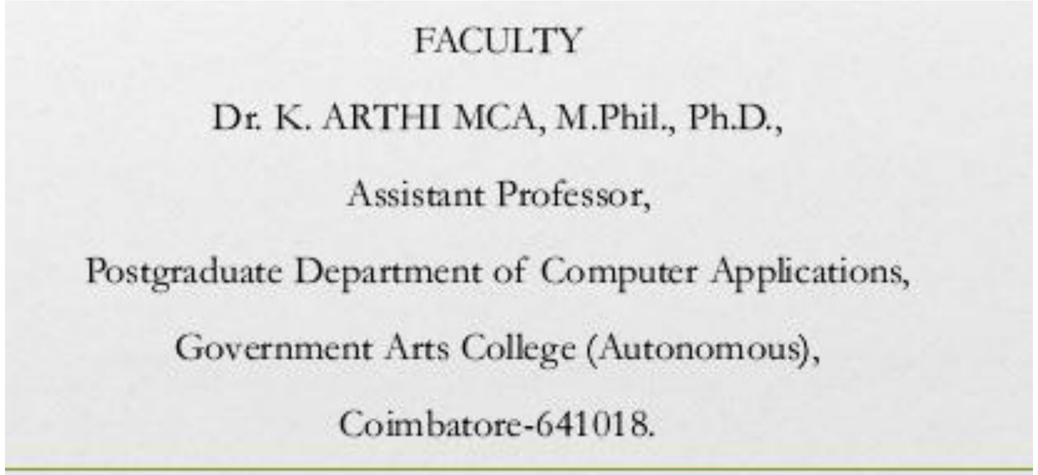
18MCA44E SOFTWARE TESTING

UNIT II – DESIGN PHASE



FACULTY

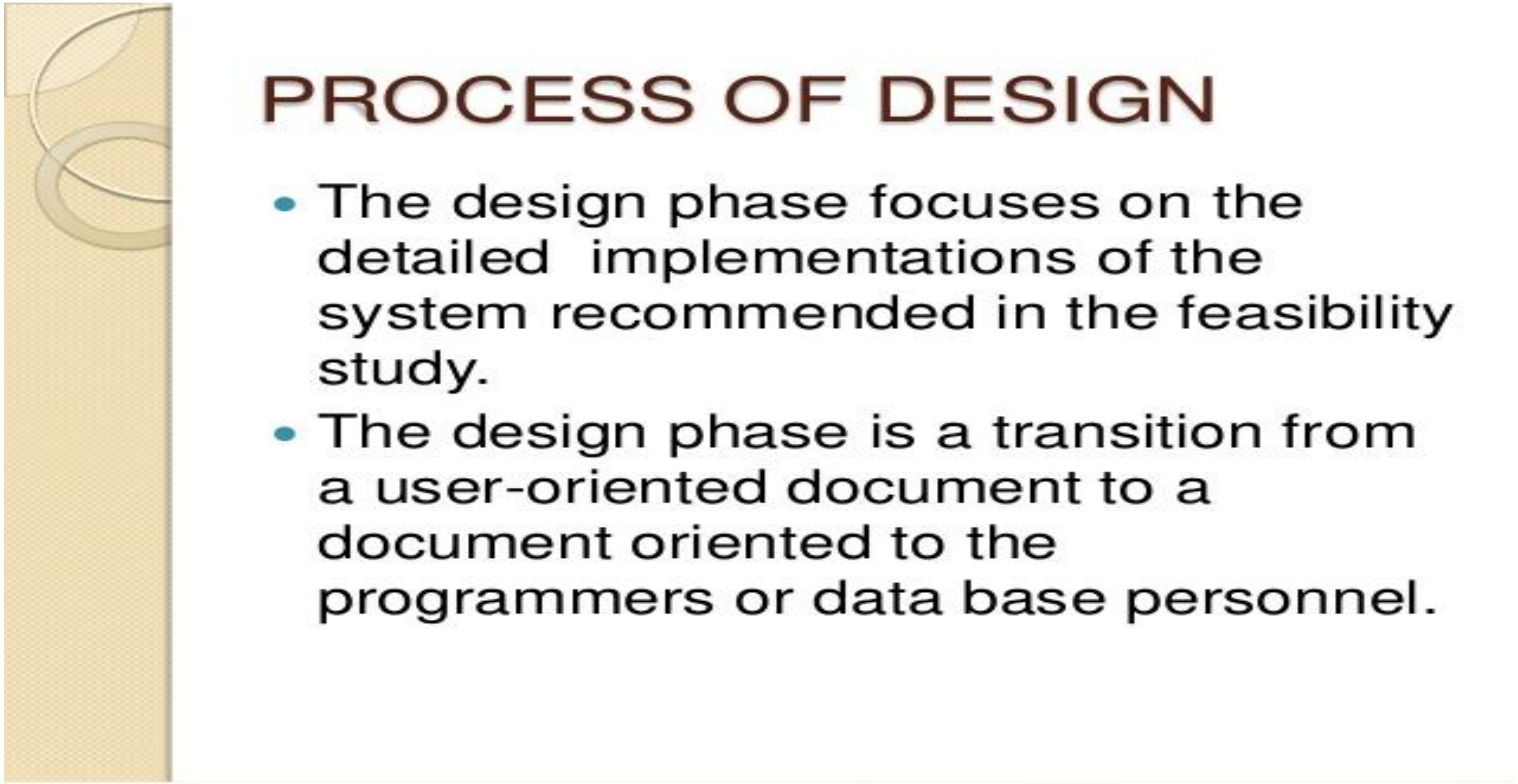
Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

Government Arts College (Autonomous),

Coimbatore-641018.

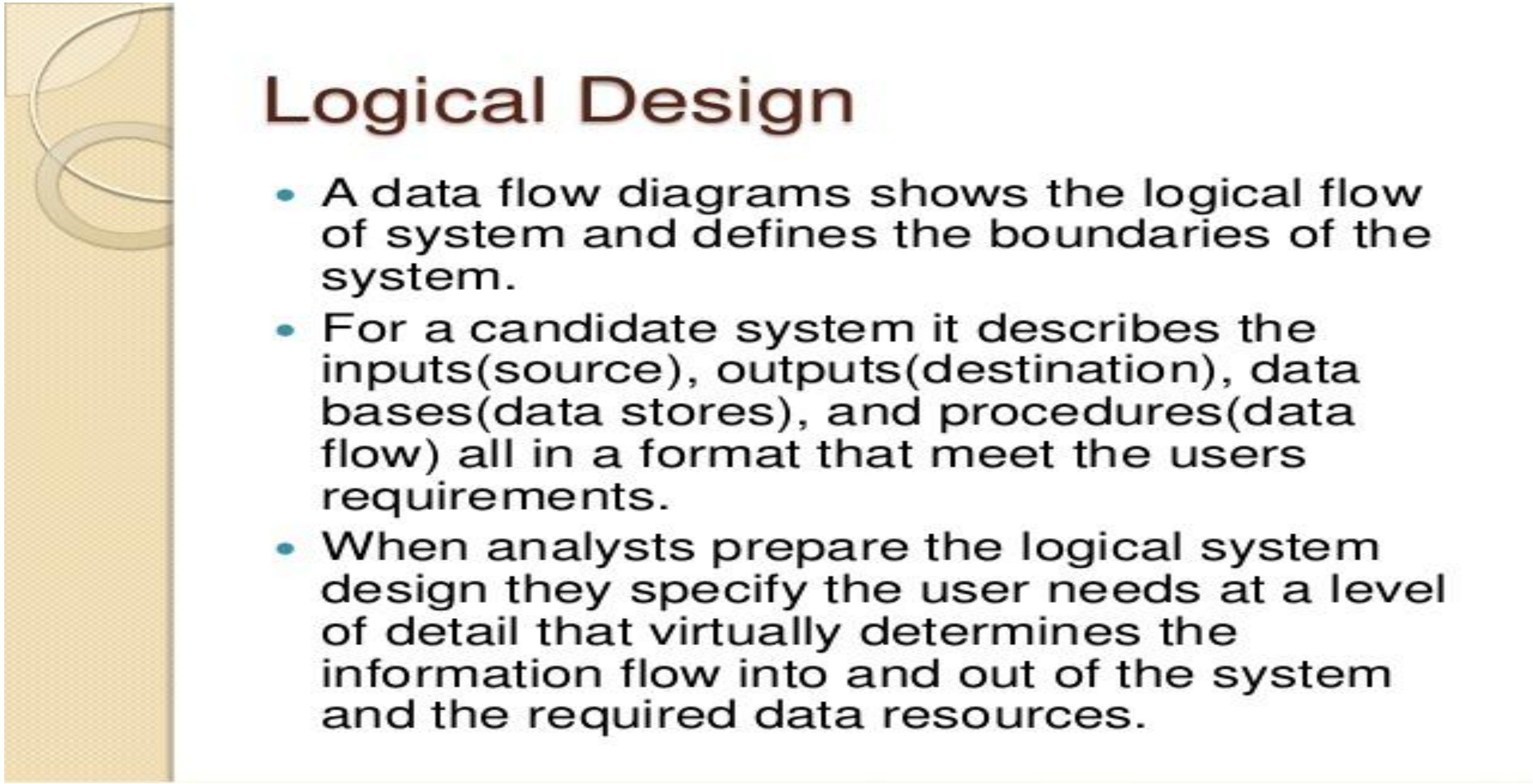


PROCESS OF DESIGN

- The design phase focuses on the detailed implementations of the system recommended in the feasibility study.
- The design phase is a transition from a user-oriented document to a document oriented to the programmers or data base personnel.

