

# Evaluation of Multimedia

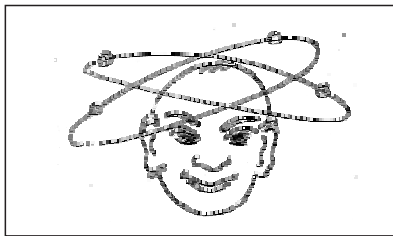
9



## Objectives

At the end of this section, you will be able to

- Describe both formative and summative evaluation of multimedia;
- Use appropriate evaluation methods and techniques;
- Undertake evaluation of prototype lessons; and
- Build a case for continuous evaluation for product improvement.



There is a remarkable growth in the interest and use of multimedia programmes besides the growth in skill in developing multimedia programmes. There is an ocean of content on the Internet, developed without really worrying about the quality of it. To design and develop a quality programme, we must spend some time to understand the qualities of a good multimedia programme and a website. As we discuss the evaluation of multimedia programmes below, we shall limit our discussion to the educational software, because the objective of evaluation and the process of evaluation of various multimedia programmes shall vary substantially.

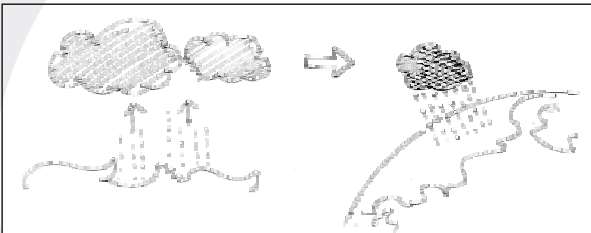
## Evaluation: Test of the Pudding

Evaluation is testing whether a multimedia programme fulfills the objectives set, and suggesting improvements it requires to make the programme useful for its target audience. Evaluation is not a uniform process and evaluation cannot be identical for all programmes. Evaluation invariably will have to be made of the objectives that the multimedia software wishes to fulfill. For our purpose, we may say, evaluation will involve testing of the content it transacts vis-à-vis the target learners, keeping in view the prime objective of the expected learning which may take place in the learners after they go through the programme.

At the broadest level, evaluation of a programme should be done at two levels: at the level of *content* and at the level of *technology* employed. Technology plays the role of only the means to attain the identified objectives. Optimal use of technology is desirable and in evaluating multimedia software it needs to be kept in mind that the technology itself should not become too cumbersome for the users, because the competence level of individuals using technology varies a lot.

Computers today can provide unlimited facilities for search, navigation, print etc. but it is dangerous to employ all the facilities without proper understanding and navigating skills because the learners may have to spend more or most of the time learning about these. Ease of use of the technology and its optimal use should be paramount on the developers' mind. Developers should refrain from using all that they know or have ever used and should primarily focus on the feasible and practical aspects from the points of view of the course objectives and the learner competence in using the technology.

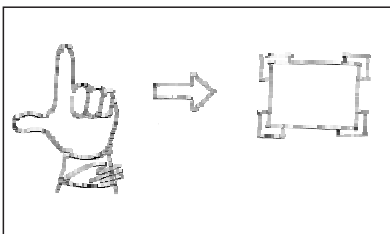
In this section we will explore how we may conduct evaluation of multimedia programmes.



### **Inherent Nature of Multimedia**

As we know, a multimedia programme is a single platform presentation for developers and access for users of various media like the text, picture, sound, video, animation etc. But, just putting all these in one CD would not qualify it to become a multimedia programme. If a software stores information in two media like the text and the picture, it is no different from a printed book, or if a software stores information in picture and sound forms, it is no different from a film. We cannot call programmes with information in just two media a multimedia programme. The facility to store information in various media is innate to technology. It is quite often noticed that software developers cannot resist the temptation to employ every facility that the system provides or of which they are aware.

Media communicates through its own 'symbol system'. Sound may communicate the same message, either through radio broadcast or through a cassette tape or through a CD, but the sound, in conjunction with a picture and text displayed on the screen of a computer, may acquire a different 'symbol system' or meaning. The impact of the sound, in conjunction with any other medium, must be seen in conveying much more than what both the media (i.e. sound and picture) could have done independently. The inherent strength of multimedia lies in the power of integration. Moreover, a television/video programme can also have all the components of multimedia like the text, pictures, graphics, audio, movies etc. What is special to multimedia is its *interactivity*, which is limited or mostly absent in video.



