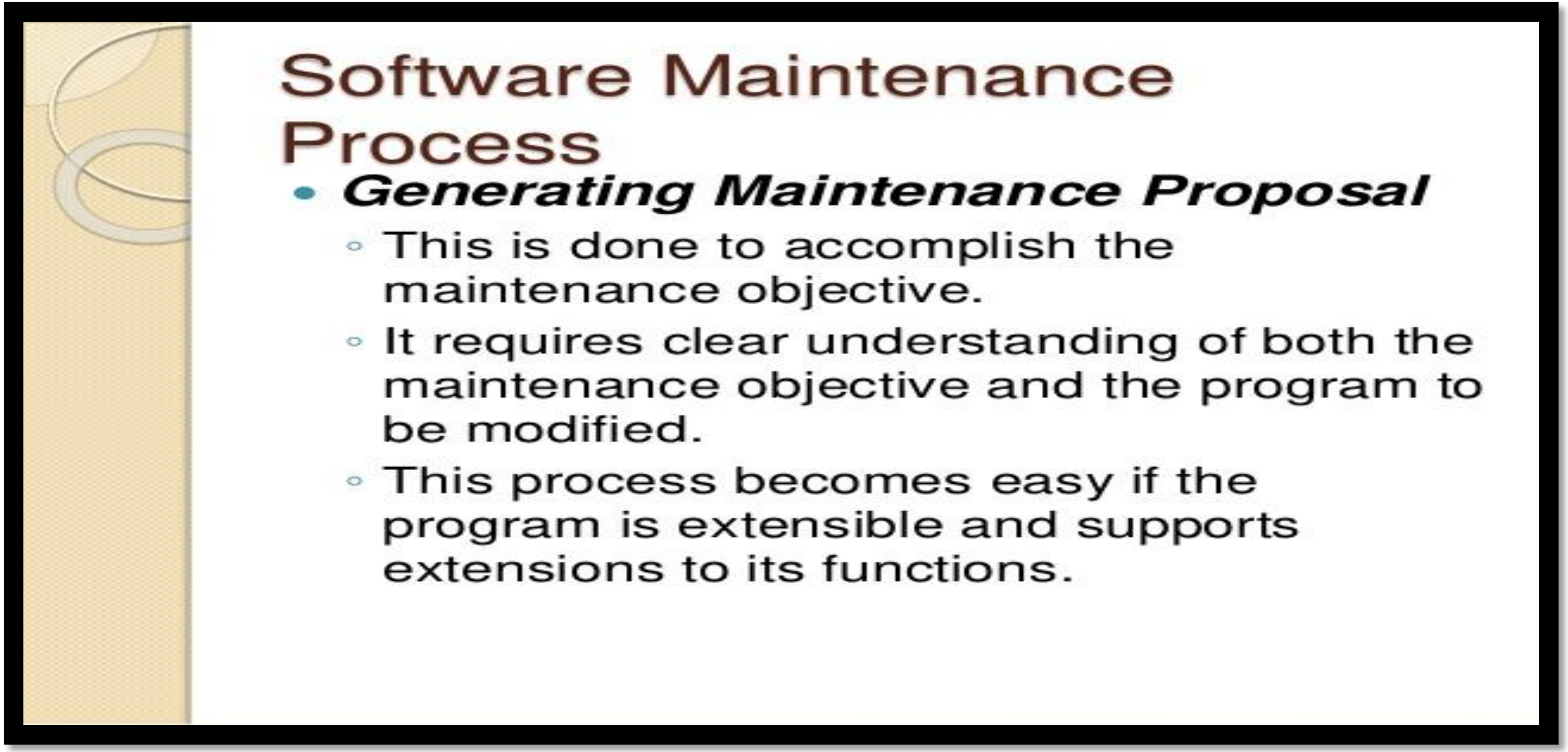
Government Arts College (Autonomous),

Coimbatore-641018.



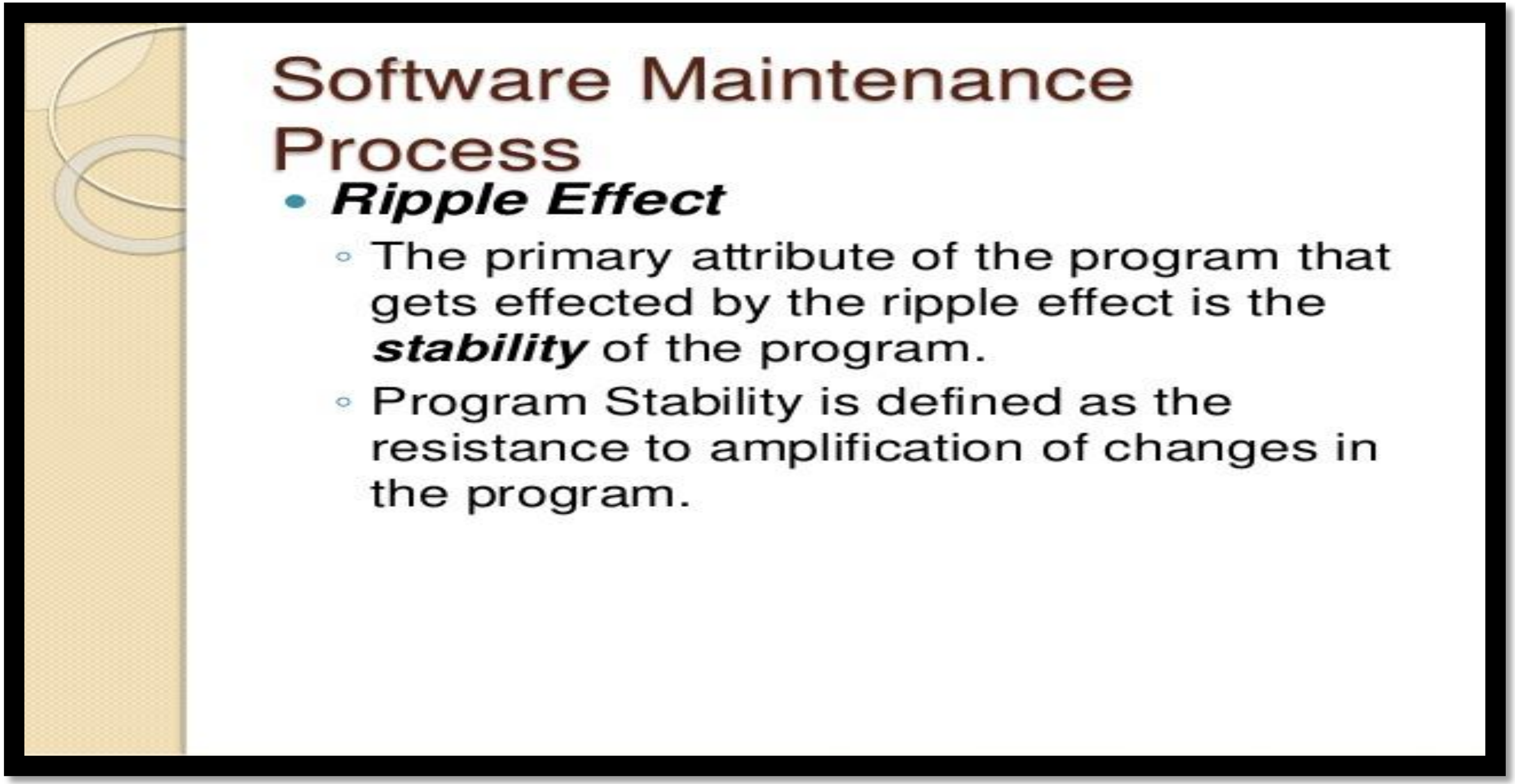
Software Maintenance Process

- ***Program Understanding***
  - Analyze the program to understand it.
  - Complexity of the program, documentation, self descriptiveness of the program help in understanding it.
  - Complexity of the program is usually based on its data or control flow.