System design goes through two phases of development

- Logical Design
- Physical Design

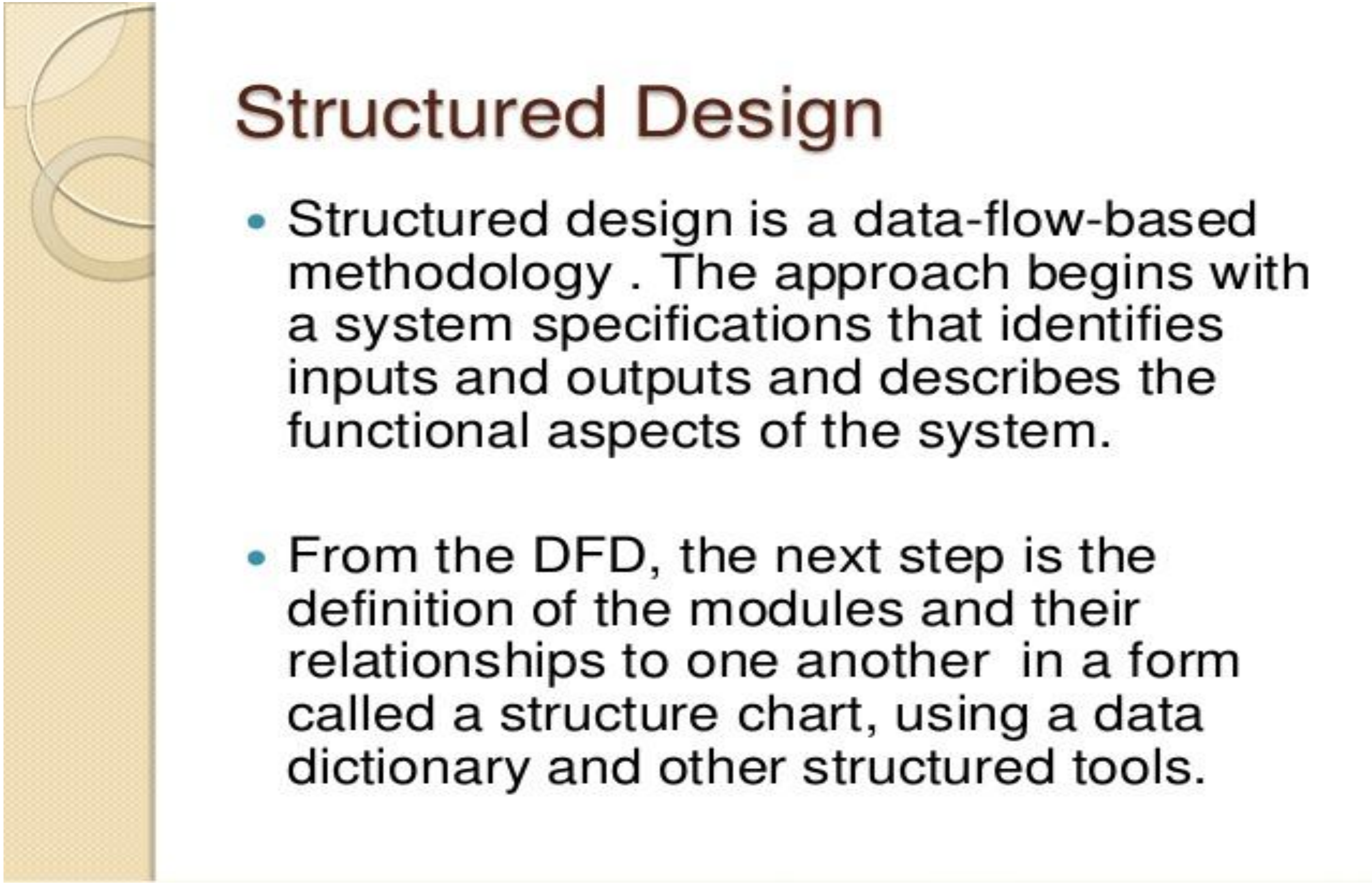


Logical Design

- A data flow diagrams shows the logical flow of system and defines the boundaries of the system.
- For a candidate system it describes the inputs(source), outputs(destination), data bases(data stores), and procedures(data flow) all in a format that meet the users requirements.
- When analysts prepare the logical system design they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources.

Physical Design

- This produces the working system by defining the design specifications that tell programmers exactly what the candidate system must do.
- In turn , the programmer writes the necessary programs or modifies the software package that accepts input from the user, performs the necessary calculations through the existing or data base , produces the report on the hard copy or display it on the screen and maintains an updated data base at all times.

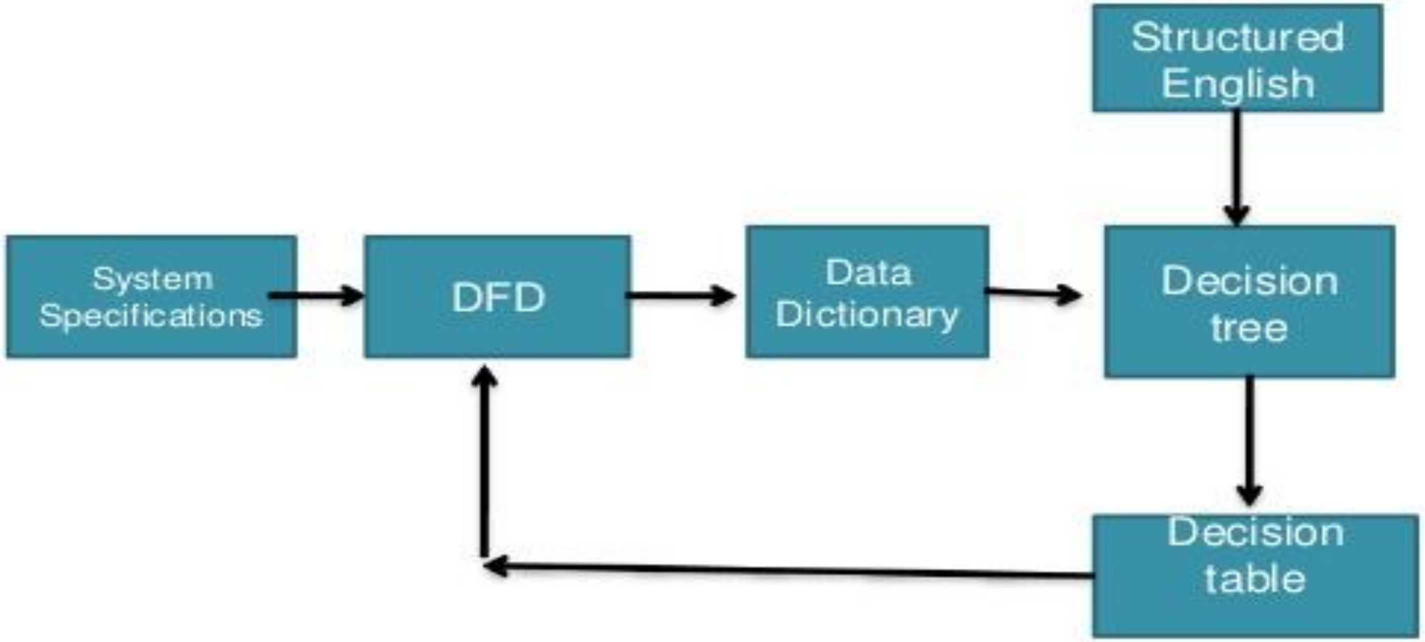


Structured Design

- Structured design is a data-flow-based methodology . The approach begins with a system specifications that identifies inputs and outputs and describes the functional aspects of the system.

- From the DFD, the next step is the definition of the modules and their relationships to one another in a form called a structure chart, using a data dictionary and other structured tools.

The Structured Design Method



Structured English

System Specifications

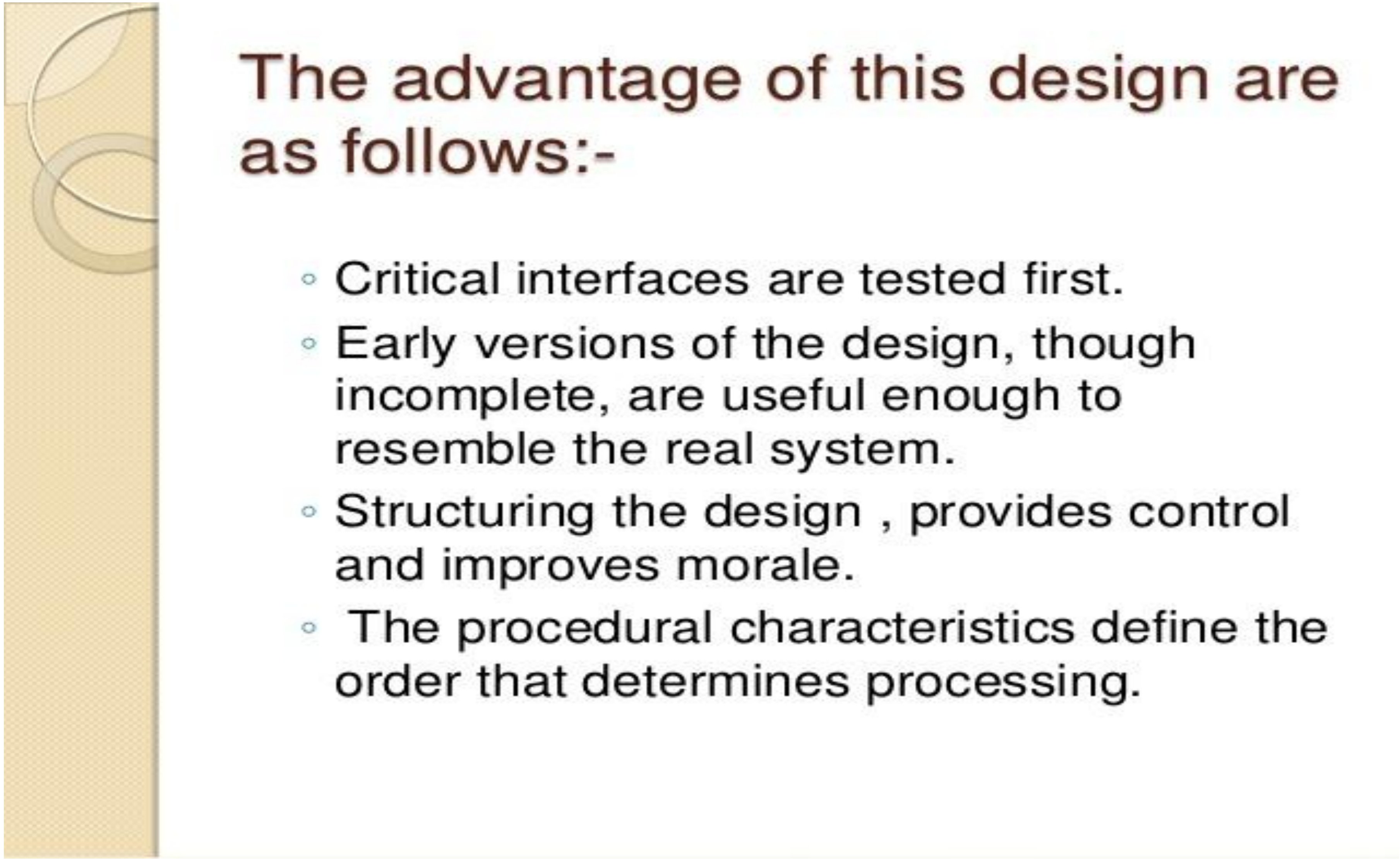
DFD

Data Dictionary

Decision tree

Decision table

- Structured design is an attempt to minimize complexity and make a problem manageable by subdividing it into smaller segments, which is called modularization or decomposition.
- A design is said to be top-down if it consists of a hierarchy of modules, with each module having a single entry and a single exit subroutine.



The advantage of this design are as follows:-

- Critical interfaces are tested first.
- Early versions of the design, though incomplete, are useful enough to resemble the real system.
- Structuring the design , provides control and improves morale.
- The procedural characteristics define the order that determines processing.

PROGRAMME:

It is a procedure of working with the people in an effort to recognize the problems and to find out the possible solutions, objectives and goals.

PLAN OF WORK:

It is an outline of activities so arranged as to enable efficient execution of entire programme.

