Heuristic evaluation – a system checklist xerox corporation

Usability Techniques Heuristic Evaluation

A System Checklist Xerox Corporation.

Heuristic Evaluation - A System Checklist:

1. Visibility of System Status The system should always keep the user informed about what is going on, through appropriate feedback within a reasonable time.

2. Does every display begin with a title or header that describes screen contents?

3. Is there a consistent icon design scheme and stylistic treatment across the system?

4. Is a single, selected icon clearly visible when surrounded by unselected icons?

5. Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?

6. In multipage data entry screens, is each page labeled to show its relation to others?

7. If overtype and insert mode are both available, is there a visible indication of which one the user is in? |

8. If pop-up windows are used to display error messages, do they allow the user to see the field in error?

9. Is there some form of system feedback for every operator action?

10.   After the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started?

11.   Is there visual feedback in menus or dialog boxes about which choices are selectable?

12.   Is there visual feedback in menus or dialog boxes about which choice the cursor is on now?

13.   If multiple options can be selected in a menu or dialog box, is there visual feedback about which options are already selected?

14.   Is there visual feedback when objects are selected or moved?

15.   Is the current status of an icon clearly indicated?

16.   Is there feedback when function keys are pressed?

17.   If there are observable delays (greater than fifteen seconds) in the system's response time, is the user kept informed of the system's progress?

18.   Are response times appropriate to the task?

19.   Typing, cursor motion, mouse selection: 50-1 50 milliseconds

20.   Simple, frequent tasks: less than 1 second

21.   Common tasks: 2-4 seconds

22.   Complex tasks: 8-12 seconds.

23.   Are response times appropriate to the user's cognitive processing?

24.   Continuity of thinking is required and information must be remembered throughout several responses: less than two seconds.

25.   High levels of concentration aren't necessary and remembering information is not required: two to fifteen seconds.

26.   Is the menu-naming terminology consistent with the user's task domain?

27.   Does the system provide visibility: that is, by looking, can the user tell the state of the system and the alternatives for action?

28.   Do GUI menus make obvious which item has been selected?

29.   Do GUI menus make obvious whether deselection is possible?

30.   If users must navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids?

31.   Match Between System and the Real World The system should speak the user's language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

32.   Are icons concrete and familiar? |

33.   Our menu choices ordered in the most logical way, given the user, the item names, and the task variables?

34.   If there is a natural sequence to menu choices, has it been used?

35.   Do related and interdependent fields appear on the same screen?

36.   If the shape is used as a visual cue, does it match cultural conventions?

37.   Do the selected colors correspond to common expectations about color codes?

38.   When prompts imply a necessary action, are the words in the message consistent with that action?

39.   Do keystroke references in prompts match actual key names?

40.   On data entry screens, are tasks described in terminology familiar to users?

41.   Are field-level prompts provided for data entry screens?

42.   For question and answer interfaces, are questions stated in clear, simple language?

43.   Do menu choices fit logically into categories that have readily understood meanings?

44.   Are menu titles parallel grammatically? |

45.   Does the command language employ user jargon and avoid computer jargon?

46.   Are command names specific rather than general? |

47.   Does the command language allow both full names and abbreviations?

48.   Are input data codes meaningful?

49.   Have uncommon letter sequences been avoided whenever possible?

50.   Does the system automatically enter leading or trailing spaces to align decimal points?

51.   Does the system automatically enter a dollar sign and decimal for monetary entries?

52.   Does the system automatically enter commas in numeric values greater than 9999?

53.   Do GUI menus offer activation: that is, make obvious how to say " now do it"?

54.   Has the system been designed so that keys with similar names do not perform opposite (and potentially dangerous) actions?

55.   Are function keys labeled clearly and distinctively, even if this means breaking consistency rules?

56.   User Control and Freedom Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked " emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

57.   If setting up windows is a low-frequency task, is it particularly easy to remember?

58.   In systems that use overlapping windows, is it easy for users to rearrange windows on the screen?

59.   In systems that use overlapping windows, is it easy for users to switch between windows?

60.   When a user's task is complete, does the system wait for a signal from the user before processing?

61.   Can users type-ahead in a system with many nested menus?

62.   Are users prompted to confirm commands that have drastic, destructive consequences?

63.   Is there an " undo" function at the level of a single action, a data entry, and a complete group of actions?

64.   Can users cancel out of operations in progress?

65.   Are character edits allowed in commands?

66.   Can users reduce data entry time by copying and modifying existing data?

67.   Are character edits allowed in data entry fields?

68.   If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code?

69.   If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut?

70.   Are menus broad (many items on a menu) rather than deep (many menu levels)?

71.   If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus?

72.   If users can go back to a previous menu, can they change their earlier menu choice?

73.   Can users move forward and backward between fields or dialog box options?

74.   If the system has multipage data entry screens, can users move backward and forward among all the pages in the set?

