UNIT 4 DESIGN AND DEVELOPMENT OF CBI COURSEWARE - III

Structure
4.1 Introduction
4.2 Objectives
4.3 Testing and Evaluation
  4.3.1 Testing
  4.3.2 Evaluation
  4.3.3 Types of Evaluation
  4.3.4 Revising CBI
4.4 Integrated Unitized Package
  4.4.1 Role of Unitized Package
  4.4.2 Methods of Integration
4.5 Implementation of Software Packages
  4.5.1 Implementation Concepts
  4.5.2 Courseware Implementation
4.6 Let Us Sum Up
4.7 Unit-end Exercises
4.8 Answers to Check Your Progress
4.9 Suggested Readings

4.1 INTRODUCTION

The process of testing, evaluating and integrating software packages is very essential for producing an effective software. Any software developed for a purpose need to be tested for its error free operation. The software has to be evaluated for its operational effectiveness and changes required, if any, are incorporated. In view of the complex nature of software for instructional usage, the development is done on unit basis and they are then integrated together so as to form a complete CBI courseware in specific problem solving area. This is followed by implementation of the software at the users place. The project manager has to ensure that the courseware should not be developed as unsound and useless since it may damage the designer’s reputation.

4.2 OBJECTIVES

After going through this unit, you will be able to:
- understand the importance of evaluation and testing in designing CBI;
- discuss the significance of integrated unitized package in CBI; and
- describe the implementation of software packages in CBI.

4.3 TESTING AND EVALUATION

The development cycle of software involves various stages. Some of these are need analysis, conception or initiation, feasibility study, detailed system study, design, development, testing and implementation. These tasks are performed by various personnel having their expertise in the specific area. Once the system is designed, a blueprint of the same is provided to the development team consisting of programmers. The programmers are responsible for developing programs (codes) according to the design specifications. Once this is done, the designers and
developers together test the system. The objective of the testing is to ensure that the programs are errorfree and there is no logical error in the programs. Once the testing is over the software is evaluated for its usefulness for the category of users for whom it has been developed.

4.3.1 Testing

You all have been familiar with this term. Generally, it means - to examine an element of data, program or any other indicator to ascertain whether some pre-determined condition is satisfied. Testing is a process of carrying out the test on some element of data or program.

The testing can be done for hardware or the software. When a computer is switched on, a testing program is executed which is designed to check the correct functioning of the hardware units of a computer such as memory, hard disk, floppy disk, keyboard, etc. The testing of a software is the process of checking the correctness of the programs and their integration in terms of dataflow, results, etc. The operational functioning of the programs is checked in this stage. When a courseware is developed, it also needs to be tested by the designers to check whether the courseware software is working according to the logic specified, is able to handle the abnormal situations and doesn't stop working abruptly giving some error. For example - incorrect typing errors, when numerical input is required, alphabetic information is keyed in; when zero or a negative number is input; the switchover from one frame to next frame without bypassing the intermediate frames, etc. It is therefore essential to perform a test run of the system. Test run is a test performed on the software system so as to check that the same is operating correctly. This testing process has to be performed before the software is released to the users or to the market.

The testing of programs is generally carried out by using the test data. Test data is the sample input data, covering as many likely and unlikely combinations as reasonable, as to test programs. While using the test data, expected results are also prepared and compared with the results produced by the computer based system. In case of any variation in the two, the program has to be corrected and the process is repeated until the expected and the actual results match.

Sometimes it is difficult to identify the fault or bug in the program. A bug is the fault in a program that forces a program not to function in accordance with the purpose for which it is developed. The process of identifying faults in a program is called debugging. For this purpose generally use is made of certain tools referred to as diagnostic procedures such as Trace, Monitor, etc. These diagnostic routines include programs for producing printouts of memory at various stages of a program and also various forms of analysis of the step by step progress of a program.

In case of computer-based instruction (CBI) courseware, before it is released for the first time, an in-house review is conducted. This is called CBI testing. The designer, programmer and other consultants are involved in the testing of the courseware. They check various conditions, generates test data, produces a schedule of expected results and compares the computerized results with the usual manual results. Some of the important factors tested are: smooth flow of the lesson, the correctness of the feedback given to both wrong and right answers, pleasant looking frames with no typographical errors, ability of the program to provide assistance to the students if extra help is required, etc. Although testing goes along with the development process i.e. testing of each program individually, but the testing of the complete ready courseware is also essential so as to test the linkage of all the programs i.e. overall functioning. Although the development, it should be kept in mind that testing is very crucial. It needs to be handled very carefully as an improper software testing may result in complete rejection of the software. If it is not done properly the software may not be acceptable to the users as the possible errors or incorrect results and their inept handling may discourage the users from using the same.