OBJECTIVES OF PROGRAMME PLANNING

- To ensure careful consideration of what is to be done and why.
- To furnish a guide against which to judge all new proposals
- To establish objectives towards which progress can be measured and evaluated
- To have a means of choosing the important / deep rooted from incidental / minor / less important problems and the permanent from the temporary changes

- To develop a common understanding about the means and ends between various functionaries and organizations
- To ensure continuity during changes in personnel
- To help develop leadership
- Avoid wastage of resources, money and promote efficiency
- To justify expenditure and ensure flow of funds
- To have available a written statement for public use.

PRINCIPLES OF PROGRAMME PLANNING

1. It should be based on an analysis of the past experience, present situation and future needs.
2. It should have clear and significant objectives which could justify important needs of the people
3. It should fix up priority on the basis of available resources and time

4)It should clearly indicate the availability and utilization of resources.

5) It should have general agreement at various levels

6) It should involve people at local level.

7) It should involve relevant institutions and organization.

8) It should have definite plan of work

9) It should provide for evaluation of results and reconsideration of the program.

10) It should provide for equitable distribution of benefits amongst the members of community.

STEPS IN PROGRAMME PLANNING PROCESS

Three phases involved in programme planning process are:

1. Programme formulation
2. Programme execution
3. Programme evaluation

These three phases have the following 8 sub-Phases

i. Collection of facts and analysis

ii. Identification of problems

iii. Determination of objectives

iv. Developing the plan of activities

v. Carrying out the activities

vi. Continuous checking

vii. Evaluation of results

viii. Reconsideration

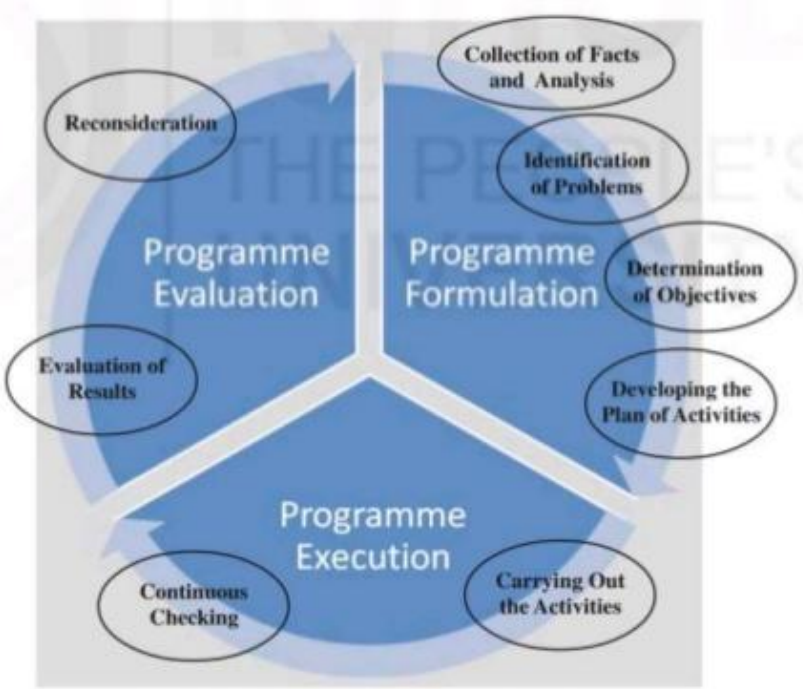

Collection of Facts and Analysis
Reconsideration
Identification of Problems
Programme Evaluation
Programme Formulation
Determination of Objectives
Evaluation of Results
Developing the Plan of Activities
Programme Execution
Continuous Checking
Carrying Out the Activities

Fig. 1.1: Development Programme Planning Process

Source: ignou peoples university

1)Collection of facts :

- It is the starting point of program planning process.
- The information that are required to develop and execute program are to be collected.
- Collection of information done through PRA, survey, available records or use of key communicators.

2) Analysis of situation

- Collected information need to be analyzed
- While analyzing, grouping of data, classification of data, use of statistical methods and interpretation are necessary for developing a program.

3)Identification of problem

- A proper analysis and interpretation of the collected data shall helps in correctly identifying the problems.
- There may be many problems, but only the urgent and significant ones (priority based) which may be solved with the available resources and with in the limit of time, should be selected.

4)Determination of objectives and goals

- The objectives are then set forth on the basis of the significant needs/problems identified.
- The objectives are to be decided because ***plan of action*** has to be developed on the basis of objectives only.
- To make objectives realistic and actionable, there is need to state them in terms of ***goals***.

5)Developing plan of work and calendar of operation

- The plan of work should be in written form and shall indicate ***who shall do what job i.e.***
- Which institutions, organizations, service departments shall be involved
- What will be ***financial requirement*** and how it shall be met
- What arrangements shall be made for marketing of the produce, training of the farmers and so on.
- ***Calendar of operations*** – it is the time schedule of program, it shall specify the how and when all significant activities shall be performed.

7) Evaluation of progress

- Process to determine the ***extent to which the objectives has been*** attained is evaluation
- It should be continuous process not only to measure the end results but also to ***ensure that all the steps are correctly followed.***

- Program evaluation involves the following essential steps

- setting up of some standards or criteria in relation to the objectives.

- collection of information.

- making judgment and drawing some unbiased and valid conclusion.

8)Reconsideration and revision of program

- On the basis of results of evaluation, the program should be considered and revised if needed.
- Reconsideration should emphasize on ***technical defects*** if noticed during execution of program.

What is debugging?

"Bug" and "debugging" are attributed to the discovery of a moth found in a Mark II computer at Harvard University.

Debugging is not troubleshooting

Debugging Rules...

1. **Understand the system**

2. Make it fail

3. **Quit thinking and look**

4. Divide and Conquer

5. **Change one thing at a time**

6. Keep an audit trail

7. **Check the plug**

8. Get a fresh view

9. **If you didn't fix it, it aint fixed**

Debugging - A Simple Method

a) Start

|

b) Construct a Hypothesis

|

c) Construct an Experiment

|

d) Hypothesis Proved? - goto b)

|

Need more evidence? - goto c)

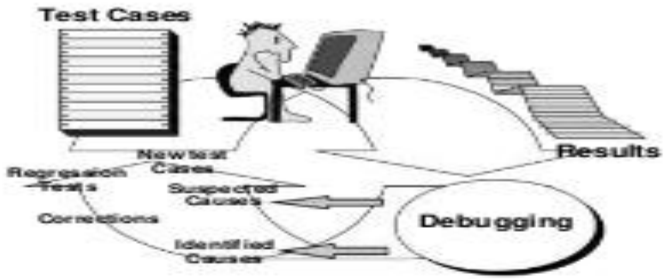
|

Stop

Debugging

- Debugging will be the process that results in the removal of the error after the execution of a test case.
- Its objective is to remove the defects uncovered by tests.
- Because the cause may not be directly linked to the symptom, it may be necessary to enumerate hypotheses explicitly and then design new test cases to allow confirmation or rejection of the hypothesized cause.

The Debugging Process



Thanks to M8034 @ Peter Lo 2006

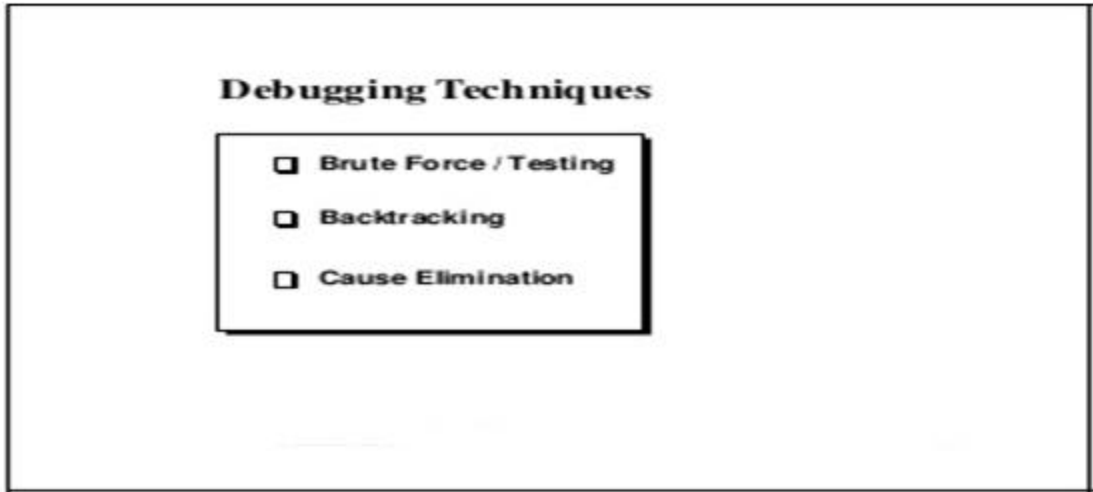
What is Bug?

- A bug is a part of a program that, if executed in the right state, will cause the system to deviate from its specification (or cause the system to deviate from the behavior desired by the user).

Characteristics of Bugs

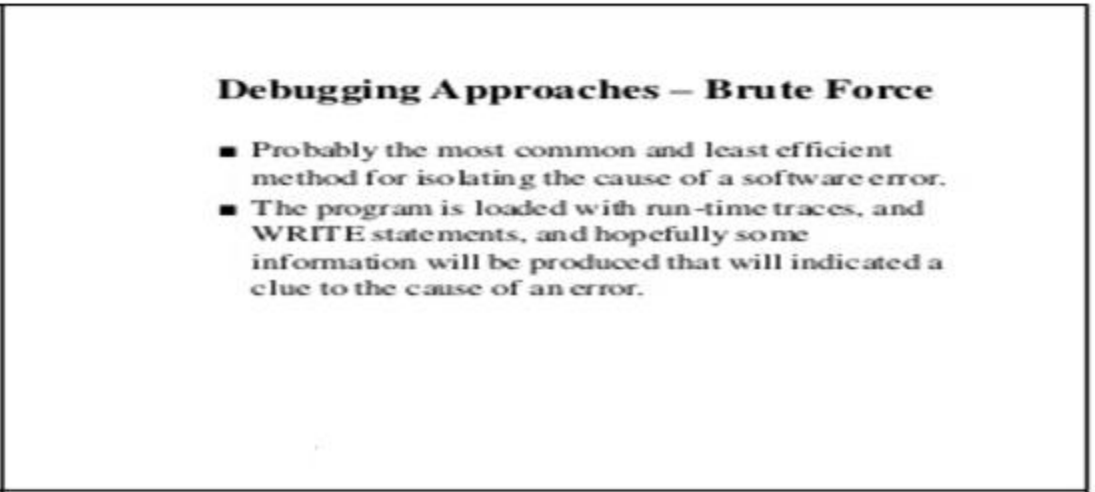
- The symptom and the cause may be geographically remote
- The symptom may disappear when another error is corrected
- The symptom may actually be caused by non-errors
- The symptom may be caused by a human error that is not easily traced
- It may be difficult to accurately reproduce input conditions
- The symptom may be intermittent. This is particularly common in embedded systems that couple hardware and software inextricably
- The symptom may be due to causes that are distributed across a number of tasks running on different processors