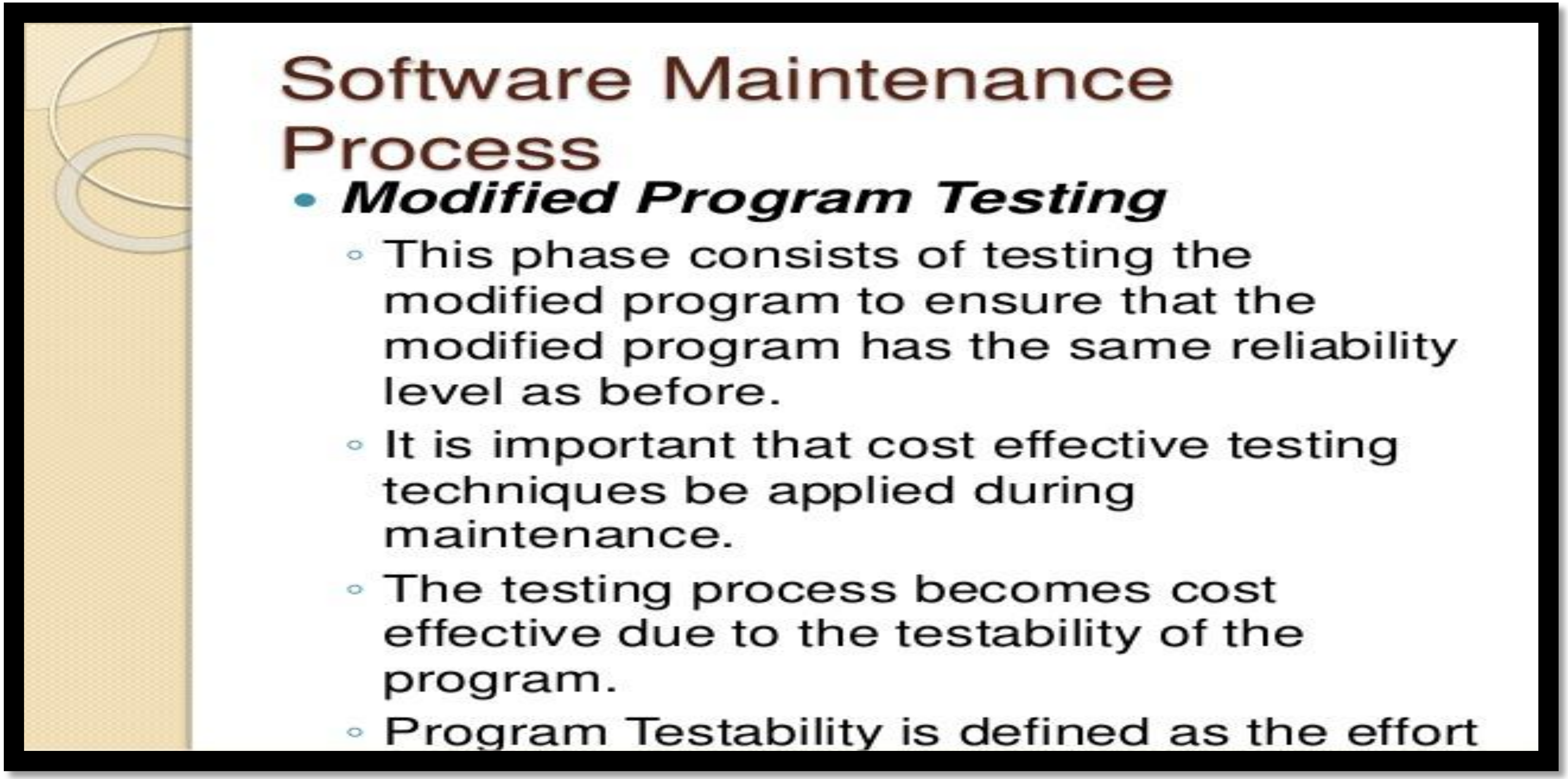
Software Maintenance Process

- ***Generating Maintenance Proposal***
  - This is done to accomplish the maintenance objective.
  - It requires clear understanding of both the maintenance objective and the program to be modified.
  - This process becomes easy if the program is extensible and supports extensions to its functions.



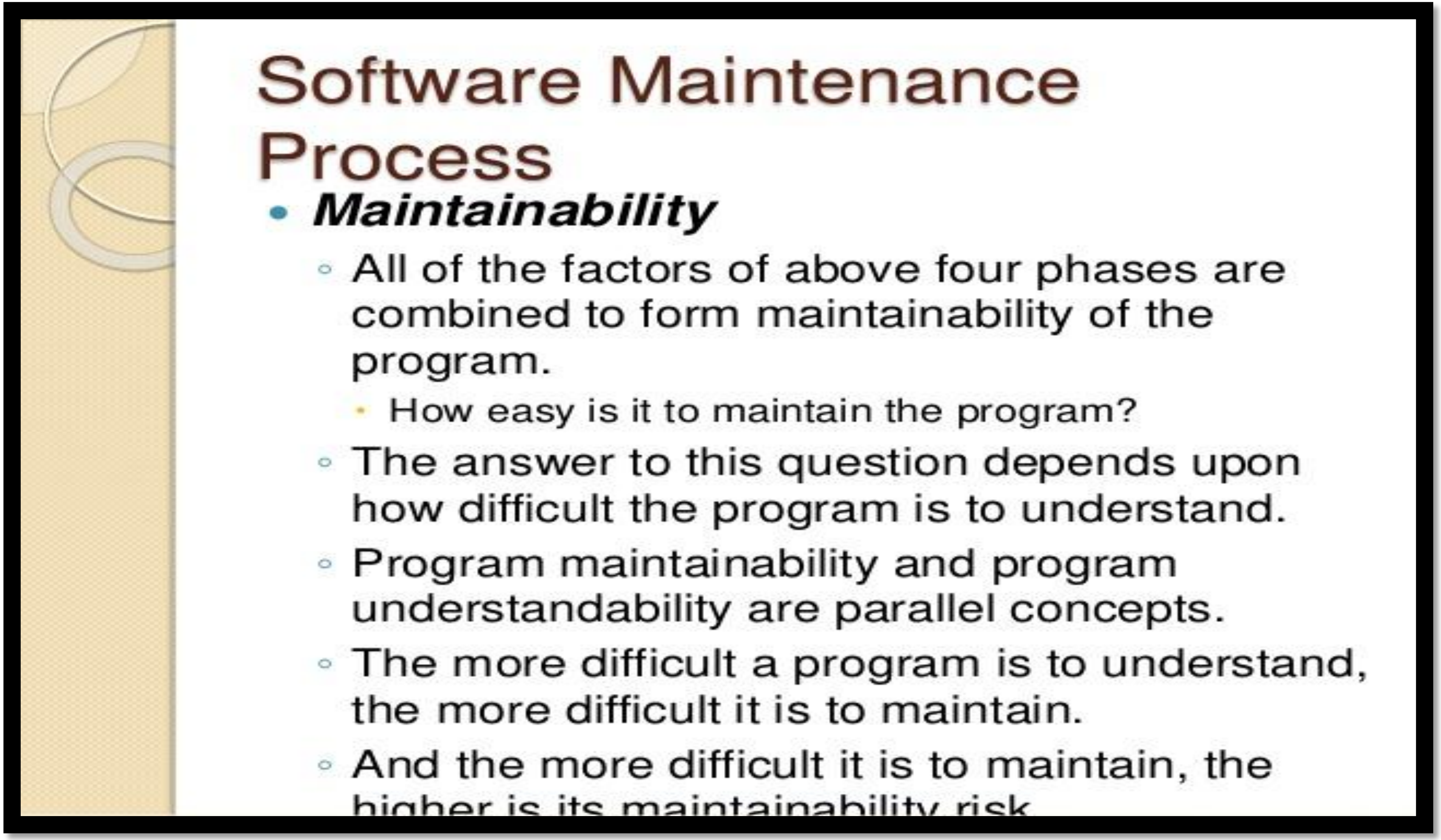
Software Maintenance Process

- ***Ripple Effect***
  - The primary attribute of the program that gets effected by the ripple effect is the ***stability*** of the program.
  - Program Stability is defined as the resistance to amplification of changes in the program.