### **Understanding Learning and Learner**

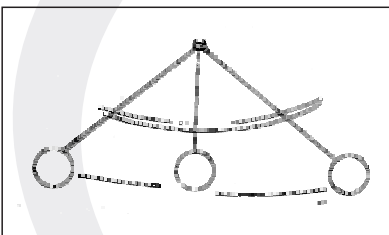
Every learner has his/her own way of learning; pace varies, timing varies and so do various other habits. Computers have provided us the facility to adapt these facilities according to one's own needs and liking. A major quality of multimedia is interaction, which the machine performs with the learners. So, while designing multimedia software it has to be paramount on our minds that the learner should be able to interact with the programme and make it suitable to his/her requirements and liking.

A major tenet of multimedia education is the acceptance of the fact that no single method or medium is appropriate and perfect for all individuals. Every individual has his/her own preferred methods of information reception and processing. The ideal

condition would be to provide numerous options to learners. However, if numerous options cannot be made available, a number of alternative instructional options based on various learning approaches must be given. This is what a multimedia programme offers.

### **Types of Evaluation: Formative and Summative**

There are certain decisions, which need to be taken before starting the process of software development. These decisions guide the process of software development. In fact these decisions become the guiding principles and the software developers have to keep asking the questions, if they are fulfilling the objectives decided upon in the planning process. Once the prototype of the software is ready, it is tested and once again the objectives can be re-examined and reformulated on the basis of feedback and evaluation. Evaluation is basically of two types: formative and summative. We shall discuss them below:



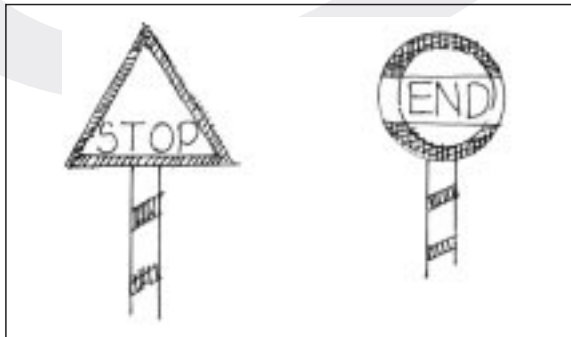
#### **Formative Evaluation**

Formative evaluation is done as a continuous process in the development of multimedia and even before the development process actually starts. Decisions taken at the beginning of the process of software development affect various aspects of the software. Answers to questions like who, why, where, and how become the guidelines for the development of the software. Depending upon the time and resources, both quantitative and qualitative methods of feedback are utilized in formative evaluation. No programme can fulfill all requirements of all learners. In fact, if a single programme can provide all the information, and answer all queries on a single topic, it should be considered successful. So, every software developer must decide and delimit the scope of the software beforehand. In other words, we have to spell out the objectives of the programme. Some questions like the following ones need to be answered because these will affect the content and the selection of technology.

- (i) Who are the target users of this software and what is the level of the target users?
- (ii) What is the level of computer familiarity expected of the learners?
- (iii) What would be the objective (in terms of content) to be covered by the programme?
- (iv) How will the programme be used?
  - a. as supplementary to classroom teaching?
  - b. as independent programme providing complete courseware?

Once decisions on the above are made, the cognitive aspects of learning are to be considered. The socio-cultural background of the target users will affect content selection and treatment of the topic. In arranging the content, the learning habits of the users would influence the decision about sequencing of information, quality of information (through various media) provided through every screen, and supplementary information to be provided through links.

It is often mentioned that no programme should aspire to be complete or self sufficient, but it should lead the users to relevant information. The software developer should provide relevant links and the content expert should word the text and place other relevant media material in a manner that they raise more inquisitiveness in the learners and they go for further search and self study.



### Summative Evaluation

After the completion of the development of the programme, the software is released for use. The actual users, then, make suggestions and these suggestions form the basis of summative evaluation. Summative evaluation is the end of the project evaluation. Some programmes involve a number of teams to develop various components of larger software, which finally integrate into the end product. While working with Microsoft Office you must have used the numerous facilities it offers. For example if you click on the Accessories it provides Calculator, Games, etc. These must have been developed by different groups separately and then finally joined to make a final product. Summative evaluation, as mentioned earlier, should be conducted for each of the components and also for the final product.

As summative evaluation is targeted at the end-users, in educational software a major focus of study is the software's *pedagogic effectiveness*. The teaching-learning objectives identified during the needs analysis become the base of summative evaluation of academic software. Evaluation should keep the educational, entertainment, ease of use or design features in mind while conducting summative evaluation of these and then overall impact on learning.