Thanks to M8034 @ Peter Lo 2006



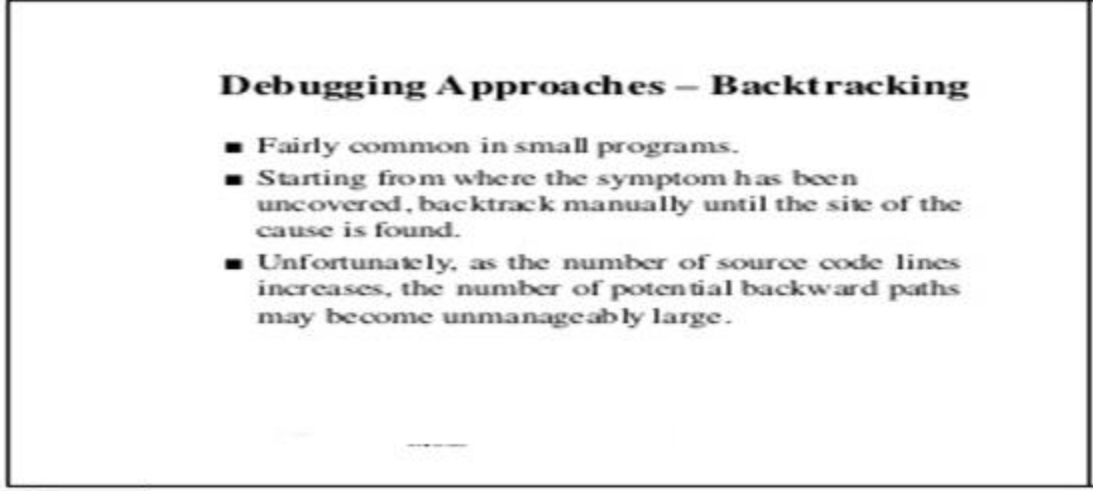
Debugging Techniques

- ❑ Brute Force / Testing
- ❑ Backtracking
- ❑ Cause Elimination



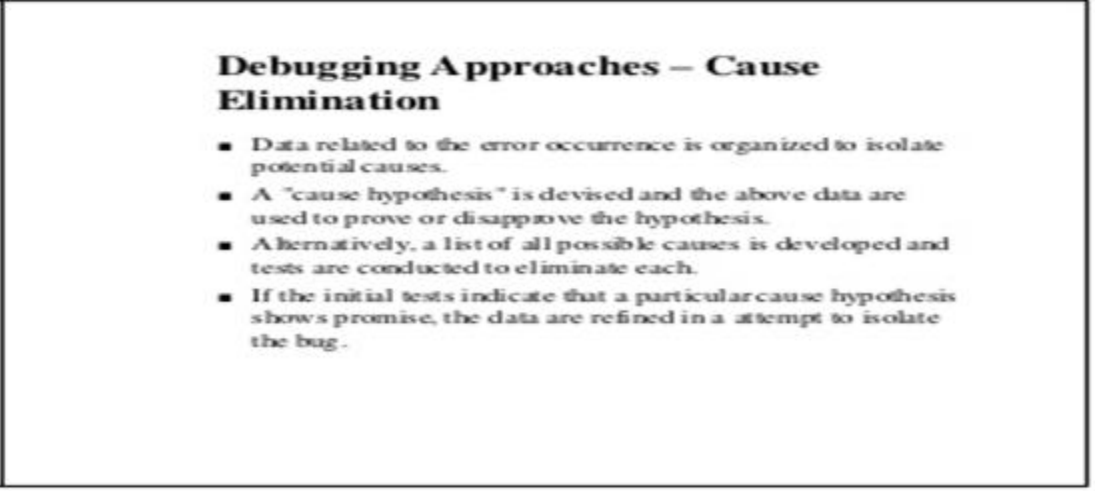
Debugging Approaches – Brute Force

- Probably the most common and least efficient method for isolating the cause of a software error.
- The program is loaded with run-time traces, and WRITE statements, and hopefully some information will be produced that will indicated a clue to the cause of an error.



Debugging Approaches – Backtracking

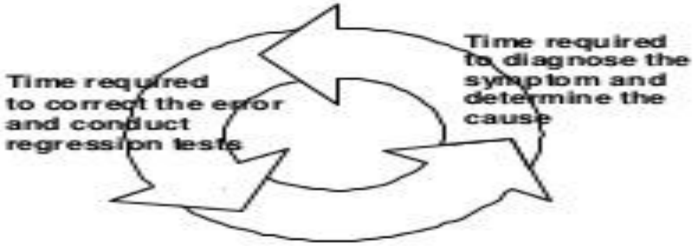
- Fairly common in small programs.
- Starting from where the symptom has been uncovered, backtrack manually until the site of the cause is found.
- Unfortunately, as the number of source code lines increases, the number of potential backward paths may become unmanageably large.



Debugging Approaches – Cause Elimination

- Data related to the error occurrence is organized to isolate potential causes.
- A "cause hypothesis" is devised and the above data are used to prove or disapprove the hypothesis.
- Alternatively, a list of all possible causes is developed and tests are conducted to eliminate each.
- If the initial tests indicate that a particular cause hypothesis shows promise, the data are refined in a attempt to isolate the bug.

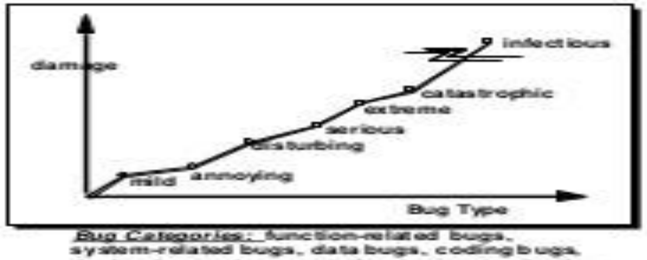
Debugging Effort



Time required to correct the error and conduct regression tests

Time required to diagnose the symptom and determine the cause

Consequences of Bugs



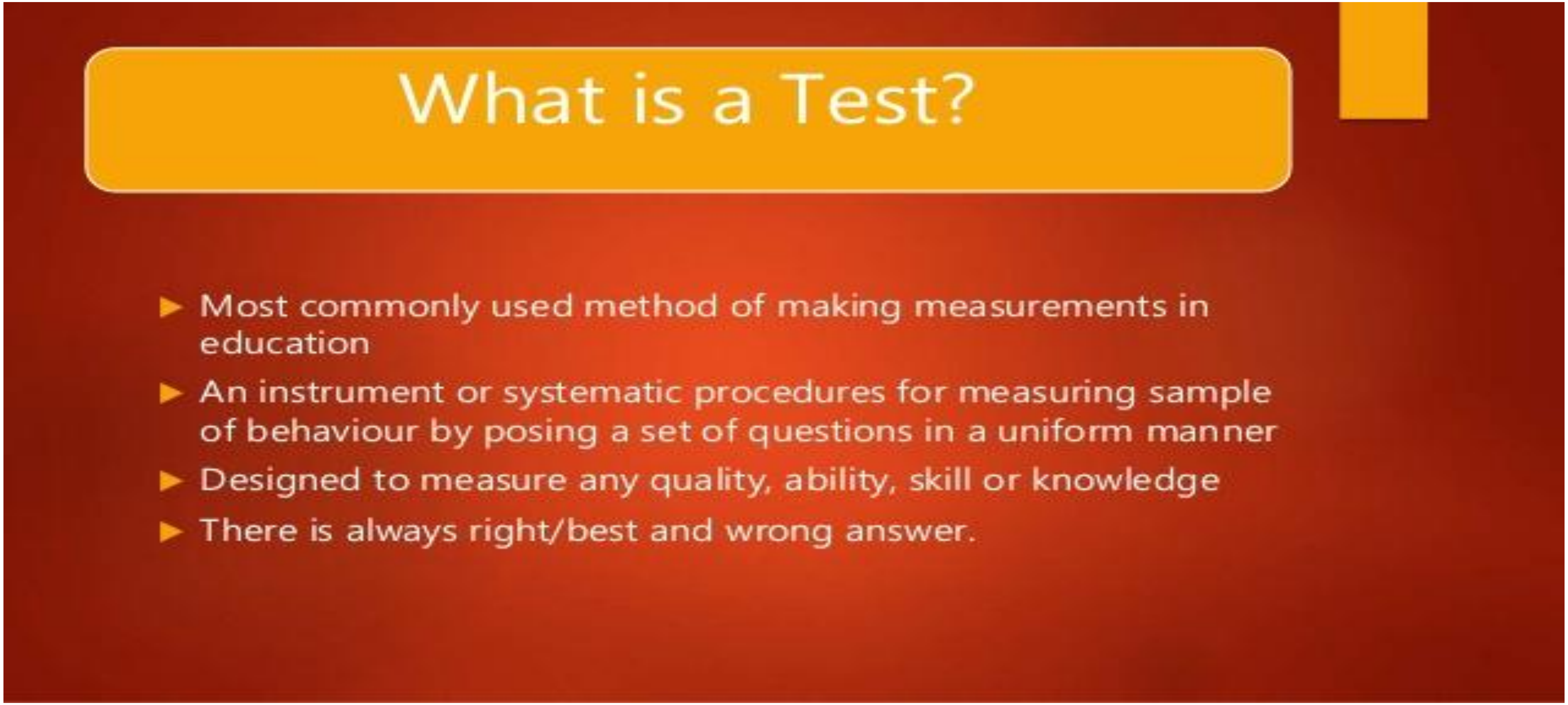
Bug Categories: function-related bugs, system-related bugs, data bugs, coding bugs, design bugs, documentation bugs, standards violations, etc.