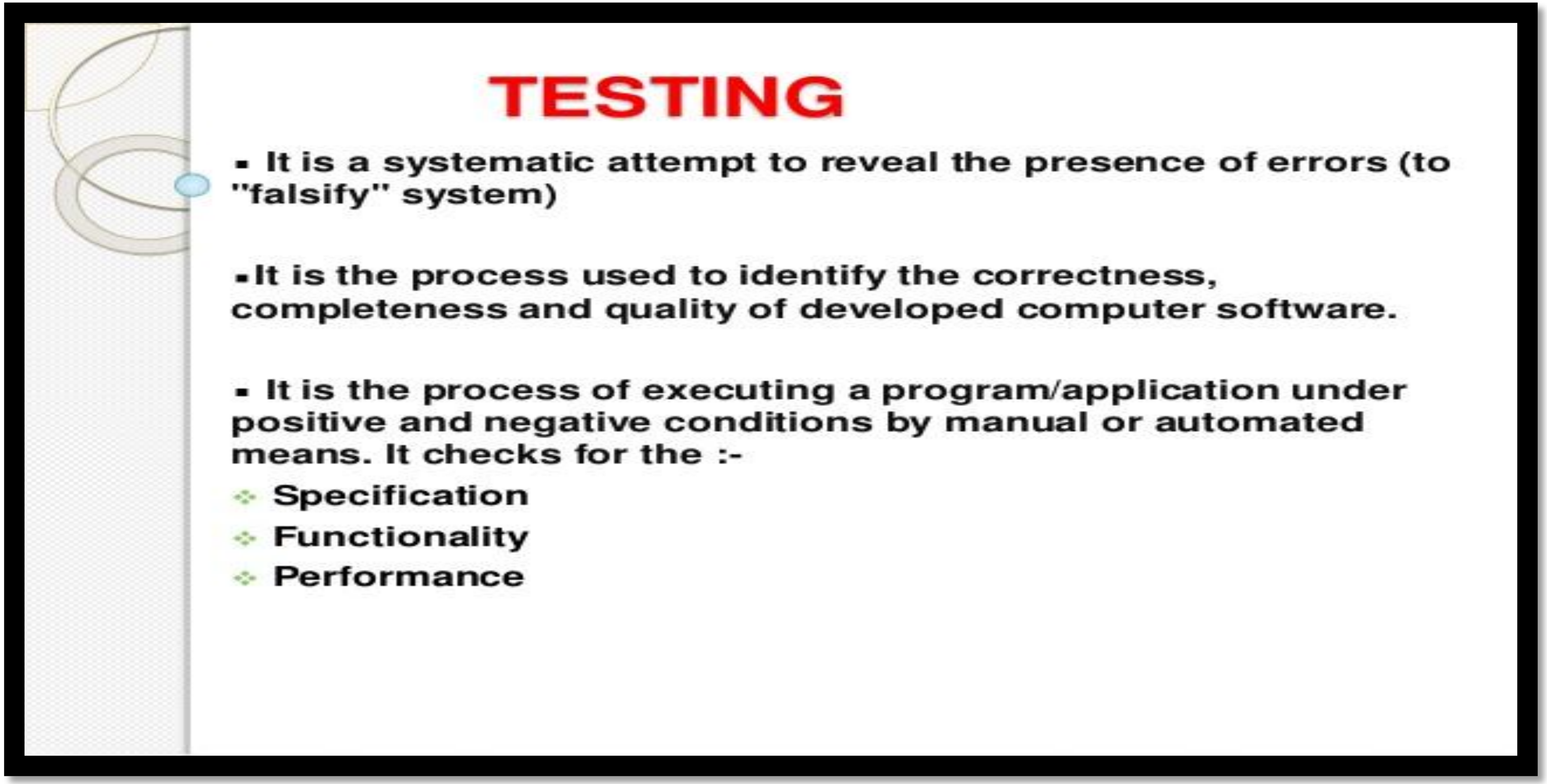
Software Maintenance Process

- ***Modified Program Testing***
  - This phase consists of testing the modified program to ensure that the modified program has the same reliability level as before.
  - It is important that cost effective testing techniques be applied during maintenance.
  - The testing process becomes cost effective due to the testability of the program.
  - Program Testability is defined as the effort



Software Maintenance Process

- ***Maintainability***
  - All of the factors of above four phases are combined to form maintainability of the program.
    - How easy is it to maintain the program?
  - The answer to this question depends upon how difficult the program is to understand.
  - Program maintainability and program understandability are parallel concepts.
  - The more difficult a program is to understand, the more difficult it is to maintain.
  - And the more difficult it is to maintain, the higher is its maintainability risk