Suggestions on the basis of summative evaluation may be for (i) short-term and (ii) long-term changes in the programme. Short-term changes may be based on your own observations and the feedback from the users and the long-term changes may be made on the basis of the decisions of the curricular design and on the basis of suggestions given by the development agencies and the organisations using your software, if it is being used outside your institution.

## **Issues in Evaluation**

Development of educational multimedia can be viewed at two levels: conceptual and presentation levels. Conceptual design involves unifying the scattered knowledge on the selected area/topic, and creating a learning web. Presentation design deals with the realization of conceptual framework into a multimedia programme, which runs on a computer.

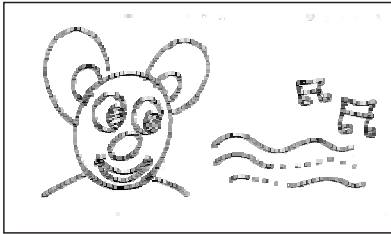
As we have a number of media available for presenting our content, it is often noticed that designers get tempted to use as many of the facilities like screen layout, color schemes and the detailed use of individual media like the moving pictures, audio, text etc. The objective of the whole effort should be to integrate elements of multimedia into a deep learning architecture.

Although the design of interactive multimedia material should be consistent with theories of both learning and teaching (already discussed in Section 4) there is a wide range of opinions on what constitutes 'good' interactive learning material, and consequently there is room for subjective judgment and for creativity and innovation. Rather than taking a prescriptive approach to design content, presentation and interaction, we consider the issues involved. There are four basic issues namely (i) educational effectiveness (ii) entertainment value (iii) user friendliness of technology and (iv) design features, which must be taken care of while designing multimedia software.

### **Educational Effectiveness**

You would agree that the goal of all educational programmes is to have high academic or pedagogic value. Some of the questions that should be asked are:

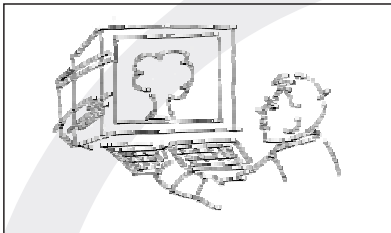
- Will the software meet the educational objectives and offer good presentation of the content areas?
- Is the software sound in terms of teaching principles and visuals to enhance the achievement of the programme's educational objectives?
- Does the software provide higher order thinking skills?
- Is the content presented in simple and neutral of gender, without ethnic and religious biases?
- Does the software offer simple, precise directions accompanied by picture choices and voice responds to the learner's own rate of learning?



### Entertainment Value

One of the major objectives of teaching through multimedia programme is to provide edutainment to the learner. And, also the main reason why multimedia is popular is its capacity to enable learning without slogging. As you design your software, you must be careful and see that the programme offers learning opportunities in a 'fun-learn' environment. Ask yourself the following questions to satisfy yourself whether your programme fulfills the criteria:

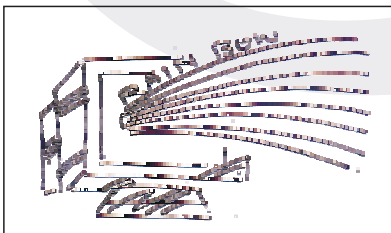
- Does the programme ( in sound and graphics) provide learning in an enjoyable environment to the target learners?
- Does the programme provide adequate flexibility to 'surf' i.e. move around and learn at one's own pace and convenience?



### User Friendliness of Technology

In teaching-learning activities dependent on technology, the main hurdle faced by the participants is the lack of or limited familiarity with the technology. It should be our endeavor to visualize all the queries that users may have, and provide icon-based information for ease of use. Target users can develop the skills to use the programme within reasonable time and independently use it after the first use. We may ask the following questions to test, if our programme provides the ease of use:

- Is the selected platform commonly available and easy to use?
- Can the learners review the sections they have read/viewed and take a test on these?
- Can learners print/save desired information?
- Can 'last action' be cancelled?



### Design Features

Let us be aware and conscious that we use facilities not because they are available but because they are essentially required for communicating the concept. For example, most softwares these days provide the facility to include moving objects and sound. These can be used for conveying information but such technologies are viewed by some as distractions which may be dispensed with. However, we can find concepts and information for which these features may be crucially required.

- Are the pages pleasing to eyes and also contain no items which may offend any user?

- Does the design have icon-based features?
- Can sound and video be played smoothly and adjusted or muted at will?