Debugging Tools

- Debugging compilers
- Dynamic debugging aides ("tracers")
- Automatic test case generators
- Memory dumps

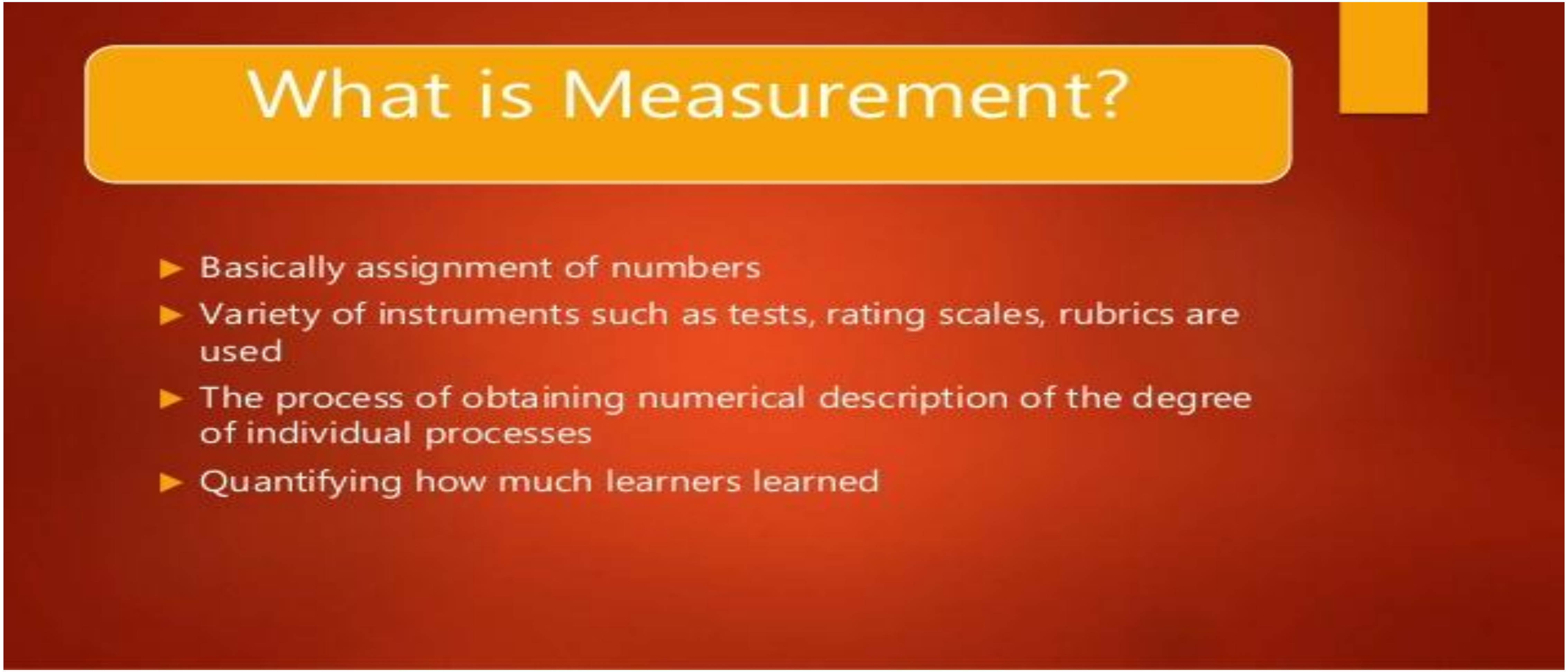
Debugging: Final Thoughts

- Don't run off half-cocked, think about the symptom you're seeing.
- Use tools (e.g., dynamic debugger) to gain more insight.
- If at an impasse, get help from someone else.
- Be absolutely sure to conduct regression tests when you do "fix" the bug.



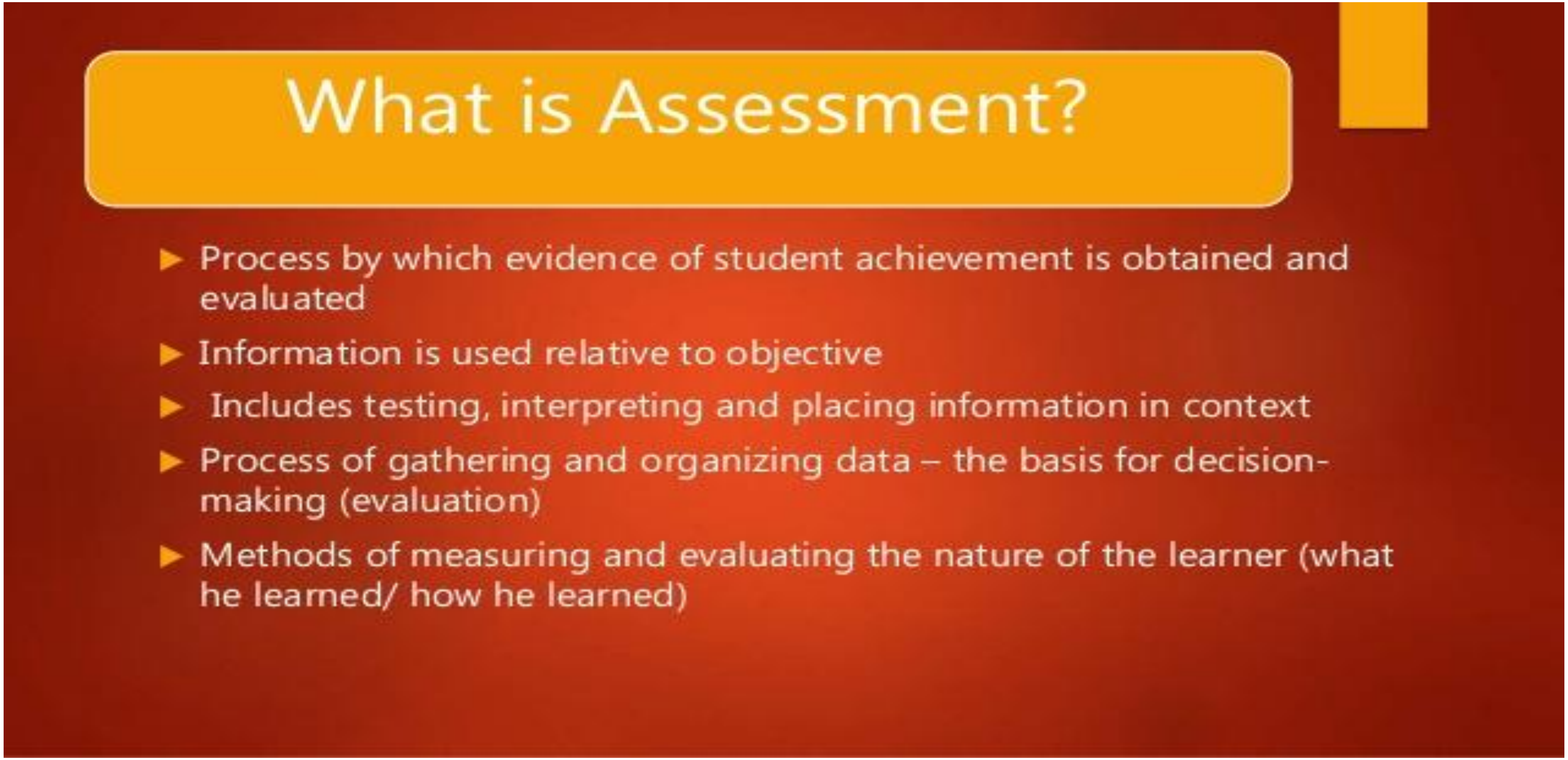
What is a Test?

- Most commonly used method of making measurements in education
- An instrument or systematic procedures for measuring sample of behaviour by posing a set of questions in a uniform manner
- Designed to measure any quality, ability, skill or knowledge
- There is always right/best and wrong answer.



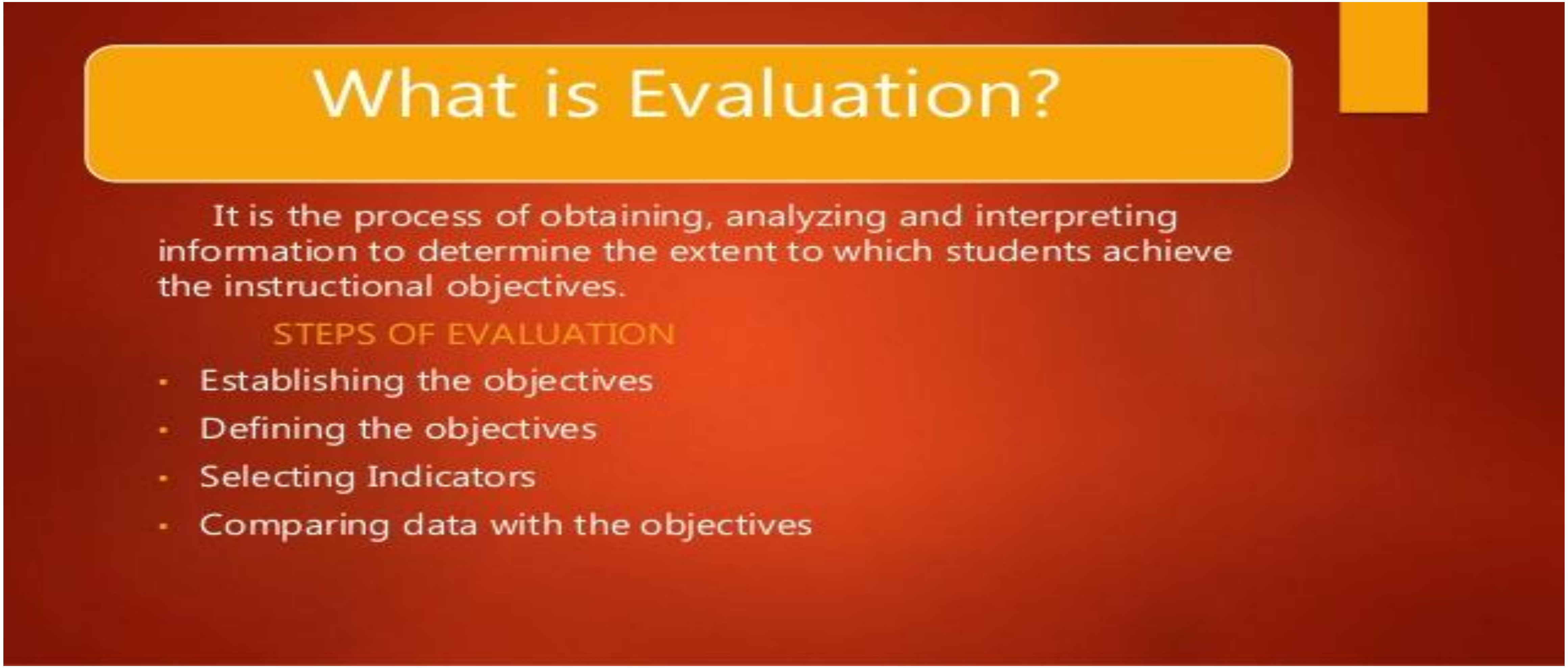
What is Measurement?

- Basically assignment of numbers
- Variety of instruments such as tests, rating scales, rubrics are used
- The process of obtaining numerical description of the degree of individual processes
- Quantifying how much learners learned



What is Assessment?