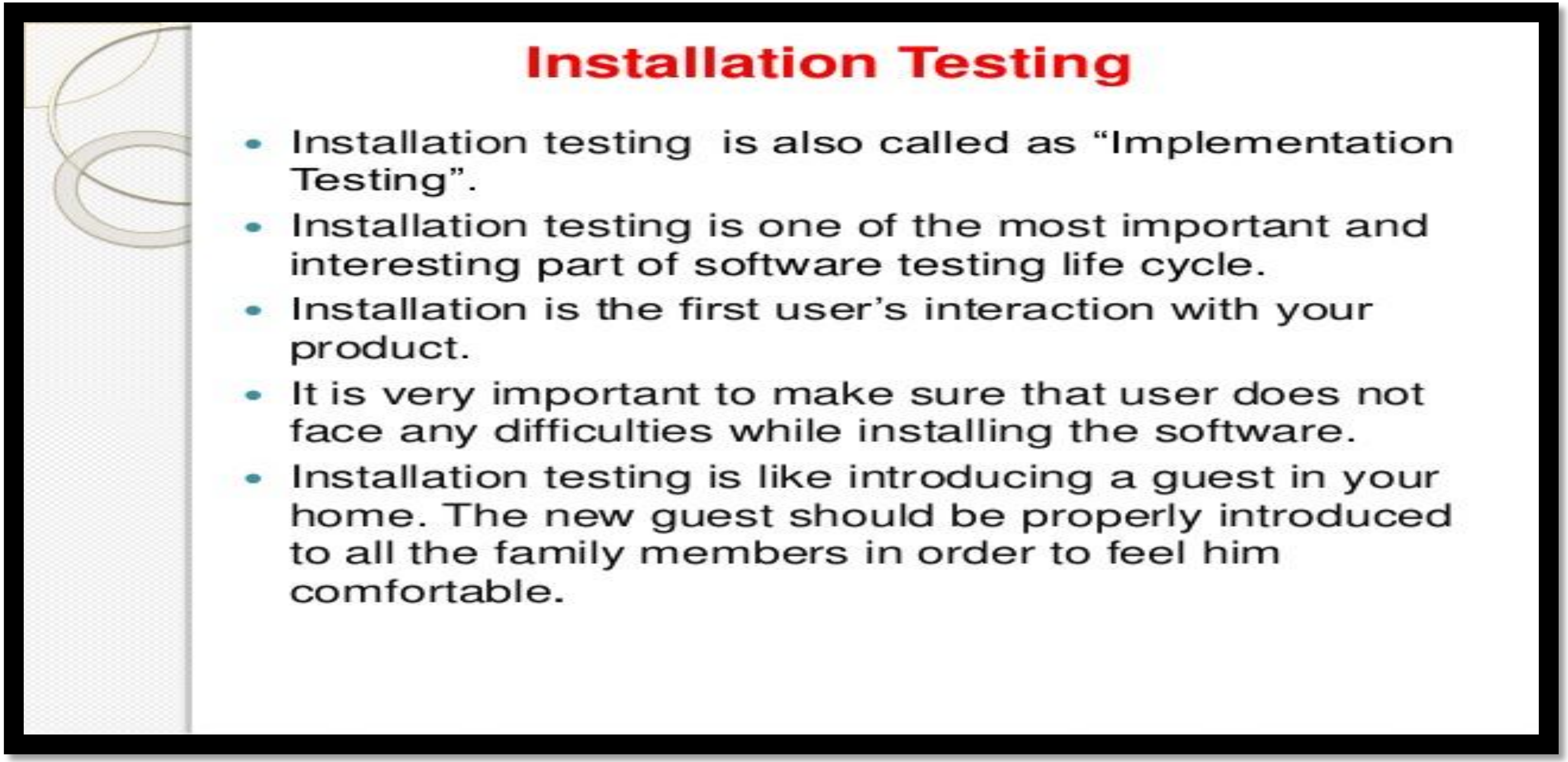
TESTING

- It is a systematic attempt to reveal the presence of errors (to "falsify" system)

- It is the process used to identify the correctness, completeness and quality of developed computer software.

- It is the process of executing a program/application under positive and negative conditions by manual or automated means. It checks for the :-

- Specification
- Functionality
- Performance

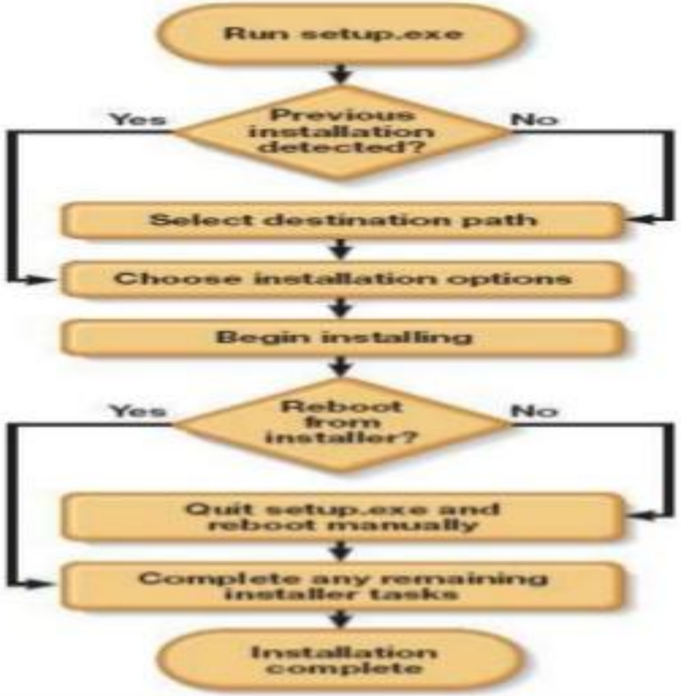


Installation Testing

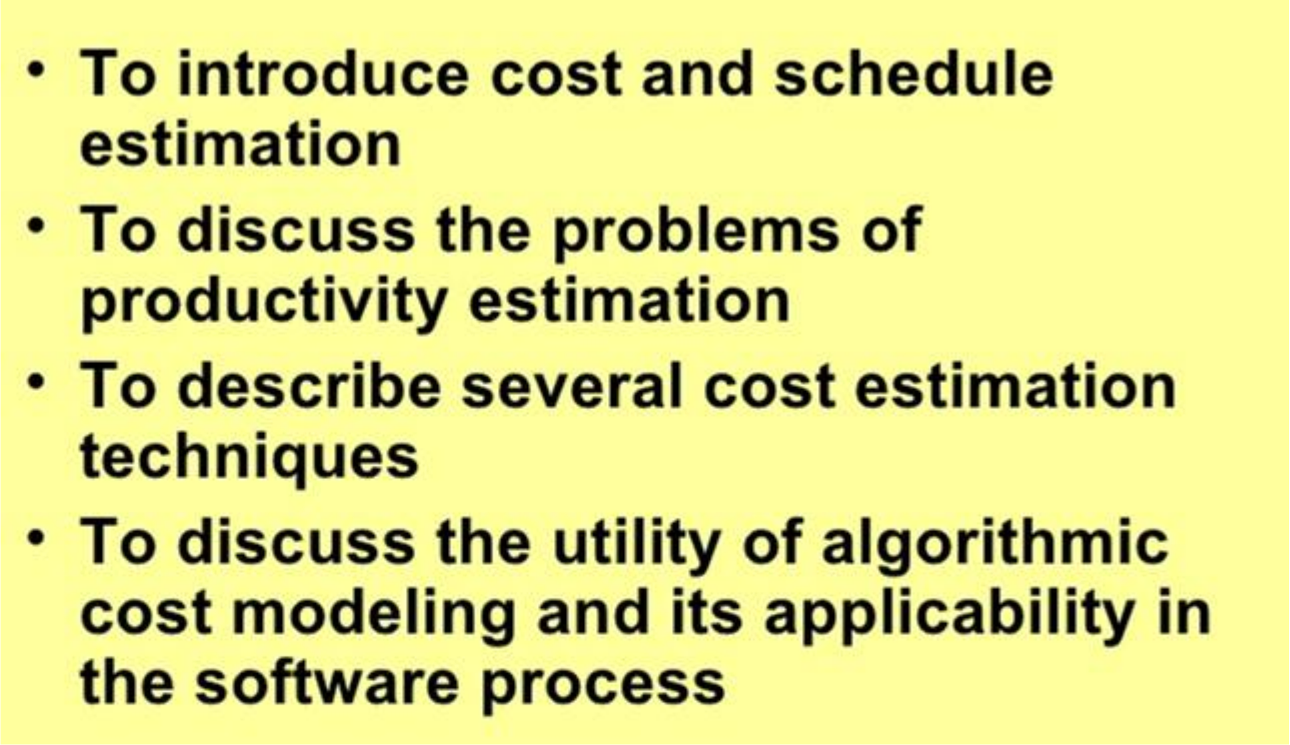
- Installation testing is also called as “Implementation Testing”.
- Installation testing is one of the most important and interesting part of software testing life cycle.
- Installation is the first user’s interaction with your product.
- It is very important to make sure that user does not face any difficulties while installing the software.
- Installation testing is like introducing a guest in your home. The new guest should be properly introduced to all the family members in order to feel him comfortable.