4.3.2 Evaluation

The last step in the design of a system is its evaluation in comparison with the system it is going to replace. The objective of evaluation is to determine the adequacy of CBI software for the target group users thereby improving the effectiveness of the software. It is determined whether the courseware is achieving the objective of imparting learning in the subject area
for which it was conceived, designed and developed. For this purpose, sometimes, users are also approached to get their views and suggestions (feedback) on the new system. On the basis of this feedback and other evaluation results, the courseware is revised or modified so as to improve the weak areas of CBI.

The evaluation of the courseware is carried out after the testing process is over. In most of the cases, evaluation is performed by the person other then analyst and is generally carried out by the subject-matter expert and the users.

4.3.3 Types of Evaluation

A thorough courseware evaluation includes four components:

- **Structural Evaluation** — It is an assessment of the structure and appearance of the lesson. It is useful in predicting how well the courseware will teach without actually having students test it. This evaluation identifies weaknesses for the lack of effectiveness.

- **Functional Evaluation** — It tests how well the courseware teaches. In this a student’s skill level is measured before and after completing the course to determine how effective was the courseware. This component in itself is not adequate but is very important in the evaluation process.

- **Cost-Effective Evaluation** — This aims at evaluating the worth of the courseware i.e. if a courseware is worth the money it requires to acquire. Weighing costs against training effectiveness leads to wise decision making.

- **User-Opinion** — Students and teacher opinions about the courseware needs to be assessed. If the users do not like the courseware and find it boring for some reason, the courseware will never succeed, no mater how effective it may be.

Courseware evaluation should be conducted at least twice. These are known as formative and summative evaluations. You have read about formative and summative evaluations in details in the course ES-333 Educational Evaluation. The first evaluation should be conducted to guide courseware revisions before the courseware is released. These are *formative evaluations*, conducted during product formation. User opinions are also obtained apart from structural evaluation. The designer then corrects the weaknesses of the courseware before releasing the product. The second evaluation called *summative evaluation* is conducted with potential courseware purchasers, potential users and courseware researchers. The data from this evaluation guide purchase and use decisions. It aims at identifying if the product i.e. courseware is acceptable, or is better than other products.

4.3.4 Revising CBI

By conducting various evaluations of the CBI courseware, as much evaluation data as possible is collected. On the basis of this data, the designer determines what type of revision of courseware is required. The various types of revisions are:

- To improve function — to make it teach.
- To Improve Opinions — to be acceptable to users.
- To improve cost-effectiveness — to be affordable.

---

Check Your Progress

1. Write your answer in the space given below:

i. Distinguish between testing and evaluation.

ii. Compare your answer with those given at the end of the unit.

2. List various types of courseware evaluation methods.

3. What is the purpose of evaluation of a courseware?
A package is a software developed so as to serve a specific purpose(s) e.g. word processing package, spreadsheet package, database package, accounting package, inventory package, etc. A package in the earlier time used to be developed as a single module i.e. all the functions provided by developing only a single program. This approach was difficult to develop, modify, upgrade in future, and document and even a single fault could make the entire package inoperative. These were required to be developed by a single individual and hence more time required for development. As a result of this, this problem was improved by adopting the modular approach in which the entire task was divided into separable units for independent development and testing by programmers. Once all the modules are developed, they were integrated together so as to form a complete software or package. This approach overcomes majority of the shortcomings of the earlier approach to software development. For example, word processing package is a collection of programs linked together so as to perform the word processing activities like text entry, formatting, editing, spell check, thesaurus, forming tables and printing of nice looking documents or reports. For each of these functions, a separate program is developed. These programs are linked or integrated together so as to provide smooth data exchange between various programs giving rise to a single package through which all these functions can be performed.

The term **Unitized package** refers to a package which is developed using the modular approach by dividing the subject knowledge into self contained independent learning units. These units are then taken up for development of CBI units. Each unit is provided with its development specifications. These specifications help the designer to easily integrate these units into a workable CBI for a subject area. For example, solution of linear equations could be one unit, solution of quadratic equations could be another unit, under the subject of algebra. This concept is treated as a unit in CBI development process. Some of the information provided in the specifications are: the topic of unit, curriculum area, what aspects the topic will cover, sample questions which can be answered after studying this unit, comparison with traditional mode, program description, sample dialogue, student model, etc. The word "integration" means combining of two or more parts into a single one. When this term integration is used in reference of software package then it means that two or more applications are combined into a single package.