- Process by which evidence of student achievement is obtained and evaluated
- Information is used relative to objective
- Includes testing, interpreting and placing information in context
- Process of gathering and organizing data – the basis for decision-making (evaluation)
- Methods of measuring and evaluating the nature of the learner (what he learned/ how he learned)

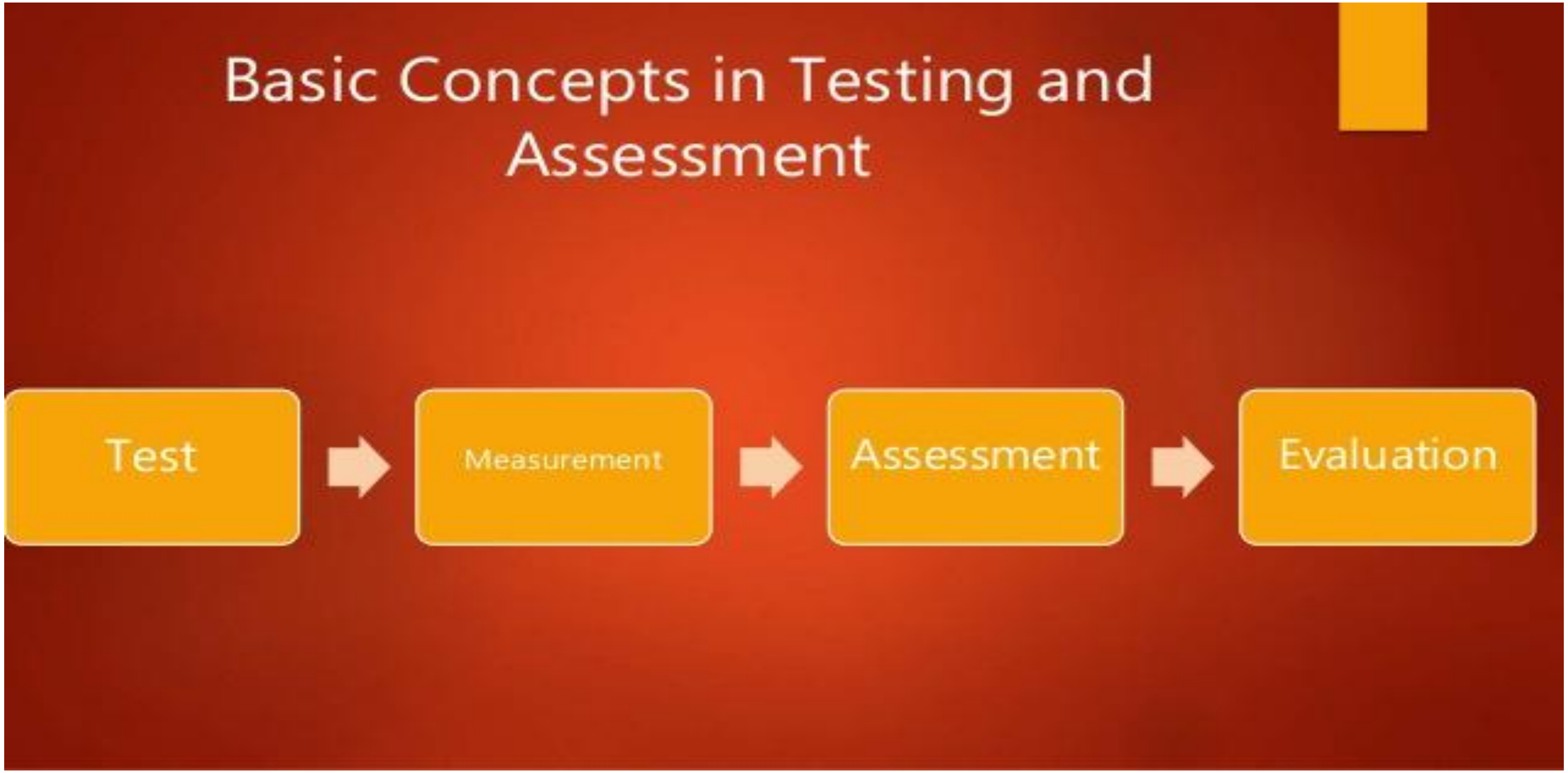


What is Evaluation?

It is the process of obtaining, analyzing and interpreting information to determine the extent to which students achieve the instructional objectives.

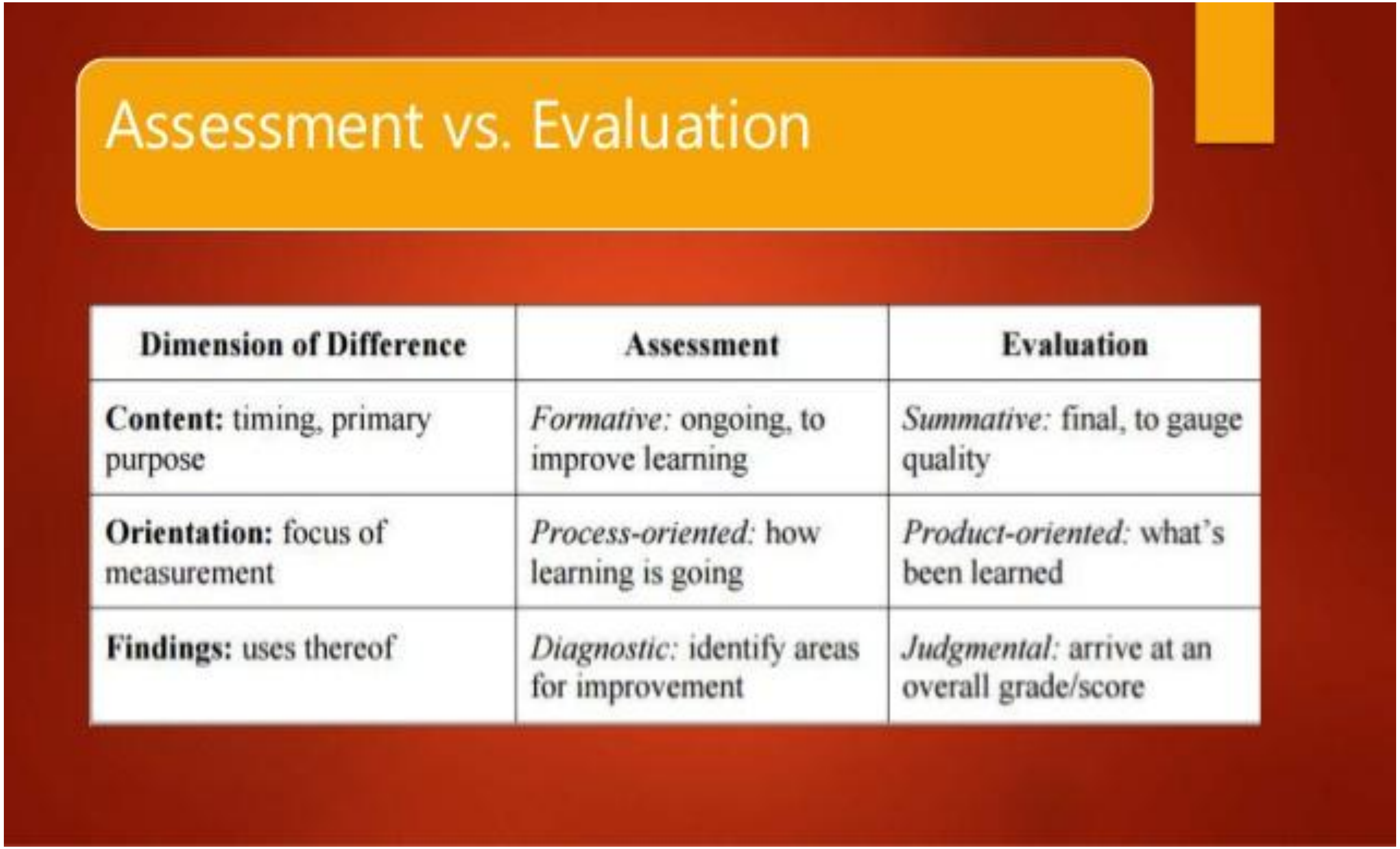
STEPS OF EVALUATION

- Establishing the objectives
- Defining the objectives
- Selecting Indicators
- Comparing data with the objectives



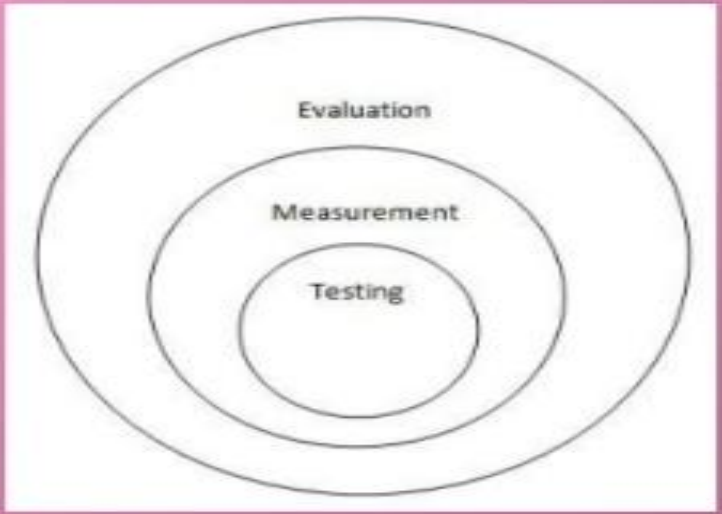
Basic Concepts in Testing and Assessment

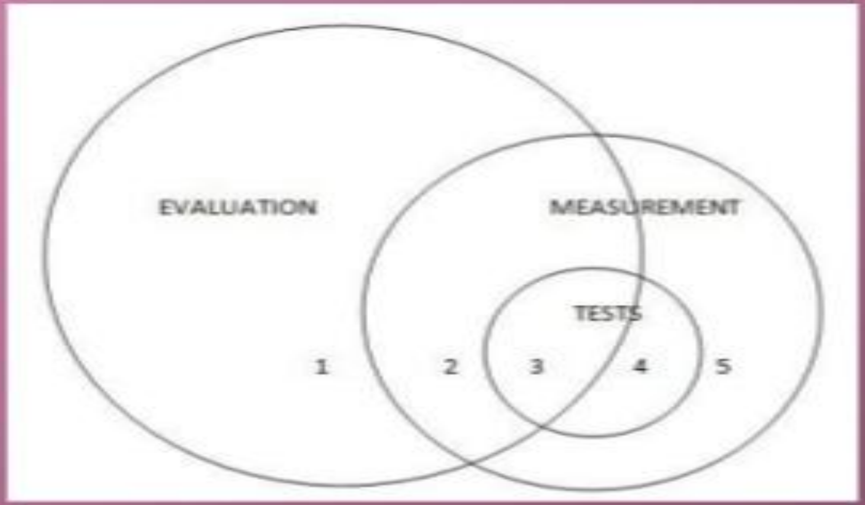
Test → Measurement → Assessment → Evaluation