Installation testing tips with some broad test cases:

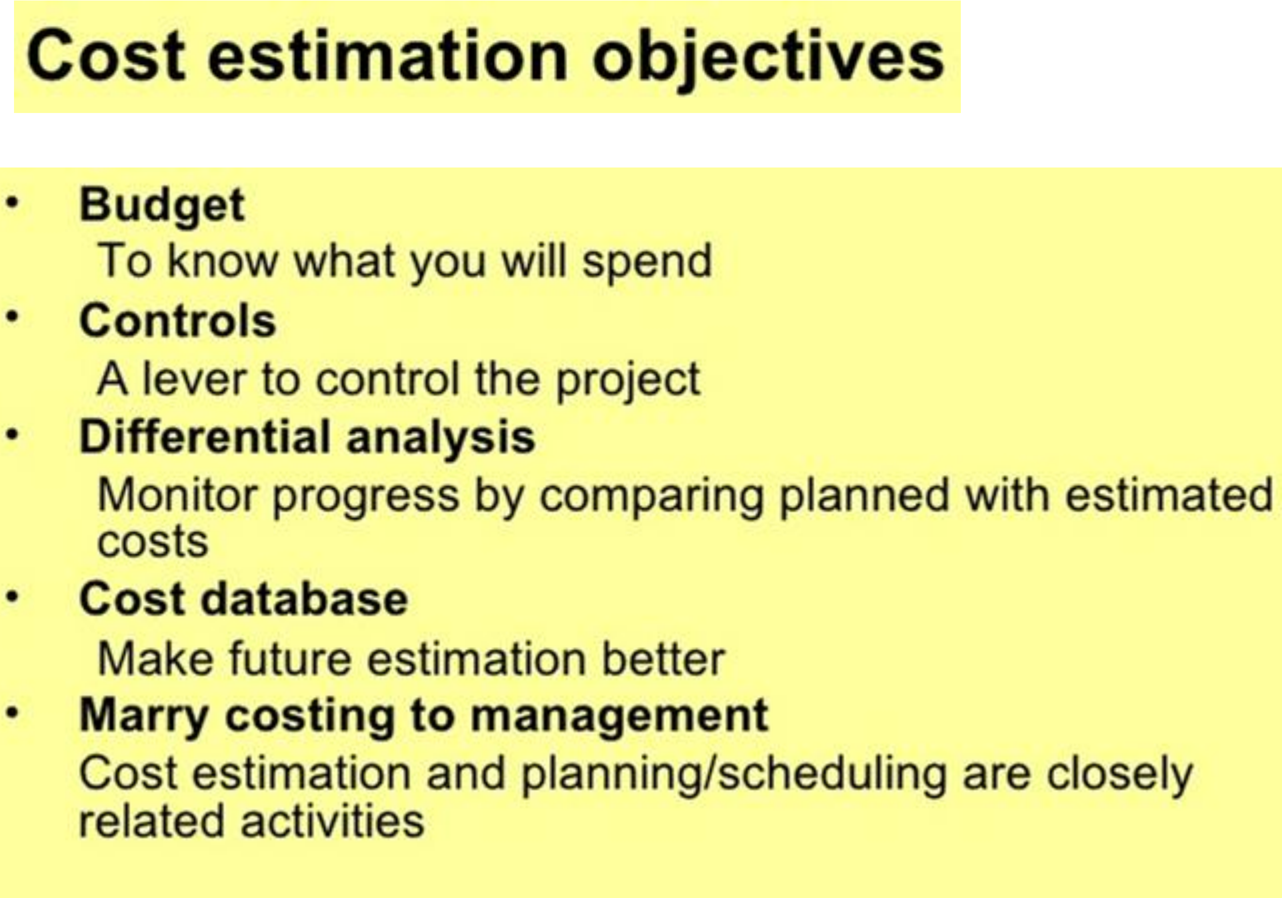
- To get **Installation testing** in step by step I have used flow chart diagram to explain in details:


Run setup.exe
Previous installation detected?
Yes
No
Select destination path
Choose installation options
Begin installing
Reboot from installer?
Yes
No
Quit setup.exe and reboot manually
Complete any remaining installer tasks
Installation complete

OBJECTIVES



- To introduce cost and schedule estimation
- To discuss the problems of productivity estimation
- To describe several cost estimation techniques
- To discuss the utility of algorithmic cost modeling and its applicability in the software process



Cost estimation objectives

- **Budget**

  To know what you will spend
- **Controls**

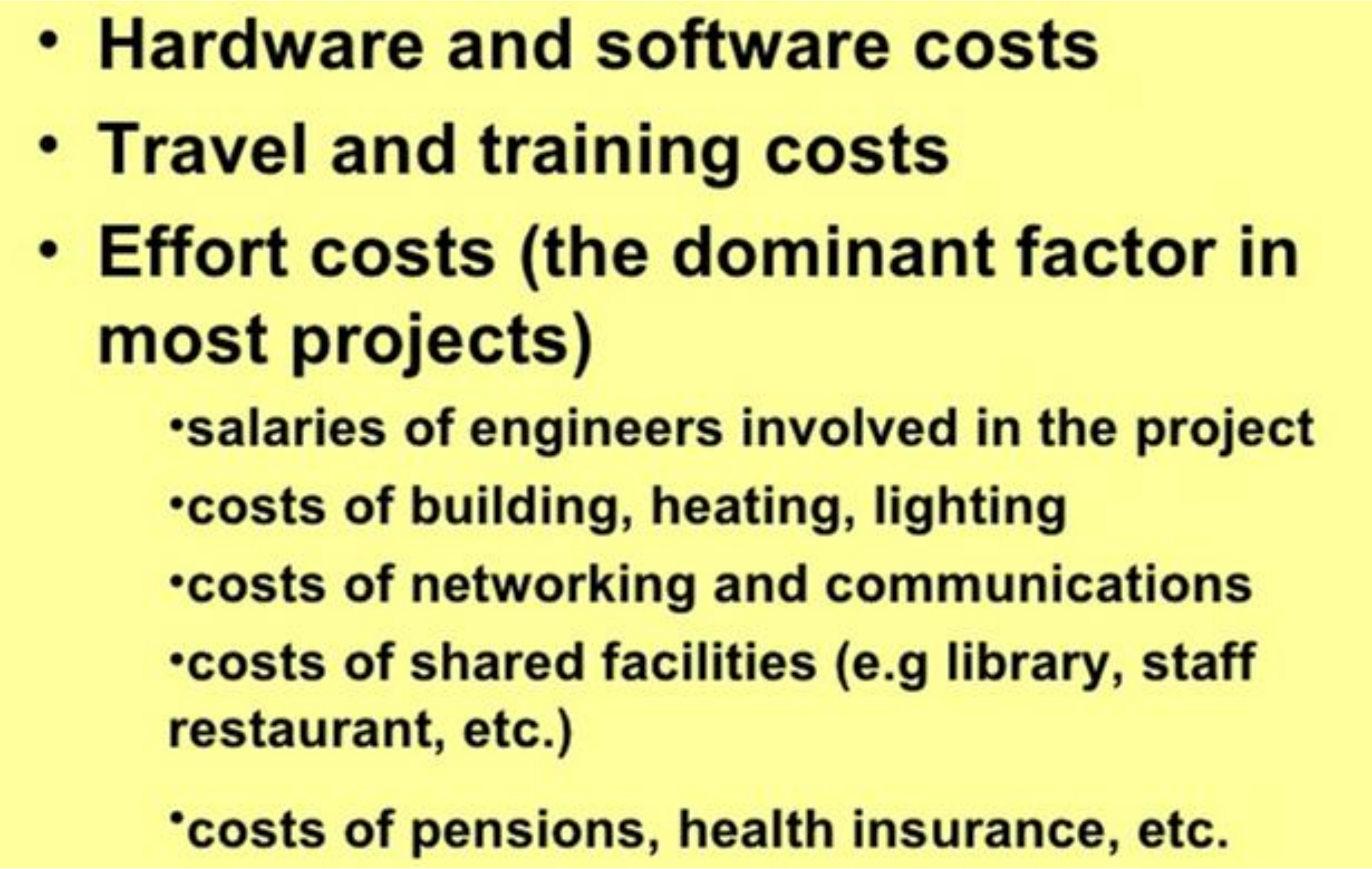
  A lever to control the project
- **Differential analysis**