### 4.4.1 Role of Unitized Package

In order to ensure that learning occurs in a specific subject area, it is essential that the material should be presented in such a way that the students can navigate within the CBI lesson according to their needs. This will need availability of a smooth flow between various independent units in terms of frames, screens, and topics within the CBI. The independent CBI units can be integrated in various ways. It could be either on the basis of subject relationship, ease of use, order of complexity, etc. This flexible integration facility makes the user feel as if the CBI lesson is some sort of a package tailored to their needs.

### 4.4.2 Methods of Integration

Methods for integrating software package are given below:

**All-in-One Package:** This is a commonly used method for integrating the software meant for mass utilization. The all-in-one package combines the common applications into a single program, for example-Frameworks by Ashton Tate, Symphony by Lotus Development and MS-Office by Microsoft. These packages are having the facility of using the spreadsheet with graphics, word processing and the database functions and vice versa. The user can conveniently move from one application to another as all are components of single package. Most of the all-in-one packages share the common user-interface.

**Integrated Series:** These are separate applications which can be installed individually but share common instructions and storage patterns. There is a quick and easy transfer of data from one application to another. The integrated series software offer the varied functions and data transfer capabilities of stand-alone programs unlike the high RAM requirement of all-in-one packages. Smart Software by Innovative Software is example of this method.

**Background Utility Programs:** This provides limited integration capabilities. Desk accessories
Desk accessories may be loaded into RAM e.g. accessories like calculator, calendar, telephone
dialer and notepads, etc. are loaded along with the main program. Once the utility software is
loaded then a stand-alone program can be loaded into RAM. Due to extra usage of memory
for the accessories, it may not be possible to load some of the applications due to insufficient
primary memory.

Check Your Progress
Notes: (i) Write your answer in the space given below.
      (ii) Compare your answer with those given at the end of the unit.
1. What is a unitized package?

5. Why do we need to integrate there?

4.5 IMPLEMENTATION OF SOFTWARE PACKAGES

Implementation of software is a very crucial stage in the development cycle of a CBI courseware.
A carefully planned and executed implementation result in success of the courseware and its
acceptance by the users.

4.5.1 Implementation Concepts

Implementation is the stage of the system when the theoretical design is turned into a working
system. Implementation is concerned with those tasks leading immediately to fully operational
system i.e. placing the completed and tested system of hardware and software into the actual
work environment of the users. Although, it involves system designers, programmer, users and
operating management (teachers) but the sole responsibility lies with the system (courseware)
designer. This is a very complex and time consuming task and need to be properly planned.
If not controlled properly, it may lead to chaos as the successful completion of this stage
determines the acceptance or rejection of a system.

The actual tasks performed during the implementation phase varies from project to project and
there is no general consensus on this. Sometimes the system testing is also considered a part
of implementation process. Even acquisition of hardware, software, site preparation, setting up
of files, etc. is considered under the umbrella of implementation. The implementation stage
involves the following tasks:

- Education and Training
- System and File Setup
- Changeover
- Review and Amendments

An implementation plan is required to be prepared before beginning this exercise. To achieve
the objectives and benefits of the designed system, it is essential for the users to be confident
in using the system. This is achieved by educating and training them on various aspects of the
system such as objectives of the system, design logic, operation, error correction, etc. All the system documentation consisting user manuals, procedure manuals, computer operating instruction and security procedure should be made available to them. System setup is the process of installing the software package (courseware) on the available hardware at the user's site. The installation will create appropriate directories, icons and make necessary changes in the system software so as to enable the courseware to run smoothly on the computer. This will also setup files required for running the courseware software. The changeover from existing system to new computer-based system may take place when:

a) the computer-based system has been proved to the satisfaction of the users (students and teachers) and the necessary setup has been completed.

b) users (teachers) are satisfied with the results of the system test, training and documentation.

The changeover may be achieved in a number of ways: The commonly used methods are:

- **Direct changeover** - complete replacement of the old system with the new. It is bold, risky and needs well supervised implementation.

- **Parallel changeover** - simultaneous running of old and the new system for some time to cross check the results. It is safe but more expensive as two systems have to operate.

- **Pilot changeover** - similar to parallel running. The data for the previous periods from the existing system is used in the new system and the results are compared. The new system results are acceptable, the system is accepted for installation.