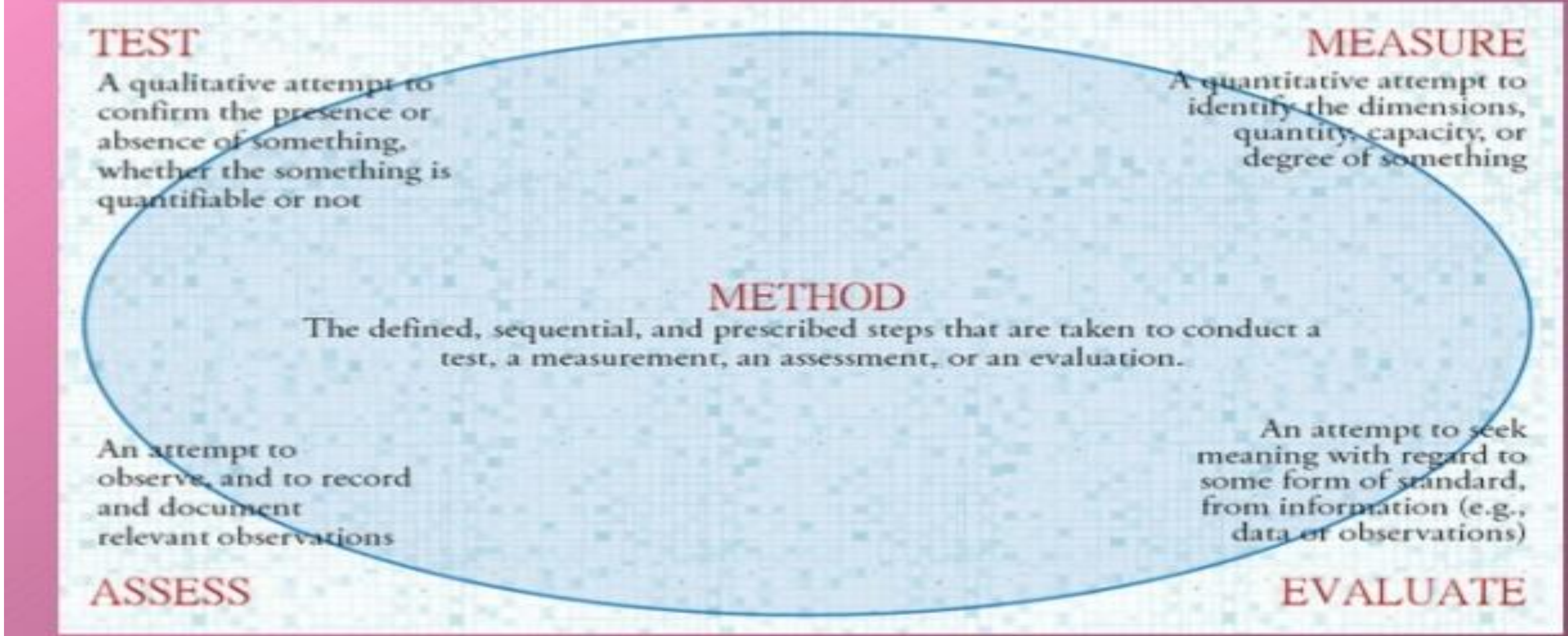
Assessment vs. Evaluation

Dimension of Difference	Assessment	Evaluation
Content: timing, primary purpose	*Formative:* ongoing, to improve learning	*Summative:* final, to gauge quality
Orientation: focus of measurement	*Process-oriented:* how learning is going	*Product-oriented:* what's been learned
Findings: uses thereof	*Diagnostic:* identify areas for improvement	*Judgmental:* arrive at an overall grade/score


Evaluation
Measurement
Testing


EVALUATION
MEASUREMENT
TESTS
1
2
3
4
5

Method Vs. Test Vs. Measure Vs. Assess Vs. Evaluate



TEST

A qualitative attempt to confirm the presence or absence of something, whether the something is quantifiable or not

MEASURE

A quantitative attempt to identify the dimensions, quantity, capacity, or degree of something

METHOD

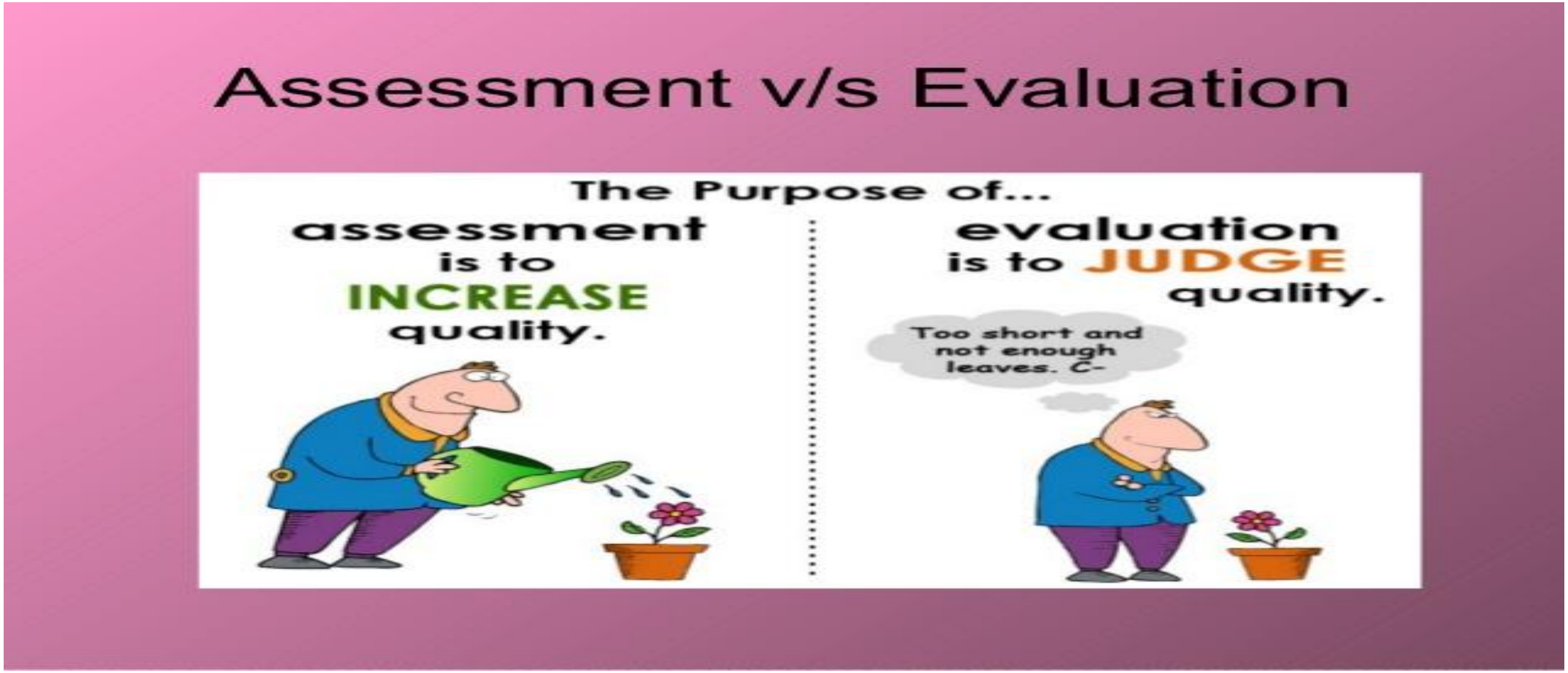
The defined, sequential, and prescribed steps that are taken to conduct a test, a measurement, an assessment, or an evaluation.

An attempt to observe, and to record and document relevant observations

ASSESS

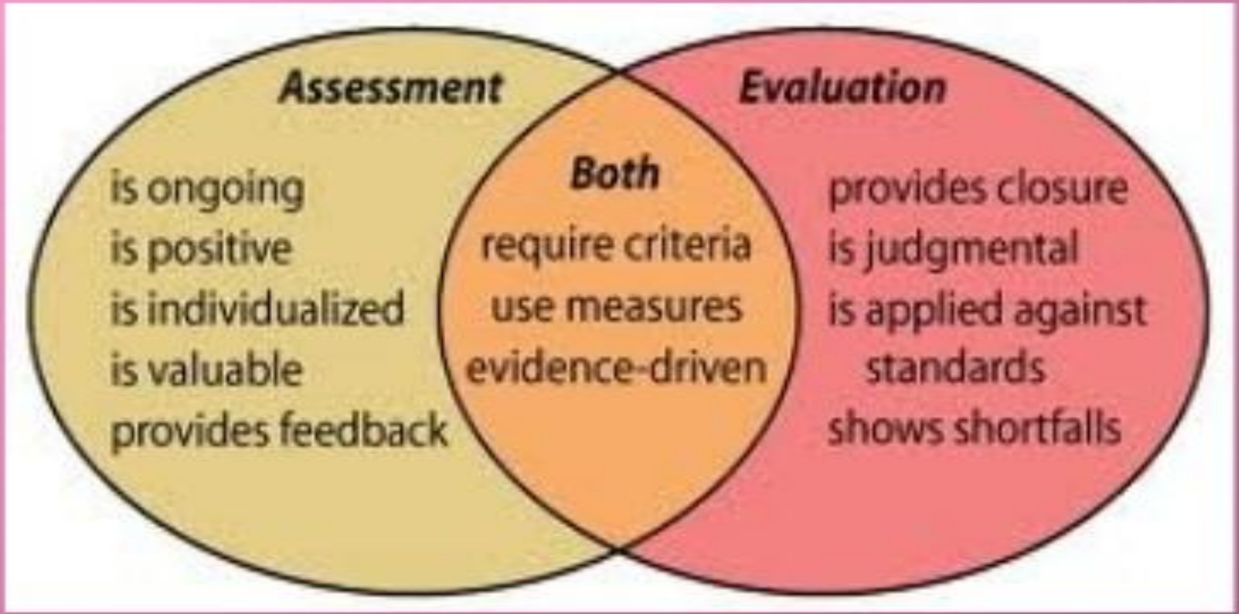
An attempt to seek meaning with regard to some form of standard, from information (e.g., data or observations)

EVALUATE



Assessment v/s Evaluation

The Purpose of...
assessment is to INCREASE quality.
evaluation is to JUDGE quality.
Too short and not enough leaves. C-



Assessment

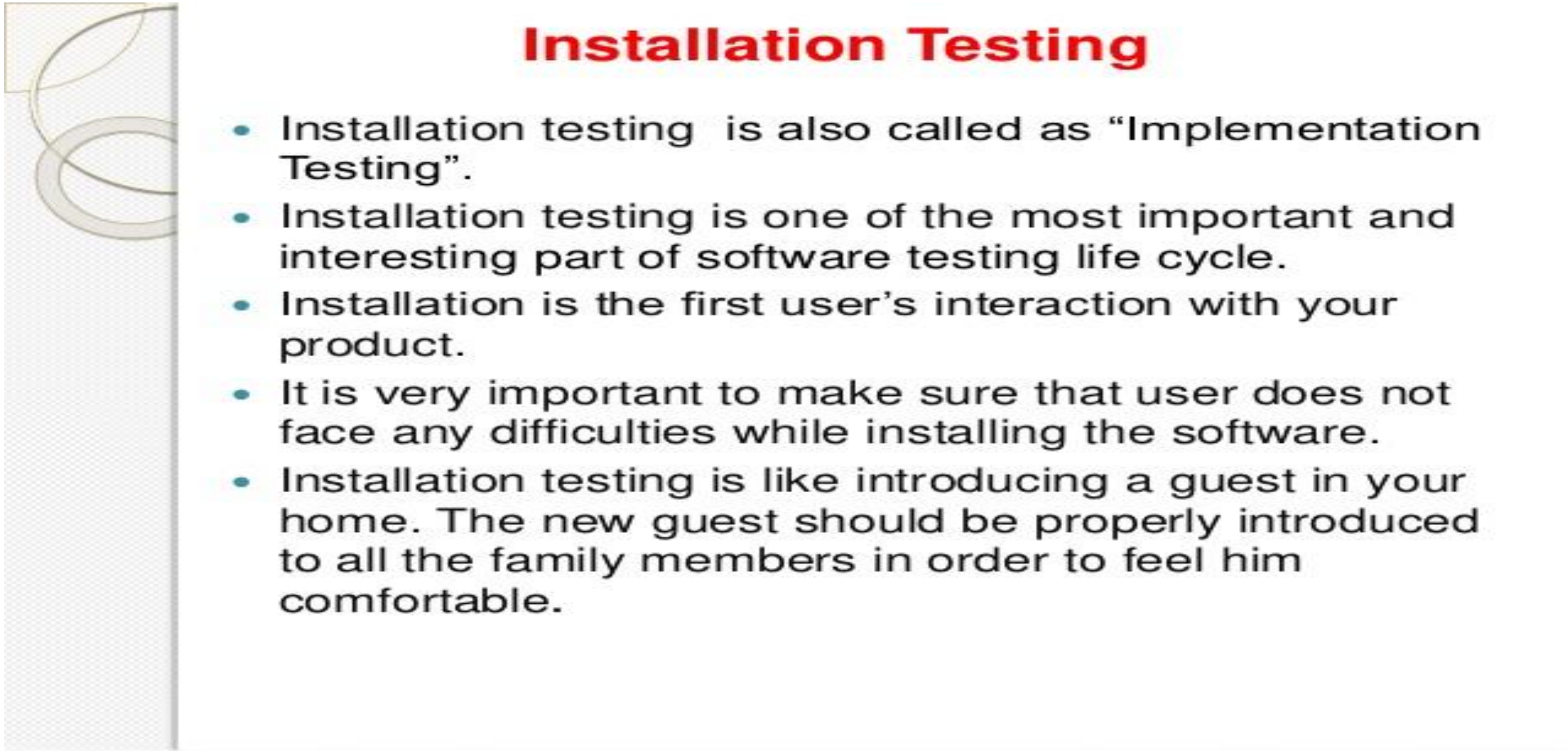
is ongoing
is positive
is individualized
is valuable
provides feedback