  Monitor progress by comparing planned with estimated costs
- **Cost database**

  Make future estimation better
- **Marry costing to management**

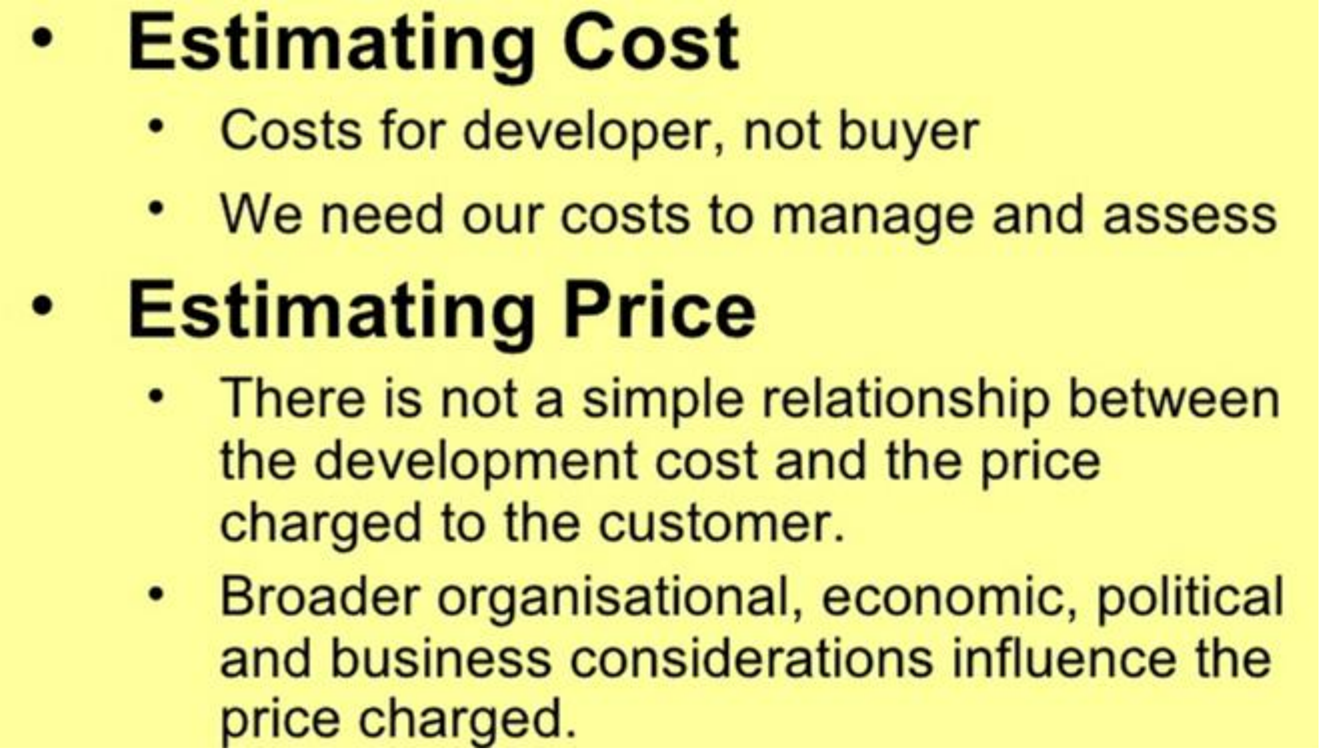
  Cost estimation and planning/scheduling are closely related activities

Software cost components



- Hardware and software costs
- Travel and training costs
- Effort costs (the dominant factor in most projects)
  - salaries of engineers involved in the project
  - costs of building, heating, lighting
  - costs of networking and communications
  - costs of shared facilities (e.g library, staff restaurant, etc.)
  - costs of pensions, health insurance, etc.

Costing and pricing



- **Estimating Cost**
  - Costs for developer, not buyer
  - We need our costs to manage and assess
- **Estimating Price**
  - There is not a simple relationship between the development cost and the price charged to the customer.
  - Broader organisational, economic, political and business considerations influence the price charged.

Productivity Measures

- **Size-related measures**
  - Must be based on some output from the software process
  - Delivered source code
  - Object code instructions
- **Function-related measures**
  - Based on an estimate of the functionality of the delivered software.
  - Function-points are the best known of this type of measure

Lines of Codes

LOC = NCLOC + CLOC

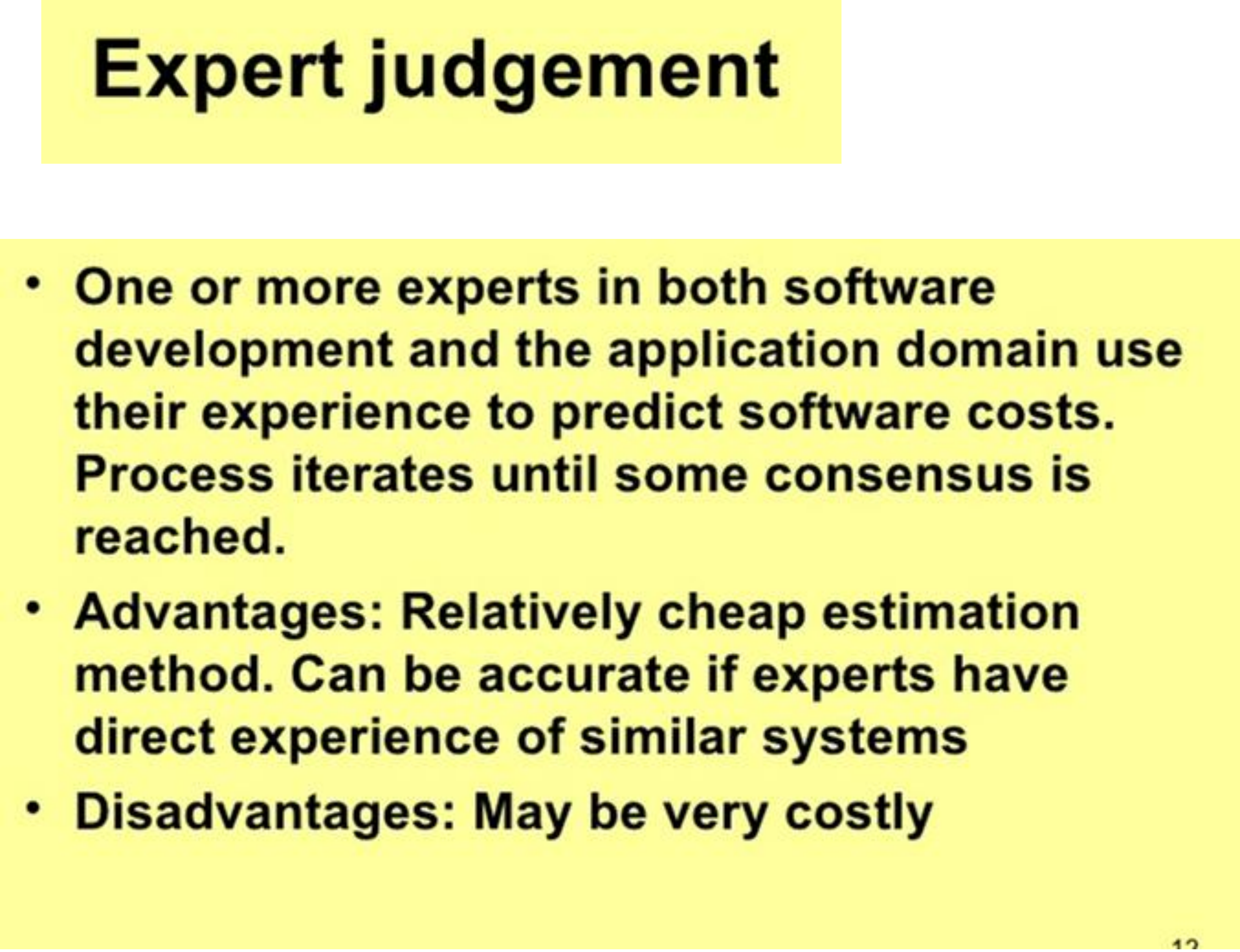
- LOC: lines of code
- NCLOC: non-commented line of code
- CLOC: commented line of code
- KLOC = one thousand of line of code