75.   If the system uses a question and answer interface, can users go back to previous questions or skip forward to later questions?

76.   Do function keys that can cause serious consequences to having an undo feature?

77.   Can users easily reverse their actions?

78.   If the system allows users to reverse their actions, is there a retracting mechanism to allow for multiple updos?

79.   Can users set their own system, session, file, and screen defaults?

80.   Consistency and Standards Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

81.   Have industry or company formatting standards been followed consistently in all screens within a system?

82.   Has a heavy use of all uppercase letters on a screen been avoided?

83.   Do abbreviations not include punctuation?

84.   Are integers right-justified and real numbers decimal-aligned?

85.   Are icons labeled?

86.   Are there no more than twelve to twenty icon types?

87.   Are there salient visual cues to identify the active window?

88.   Does each window have a title?

89.   Is vertical and horizontal scrolling possible in each window?

90.   Does the menu structure match the task structure?

91.   Have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system?

92.   Our menu choice lists presented vertically?

93.   If " exit" is a menu choice, does it always appear at the bottom of the list?

94.   Are menu titles either centered or left-justified?

95.   Our menu items left-justified, with the item number or mnemonic preceding the name?

96.   Do embedded field-level prompts appear to the right of the field label?

97.   Do on-line instructions appear in a consistent location across screens?

98.   Are field labels and fields distinguished typographically?

99.   Are field labels consistent from one data entry screen to another?

100.   Are fields and labels left-justified for alpha lists and right-justified for numeric lists?

101.   Do field labels appear to the left of single fields and above list fields?

102.   Are attention-getting techniques used with care?

103.   Intensity: two levels only

104.   Are there no more than four to seven colors, and are they far apart along the visible spectrum?

105.   Is a legend provided if color codes are numerous or not obvious in meaning?

106.   Have pairings of high-chroma, spectrally extreme colors been avoided?

107.   Are saturated blues avoided for text or other small, thin line symbols?

108.   Is the most important information placed at the beginning of the prompt?

109.   Are user actions named consistently across all prompts in the system?

110.   Are system objects named consistently across all prompts in the system

111.   Do field-level prompts to provide more information than a restatement of the field name?

112.   For question and answer interfaces, are the valid inputs for a question listed?

113.   Our menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?

114.   Does the structure of menu choice names match their corresponding menu titles?

115.   Are commands used the same way, and do they mean the same thing, in all parts of the system?

116.   Does the command language have a consistent, natural, and mnemonic syntax?

117.   Do abbreviations follow a simple primary rule and, if necessary, a simple secondary rule for abbreviations that otherwise would be duplicated?

118.   Is the secondary rule used only when necessary? 4. 45| Are abbreviated words all the same length?

119.   Is the structure of a data entry value consistent from screen to screen?

120.   Is the method for moving the cursor to the next or previous field consistent throughout the system?

121.   If the system has multipage data entry screens, do all pages have the same title?

122.   If the system has multipage data entry screens, does each page have a sequential page number?

123.   Does the system follow industry or company standards for function key assignments?

124.   Our high-value, high-chroma colors used to attract attention?

125.   5. Help Users Recognize, Diagnose, and Recover From Errors Error messages should be expressed in plain language (NO CODES).

126.   Is sound used to signal an error?

127.   Are prompts stated constructively, without overt or implied criticism of the user?

128.   Do prompts imply that the user is in control?

129.   Are prompts brief and unambiguous.

130.   Are error messages worded so that the system, not the user, takes the blame?

131.   If humorous error messages are used, are they appropriate and inoffensive to the user population?

132.   Are error messages grammatically correct?

133.   Do error messages avoid the use of exclamation points?

134.   Do error messages avoid the use of violent or hostile words?

135.   Do error messages avoid an anthropomorphic tone? | O O O|

136.   Do all error messages in the system use consistent grammatical style, form, terminology, and abbreviations?

137.   Do messages place users in control of the system?

138.   Does the command language use normal action-object syntax?

139.   Does the command language avoid arbitrary, non-English use of punctuation, except for symbols that users already know?

