

Assignment 8 – Creating My Own Evaluation Criteria

Mark Rash | markrash@vt.edu | April 23, 2010

Activity 1

| Checklist Description | Mnemonic or Evaluative? | Why? |
|--|--------------------------------|--|
| Task checklist in Outlook – used to track school assignments and work responsibilities | Mnemonic | Does not evaluate the quality of work; serves as a task reminder and aid for prioritizing actions |
| Shopping list | Mnemonic | Does not evaluate quality or value of products; serves as an aid for remembering to purchase everything needed |
| Instructional Design Peer Review Form – informal checklist used for review of a completed Instructional Design project | Evaluative | Serves as an evaluation of a completed Instructional Design project; allows other ID experts to review a project before moving into formal evaluation phases |
| Project Closedown Checklist for Instructional Development | Mnemonic | Does not evaluate the quality of the project; ensures all tasks for completing a project have been completed |
| Grading Criteria used on every ITMA assignment | Evaluative | Serves as an evaluation as to the completion of various task components and their quality |

Software Evaluation Checklist Part 1

Approach: Devise a checklist that would be used to determine if an existing software package is meeting your instructional needs.

- User Interface and Navigation
 - Aligns with Operating System standards
 - Navigation is intuitive
 - Complete help documentation is provided
 - Menus are clean and organized
 - User is informed of progress in the program
- Aesthetics
 - Screens are clean and organized
 - Text clearly contrasts from background
 - Fonts are legible
 - Colors and fonts are consistent throughout the program
 - Screen layouts generally follow the “rule of thirds”
 - Graphics are appropriate for the topic at hand
- Content
 - Content is chunked into manageable sections (5-7 items within a topic)
 - Content is labeled with headings and/or sub-headings
 - Material is accurate
 - Material is up-to-date
 - Content is relevant to the audience
 - Content is relevant to the objectives
 - Content is appropriate for the audience
 - Content is presented at the appropriate depth
- Technical Qualities
 - Technological requirements clearly specified
 - Technological requirements reasonable for the program’s purpose
 - Program is free of errors
 - Load times are reasonable for the type of content being presented
- Pedagogical Qualities
 - The audience is defined
 - Learning objectives are clearly defined in performance terms
 - Instructional context closely resembles the performance context
 - Strategies are incorporated to motivate the user
 - Summaries and/or reviews are provided
 - The audience is provided with opportunity for practice
 - Immediate feedback is provided
 - Feedback is detailed
 - Assessments are present and connect to learning objectives

Software Evaluation Checklist Part 2

Indicate Yes or No for each criterion of merit. Also note any comments, as applicable. If existing software does not score well in the first section, it may not be meeting your instructional need(s).

| | | | |
|--------------|--|-------------------|--|
| Program Name | | Version | |
| Description | | Intended Audience | |

| Criterion | Yes | No | Comments |
|--|-----|----|----------|
| Pedagogical Qualities | | | |
| The audience is defined | | | |
| Learning objectives are clearly defined in performance terms | | | |
| Instructional context closely resembles the performance context | | | |
| Strategies are included to motivate the user | | | |
| Summaries and/or reviews are provided | | | |
| The audience is provided with opportunity for practice | | | |
| Immediate feedback is provided | | | |
| Feedback is detailed and meaningful | | | |
| Assessments connect to learning objectives | | | |
| User Interface and Navigation | | | |
| Aligns with Operating System standards | | | |
| Navigation is intuitive | | | |
| Complete help documentation is provided | | | |
| Menus are clean and organized | | | |
| User is informed of progress in the program | | | |
| Aesthetics | | | |
| Screens are clean and organized | | | |
| Text clearly contrasts from background | | | |
| Fonts are legible | | | |
| Colors are consistent throughout the program | | | |
| Fonts are consistent throughout the program | | | |
| Screen layouts generally follow the “rule of thirds” | | | |
| Screen reads in the direction common to the culture of its intended audience | | | |
| Graphics are appropriate for the topic at hand | | | |
| Graphics are clear | | | |
| Multimedia content is used effectively with specific purposes | | | |
| Multimedia content is of good quality | | | |

| Criterion | Yes | No | Comments |
|--|------------|-----------|-----------------|
| Content | | | |
| Information is chunked into manageable sections (5-7 items within a topic) | | | |
| Content is labeled with headings and/or sub-headings | | | |
| Material is accurate | | | |
| Material is up-to-date | | | |
| Content is relevant to the audience | | | |
| Content is relevant to the objectives | | | |
| Content is appropriate for the audience | | | |
| Content is presented at the appropriate depth | | | |
| Technical Qualities | | | |
| Technological requirements are clearly specified | | | |
| Technological requirements are reasonable for the program's purpose | | | |
| Program runs without errors or bugs | | | |
| Load times are reasonable for the type of content being presented | | | |

Software Evaluation Checklist Part 3

Pedagogical qualities are of the greatest weight and importance with regard to this checklist because they seek to determine whether or not the software has instructional value. For the software to be truly instructional, it must define the audience and performance objectives. It also needs to provide a context that resembles the performance context as closely as possible if the users are to be able to apply any resulting knowledge. The software must employ strategies (such as ARCS) that motivate users to take advantage of the software. It also must provide reviews, practice and feedback, and assessments that connect to the learning objectives. All of these are essential components of any learning solution and must have a great impact on evaluating the instructional merit of software. This category is also given the greatest weight because, for this particular checklist, the software is already in use and the goal is to determine the instructional effectiveness of the software.

The user interface category deals with how the user actually interacts with the software itself. This is important because the functionality of the software can directly impact the user's experience and learning. Generally, the software needs to be clean, intuitive, in alignment with technical and cultural standards of the audience, and well-organized. These factors will influence the user's general feeling of the software and can positively or negatively impact the user's perception of the program's quality and merit.

Similarly, aesthetics are important because they determine how the user will see and interpret the program's messages. Even if the content is strong, poor presentation from an aesthetic standpoint could negatively impact the user's perception.

Content is also an important consideration because it is the "meat" of the program. It consists of all the messages that will be communicated. To be meaningful, the content needs to be chunked into manageable bits that are labeled appropriately. It also needs to be accurate, timely, and relevant to the audience. In order for the program to be of instructional value, the content needs to support the learning objectives. The content needs to be appropriate for the audience and the topics need to be covered at sufficient depth to adequately cover the material without overwhelming the user.

Finally, I selected some technical aspects that are also important to consider. While these are not indicators of the instructional merit of the program, they are important considerations because they determine whether the software can be used efficiently. The software needs to clearly define its technological requirements and those requirements must be reasonable for the program's purpose. The software also must run without crashing or other errors. Additionally, it is important to evaluate whether the program elements load within a reasonable amount of time so as not to detract from the instructional experience.