Function points

- Based on a combination of program characteristics
  - external inputs and outputs
  - user interactions
  - external interfaces
  - files used by the system
- A weight is associated with each of these
- The function point count is computed by multiplying each raw count by the weight and summing all values

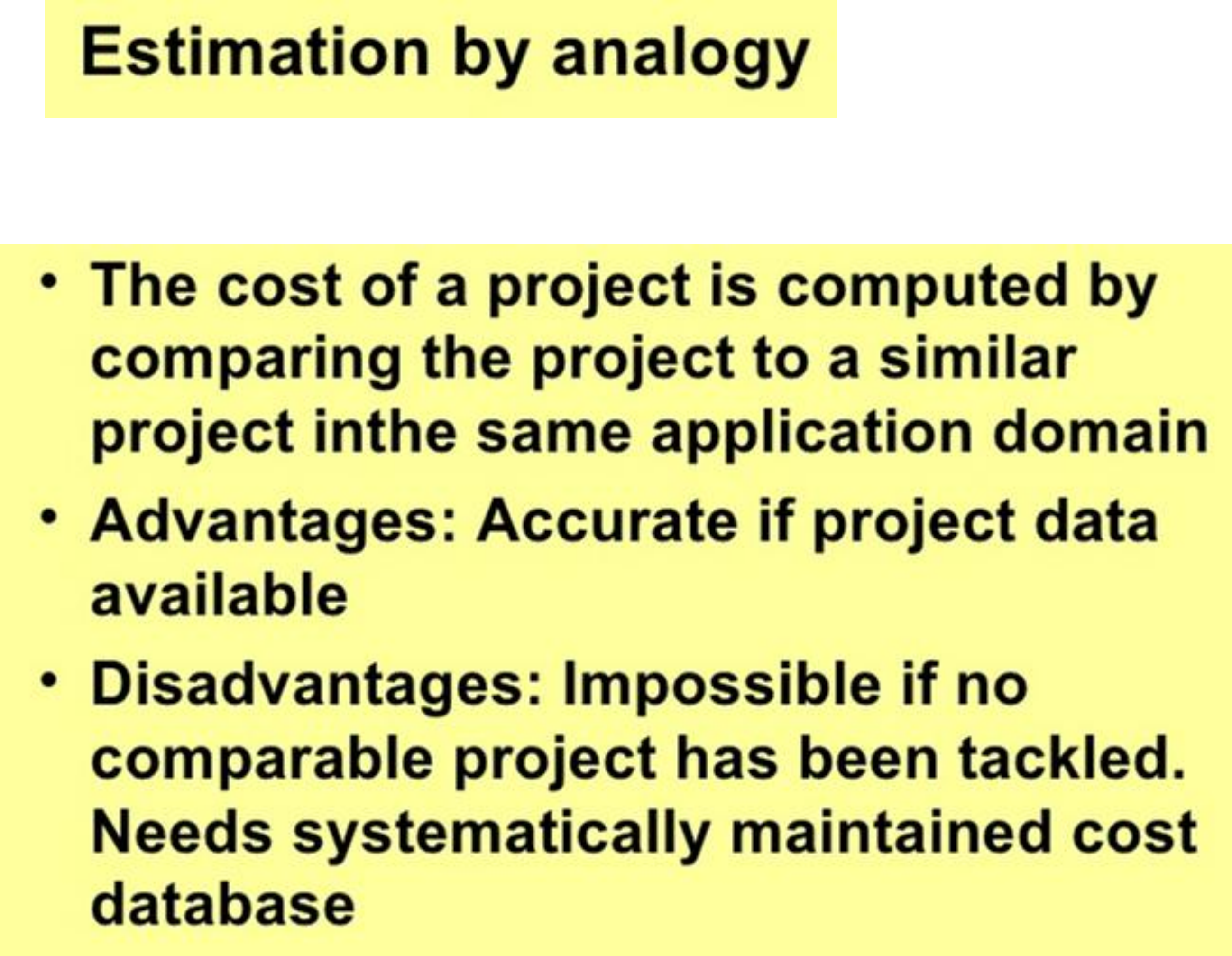
Estimation techniques

- Expert judgement
- Estimation by analogy
- Parkinson’s Law
- Pricing to win
- Top-down estimation
- Bottom-up estimation
- Algorithmic cost modelling



Expert judgement

- One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached.
- Advantages: Relatively cheap estimation method. Can be accurate if experts have direct experience of similar systems
- Disadvantages: May be very costly



Estimation by analogy

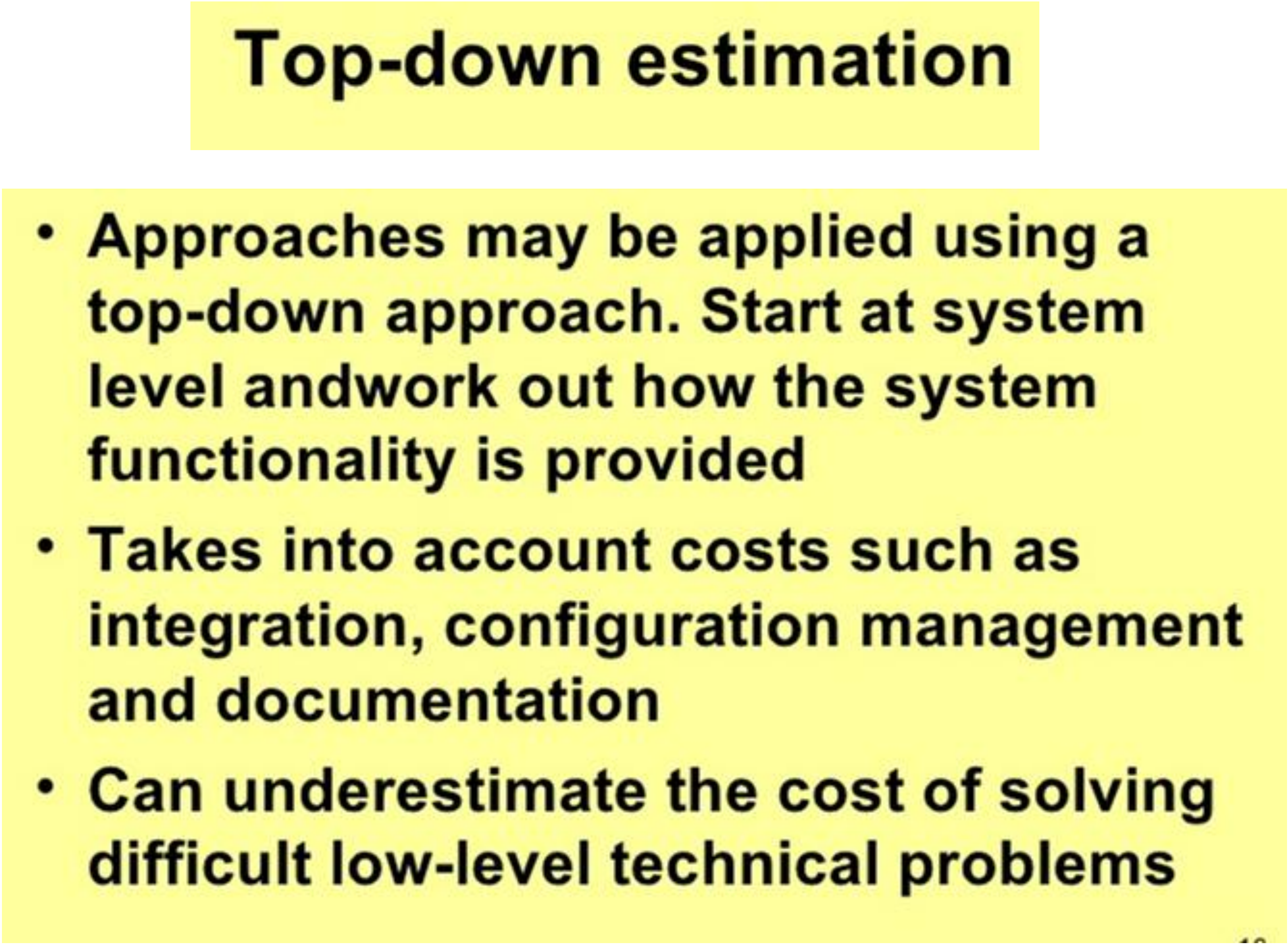
- The cost of a project is computed by comparing the project to a similar project inthe same application domain
- Advantages: Accurate if project data available
- Disadvantages: Impossible if no comparable project has been tackled. Needs systematically maintained cost database