140.   If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error?

141.   Do error messages inform the user of the error's severity?

142.   Do error messages suggest the cause of the problem?

143.   Do error messages provide appropriate semantic information?

144.   Do error messages provide appropriate syntactic information?

145.   Do error messages indicate what action the user needs to take to correct the error?

146.   If the system supports both novice and expert users, are multiple levels of error-message detail available?

147.   Error Prevention Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

148.   If the database includes groups of data, can users enter more than one group on a single screen?

149.   Have dots or underscores been used to indicate field length?

150.   Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu?

151.   Our menu choices logical, distinctive, and mutually exclusive?

152.   Are data inputs case-blind whenever possible? |

153.   If the system displays multiple windows, is navigation between windows simple and visible?

154.   Are the function keys that can cause the most serious consequences in hard-to-reach positions?

155.   Are the function keys that can cause the most serious consequences located far away from low-consequence and high-use keys?

156.   Has the use of qualifier keys been minimized?

157.   If the system uses qualifier keys, are they used consistently throughout the system?

158.   Does the system prevent users from making errors whenever possible?

159.   Does the system warn users if they are about to make a potentially serious error?

160.   Does the system intelligently interpret variations in user commands?

161.   Do data entry screens and dialog boxes indicate the number of character spaces available in a field?

162.   Do fields in data entry screens and dialog boxes contain default values when appropriate?

163.   Recognition Rather Than Recall Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

164.   For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?

165.   Does the data display start in the upper-left corner of the screen?

166.   Are multiword field labels placed horizontally (not stacked vertically)?

167.   Are all data a user needs on display at each step in a transaction sequence?

168.   Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?

169.   Have prompts been formatted using white space, justification, and visual cues for easy scanning?

170.   Do text areas have " breathing space" around them?

171.   Is there an obvious visual distinction made between " choose one" menu and " choose many" menus?

172.   Have spatial relationships between soft function keys (on-screen cues) and keyboard function keys been preserved?

173.   Does the system gray out or delete labels of currently inactive soft function keys?

174.   Is white space used to create symmetry and lead the eye in the appropriate direction?

175.   Have items been grouped into logical zones, and have headings been used to distinguish between zones?

176.   Are zones no more than twelve to fourteen characters wide and six to seven lines high?

177.   Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas?

178.   Are field labels close to fields, but separated by at least one space?

179.   Are long columnar fields broken up into groups of five, separated by a blank line?

180.   Are optional data entry fields clearly marked?

181.   Are symbols used to break long input strings into " chunks"?

182.   Is reverse video or color highlighting used to get the user's attention?

183.   Is reverse video used to indicate that an item has been selected?

184.   Are size, boldface, underlining, color, shading, or typography used to show relative quantity or importance of different screen items?

185.   Are borders used to identify meaningful groups?

186.   Has the same color been used to group related elements?

187.   Is color coding consistent throughout the system?

188.   Is color used in conjunction with some other redundant cue?

189.   Is there good color and brightness contrast between image and background colors?

190.   Have light, bright, saturated colors been used to emphasize data and have darker, duller, and desaturated colors been used to de-emphasize data?

191.   Is the first word of each menu choice the most important?

192.   Does the system provide mapping: that is, are the relationships between controls and actions appear to the user?

193.   Are input data codes distinctive?

194.   Have frequently confused data pairs been eliminated whenever possible?

195.   Have large strings of numbers or letters been broken into chunks?

196.   Are inactive menu items grayed out or omitted?

197.   Are there menu selection defaults?

198.   If the system has many menu levels or complex menu levels, do users have access to an on-line spatial menu map?

199.   Do GUI menus offer affordance: that is, make obvious where selection is possible?

200.   Are there salient visual cues to identify the active window?

201.   Are function keys arranged in logical groups?

202.   Do data entry screens and dialog boxes indicate when fields are optional?

203.   On data entry screens and dialog boxes are dependent fields displayed only when necessary?

204.   Flexibility and Minimalist Design Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the " average" user (e. . , physical or cognitive ability, culture, language, etc.)

205.   If the system supports both novice and expert users, are multiple levels of error message detail available? Does the system allow novices to use a keyword grammar and experts to use a positional grammar?

206.   Can users define their own synonyms for commands?

207.   Does the system allow novice users to enter the simplest, most common form of each command, and allow expert users to add parameters?

208.   Do expert users have the option of entering multiple commands in a single string?

209.   Does the system provide function keys for high-frequency commands?

210.   For data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen?

Reference:

1. Primary Source Making Computers-People Literate. © Copyright 1993.

2. By Elaine Weiss ISBN: 0-471-01877-5 Secondary Source Usability Inspection Methods. © Copyright 1994.

3. By Jakob Nielsen and Robert Mack ISBN: 1-55542-622-0