- **Stage of phased changeover** - series of limited size direct changeover, i.e. new system is inducted piece by piece. A logical part is implemented on the new system and the other parts continue on the old system. Only when the logical part is acceptable, the next logical part is taken up by the new system.

The implemented system needs to be reviewed (for performance) periodically for changes in the environment, technology, user needs, etc. This may provide changes to be incorporated into the system either to correct the errors or improvements. The review is generally performed by a process called System Audit. The audit leads to appropriate amendments or modifications in the operational system so as to take care of the changes desired.

### 4.5.2 Courseware Implementation

Courseware implementation means making the courseware operational at the user site which may be a school or any other training institution or may be a home. It should make sure that the courseware has been installed and is running on the hardware. It also involves the training of the user or instructing personnel for operating the same. The designer/developer has to initially sit with the users to make the software fully operational. Either developer may add/delete or modify certain things on the request of user at the initial stage or the suggestions/feedback given by the user may be considered to incorporate in the next version of software.

It is desirable that a CBI designer should follow up the courseware so as to see if her users continue to be satisfied, i.e. study the user satisfaction. In practice this rarely happens for various reasons. However, the users should be encouraged to provide the feedback to courseware producers for future modifications or revisions. For example, many textbooks contain a feedback form for the readers.

<table>
<thead>
<tr>
<th>Check Your Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes: 1) Write your answer in the space given below.</td>
</tr>
<tr>
<td>2) Compare your answer with those given at the end of the unit.</td>
</tr>
<tr>
<td>3) What are the tasks generally performed during the implementation stage of software development?</td>
</tr>
</tbody>
</table>

---

Design and Development of CBI Courseware-III
4.6  LET US SUM UP

Testing is the process of executing a software package to find out the errors or bugs, if any, before releasing it to the users. Testing attempts to make the courseware software free of any programming errors. Courseware evaluation follows the testing process in order to make the courseware more effective. This is done twice. First in the form of formative evaluation and secondly as a summative evaluation. The evaluation helps in revising the CBI courseware to increase its utility and effectiveness. The CBI courseware is developed as small units covering individual concepts and are called CBI units. These units are then integrated together so as to provide integrated CBI software in a specific subject area. Implementation is the last stage in the CBI development process. It involves various tasks which actually varies from project to project. The commonly performed tasks are: Education and Training, System Setup, Changeover and finally Review and Amendments to the courseware to ensure that the objective is continued to be met.

4.7  UNIT-END EXERCISES

1. Select any two software packages available in your school or you have come to know. Compare both of them with respect to the four major components of evaluation. Discuss your observations from your peer colleagues in the school or at the study centre. Write a report.

2. Organise a brain-storming session in your study centre on the “Integrated Unitized Package” and discuss available software package related to the school subjects, you have been teaching. Write a report highlighting the software and its detailed description on its role and methods of integration.

4.8  ANSWERS TO CHECK YOUR PROGRESS

1. Testing is done before releasing the software while evaluation is done after the implementation. Testing is done by the designer and programmers while evaluation is done mainly by the users. Evaluation results in revision of CBI.


3. Purpose of evaluation is to make it more useful, effective in terms of functions and cost by revising the CBI courseware.

4. A unitized package is an independent CBI courseware unit (CBI unit) which can be used to teach a concept.

5. Integration helps in bringing together various CBI units so as to provide learning in a broader subject area as compared to smaller independent concepts.

7. The implementation of the system is the sole responsibility of the designer/developer.

8. Parallel changeover is safest, but more time consuming and more expensive. Direct changeover is highly risky, least time delay but less expensive as compared to parallel. Pilot changeover is safe, less time consuming as compared to parallel changeover. The phased changeover is less expensive and safe but most time consuming. The designers draw advantage from each stage which helps in better and successful implementation of the subsequent stages.

4.8 SUGGESTED READINGS


Student's Feedback Form

Dear Students

Welcome to opting this course in your B.Ed. programme. This course has been identified considering the potentialities of computer in teaching-learning and other administrative situations. This course has been developed in an interactive and self-learning manner. As this is the first version, you have been requested to give your constructive and valuable comments towards improvement of this course. Your comments on each unit and block are sought.

Thanking you,

(DR. VIBHA JOSHI)
Course Coordinator

Address for Communication

Dr. Vibha Joshi
(Course Coordinator)
School of Education
IGNOU
Maidan Garhi
New Delhi - 110 068