Both

require criteria
use measures
evidence-driven

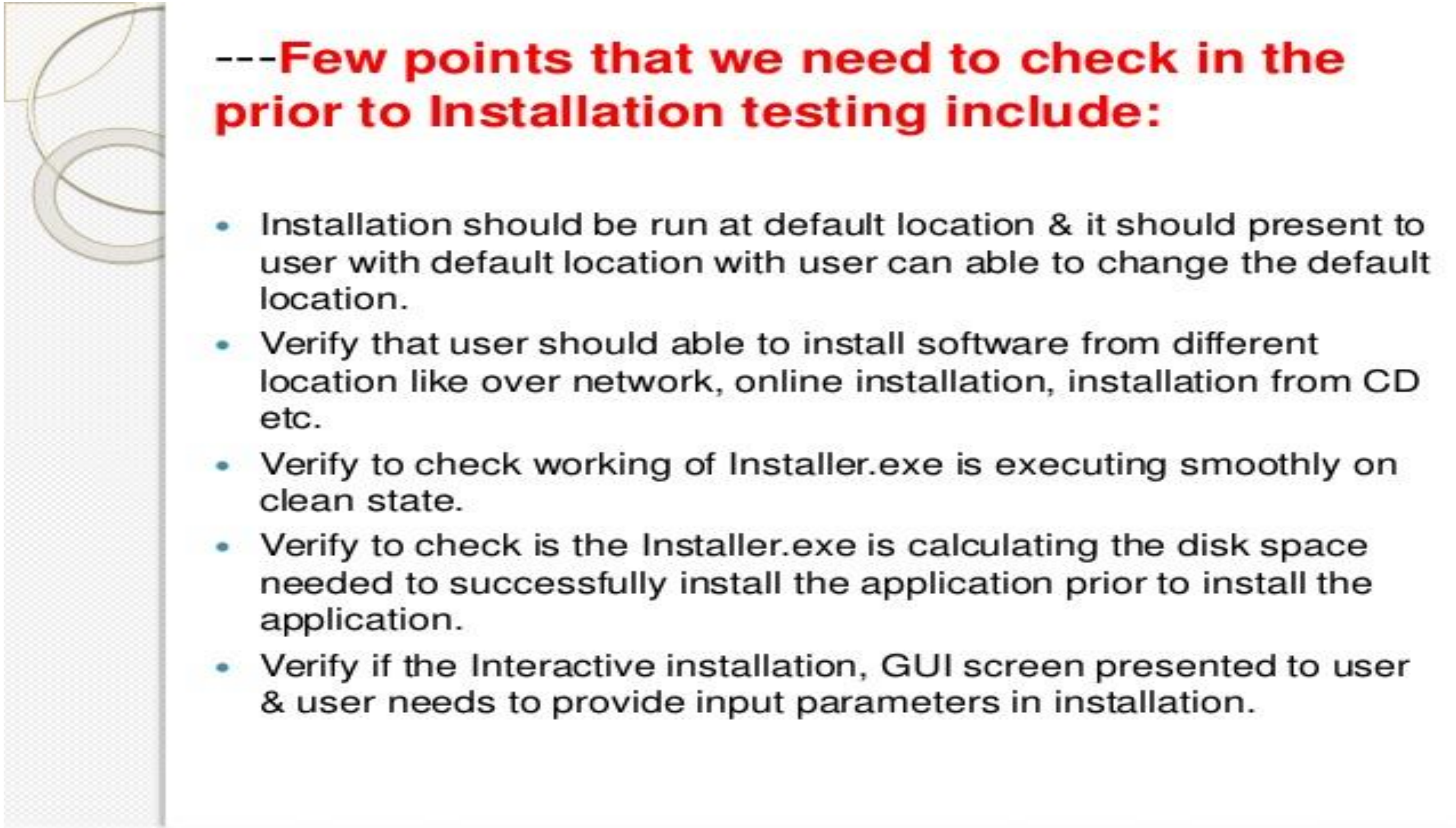
Evaluation

provides closure
is judgmental
is applied against standards
shows shortfalls



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.



---Few points that we need to check in the prior to Installation testing include:

- Installation should be run at default location & it should present to user with default location with user can able to change the default location.
- Verify that user should able to install software from different location like over network, online installation, installation from CD etc.
- Verify to check working of Installer.exe is executing smoothly on clean state.
- Verify to check is the Installer.exe is calculating the disk space needed to successfully install the application prior to install the application.
- Verify if the Interactive installation, GUI screen presented to user & user needs to provide input parameters in installation.

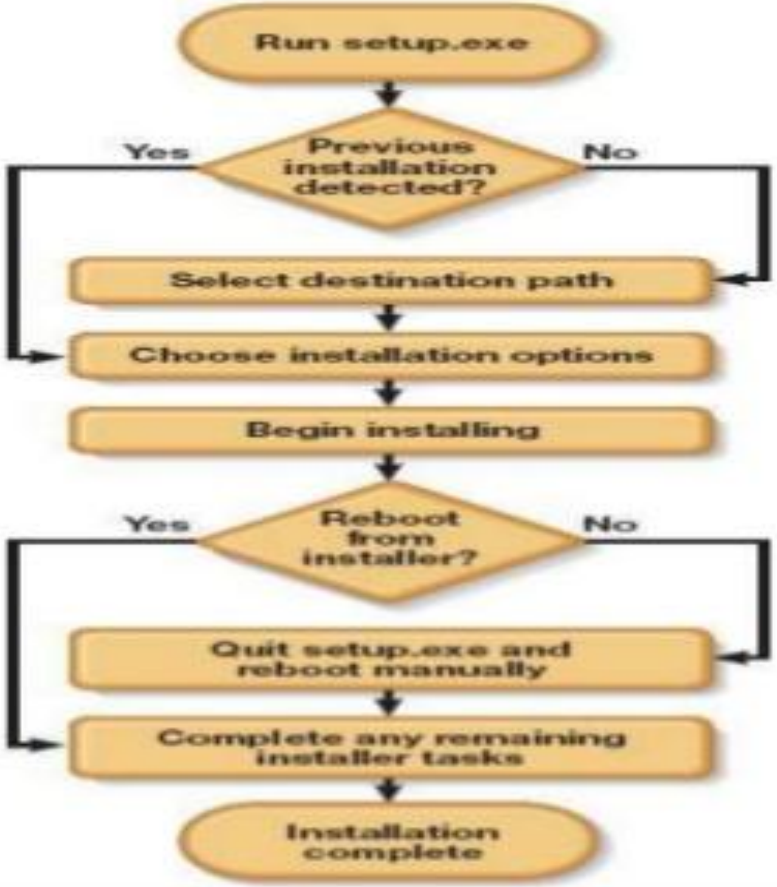
- Verify that after successful installation of software is it working as per mention in specification document & meet user needs.
- Upon un-installation of software application check for the all previously installed files and registry entries are removed or not.
- Verify that is user able to uninstall or repair the software application.

---If your installation is successful on the new system then customer will be definitely happy but what if things are completely opposite.

---If installation fails then our program will not work on that system not only this but can leave user's system badly damaged. User might require to reinstall the full operating system.

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

1) **Install Full Version Of Application:** In case if you upgrading the application or previously installed a basic version now installing the full version on same path then system should allow you to install full version application without any error. Before installing newer version it should display a message on GUI saying old version is detected & do you want to continue.

2) **Automate testing efforts:** The flow chart is useful to create the automated scripts, using this flow chart you can easily make out the automated script for installation testing.

3) **Required Disk Space check in installation**: This is most critical scenario in the installation testing. The Disk space checking is done using automated and manual scripts.

If script is prompting required disk space 1MB, then make sure exactly 1MB is used or whether more disk space utilized during installation. If yes flag this as error.

Test disk space requirement on different file system format. Like FAT16 , NTFS or FAT32 file systems.

4) Use of Distributed Testing Environment: To execute test cases effectively you can use of distributed testing environment.

Distributed environment simply save your time and you can effectively manage all the different test cases from a single machine. The good approach for this is to create a master machine, which will drive different slave machines on network. You can start installation simultaneously on different machine from the master system.

5) Automate the check of files installed after installation: You can use automated script to check that all required files got installed successfully. So you can use this script for checking completeness of installed application.
In manual methods you can check free disk space available on drive before installation and disk space reported by installer script to check whether installer is calculating and reporting disk space accurately.

6) Confirm for registry changes after installation: You have to check for registry changes after the installation of software. This is to check whether expected changes are reflected in registry files. In the market many registry checking software's are available for free.

7) Negative testing in Installation Testing: Intensely try to break the installation process to check the behavior of application under such condition. Such negative test cases should be executed on every step in **Installation testing**.

The most important check is to confirm that whether system recovers to its original state without any issues or not.

Also check system behavior installation in low disk space conditions.

8) Uninstallation testing: You should also check for un-install testing to check whether the user is able to uninstall the application without any error & removing all folders & files relate to installation.

Also need to confirm that it does not impact the other working features in the system which was previously working fine.

Types of Acceptance Testing

- Acceptance testing is a formal testing conducted to determine whether a system satisfies its acceptance criteria
- There are two categories of acceptance testing:
  - User Acceptance Testing (UAT)
    - It is conducted by the customer to ensure that system satisfies the contractual acceptance criteria before being signed-off as meeting user needs.
  - Business Acceptance Testing (BAT)
    - It is undertaken within the development organization of the supplier to ensure that the system will eventually pass the user acceptance testing.

THANK YOU

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