Parkinson's Law

- The project costs whatever resources are available
- Advantages: No overspending
- Disadvantages: System is usually unfinished

Pricing to win

- The project costs whatever the customer has to spend on it
- Advantages: You get the contract
- Disadvantages: The probability that the customer gets the system he or she wants is small. Costs do not accurately reflect the work required



Top-down estimation

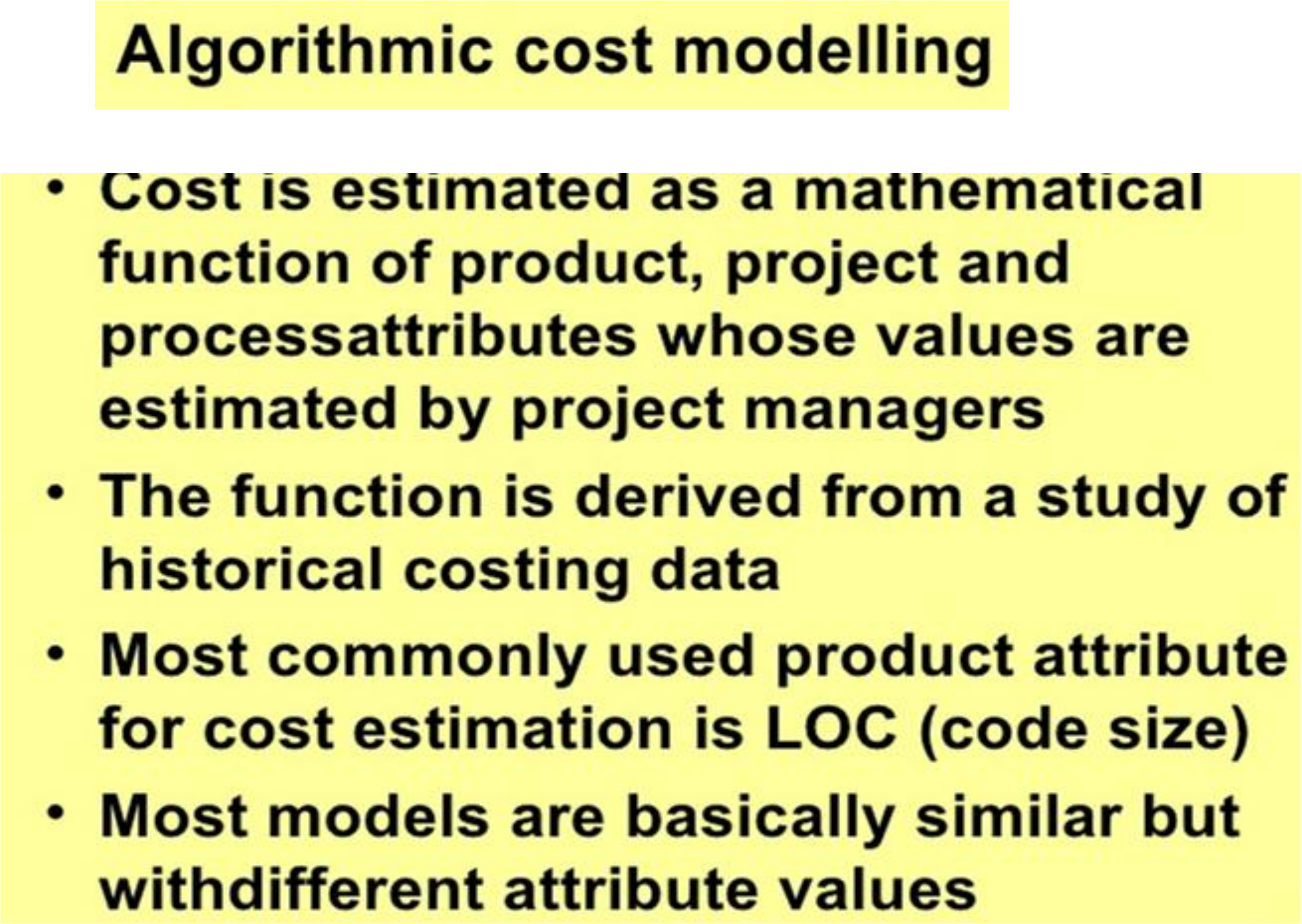
- Approaches may be applied using a top-down approach. Start at system level andwork out how the system functionality is provided
- Takes into account costs such as integration, configuration management and documentation
- Can underestimate the cost of solving difficult low-level technical problems

Bottom-up estimation

- Start at the lowest system level. The cost of each component is estimated individually.These costs are summed to give final cost estimate
- Accurate method if the system has been designed in detail
- May underestimate costs of system level activities such as integration and documentation

Estimation methods

- Each method has strengths and weaknesses
- Estimation should be based on several methods
- If these do not return approximately the same result, there is insufficient information available
- Some action should be taken to find out more in order to make more accurate estimates
- Pricing to win is sometimes the only applicable method



Algorithmic cost modelling

- Cost is estimated as a mathematical function of product, project and processattributes whose values are estimated by project managers
- The function is derived from a study of historical costing data
- Most commonly used product attribute for cost estimation is LOC (code size)
- Most models are basically similar but withdifferent attribute values

Examples of cost models

- **General form: $E = A + B \times S^{C}$**
- E: Effort cost; S: Size; A, B, C: constants

Examples:

$E = 5.2 \times (KLOC)^{0.91}$	Walston-Felix Model
$E = 5.5 + 0.73 \times (KLOC)^{1.16}$	Bailey-Basili Model
$E = 3.2 \times (KLOC)^{1.05}$	COCOMO Basic Model
$E = 5.288 \times (KLOC)^{1.047}$	Doty Model for KLOC > 9

THANK YOU
This content is taken from the text books and reference books prescribed in the syllabus.