### **Developmental Testing or Alpha and Beta Testing**

Unlike the terms 'formative' and 'summative' evaluation, which are borrowed from the educational research, the terms commonly used by software developers are 'Alpha testing' and 'Beta testing'. These are frequently used terms with standardized procedure and meaning. Also, there is no 'wrong way' of using a software and the testers/users should be left to use it in the manner they find most suitable. During the testing procedure we may try to track the most favorite or natural way of browsing followed by the users. Anything that they may be doing 'incorrectly' would help us identify areas which need redesigning. We shall explain what these terms denote in the process of software development.

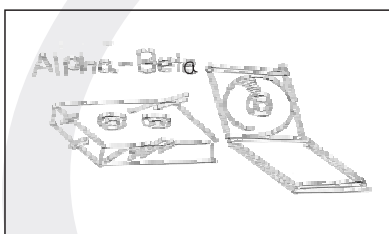
#### **Alpha testing**

Alpha testing is conducted when the prototype of the software is ready and the software can take inputs and generate outputs. At this stage the software is not fully functional and so the software is normally not sent to the end users but tested inside the organization, in our context, on the peer group. The software is run on a different machine within the organization so that, if there are some bugs, they can be detected and eliminated. The software is tested for aspects like *navigation*, look, and *feel* of the software. Alpha test informs us about the tools that worked and tools that did not work. We get the perspective of the content experts and the designers (in this case both may be you) and the learners.

Alpha testing becomes particularly relevant when a number of developers are involved in developing various modules and when these are integrated; it becomes important that they work in conjunction. Sometimes, various modules developed by various developers do not integrate. In developing software single handed the objective of alpha testing should be to find if text, voice, picture etc. integrate well and pop-up as and when desired, and do not interfere with learning.

#### **Beta testing**

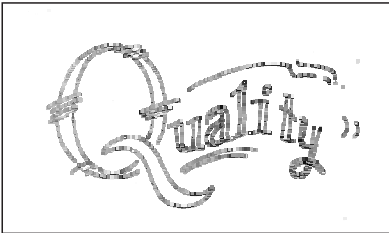
It is very important to eliminate defects as soon as they are detected. However, it is always boring to eliminate defects and developers often enjoy adding new features. The proverb 'a mend in time saves nine' is never more true than in this case. Every





time problem elimination is postponed, the problem is slightly increased because it would not be fresh in mind the next time. Precisely, the functional specifications and the source code are reviewed at the earliest with the objective of eliminating defects before they start to waste cycles.

The software is sent to people outside the organisation or end-user for their review. Outside reviewers will be able to reflect on the requirement of the software at this point and make relevant suggestions. At this stage testing for software reliability, installation and documentation are done as well.



### **Evaluation as Continuous Process of Quality Improvement**

Evaluation of multimedia software cannot be and should not be a terminal or one-time activity. It has to be an on-going and continuous process. As new information get generated and demand for information is made, provision for incorporating new information may be a condition for evaluating a multimedia software. Multimedia software, which is rated highly today, may be rated poorly after only a few months, if it is not continuously updated. This would be particularly true in the case of growing disciplines.

Developments in technology is also an important issue. As a new technology comes up and people start using it, any programme not providing such a facility would not be appreciated. To keep a programme floating and popular new options and facilities need to be incorporated. Sometimes, some of the basic design concepts and the architecture of these older programmes may well appear to have been superceded by recent developments. Users who may be familiar with the features that are being replaced, and so, replacing them may demand users to learn new features and unlearn the old ones. New features do not always result in improved functionality. For the sake of simplicity and elegance of design, older features could be resuscitated.

For the above reasons multimedia software needs to be update at short intervals with the help of content experts and software engineers.

It is said 'practice brings perfection' and this is entirely true of multimedia development. In this section, we have mentioned under different sub-sections the various issues in software evaluation, including formative evaluation, summative evaluation, beta testing, alpha testing and the need for continuous improvement in multimedia development.



**A suggestive instrument for evaluation for Multimedia**

<b>Questions</b>	<b>Very useful</b>	<b>Useful</b>	<b>Not useful</b>
Did you find the information that you wanted?			
Do you think the time spent in processing the information was worth it?			
How useful is the programme for learners using it in isolation?			
How useful is the programme for group learning?			
How easy was the use of the programme after the first demonstration session?			
Did the programme provide helpful on-line information when required?			
Were navigation tools provided through menus and icons readily available?			
How easy was it to cancel the last move?			
How easy was it to 'fly' or print or save selected items?			