Evaluation criteria

Evaluate what?

1

Evaluation criteria

Are the way in which you'll assess whether your solution actually met the brief (the requirements in the SRS).

Used twice:

- To evaluate your two designs, and choose the one which will best meet the brief (part of the Design step)
- To evaluate the project after it's finished (and usually after it's implemented), to ensure it's working as required
  (you'll use the eval criteria again in the final, Evaluation stage)

2

Evaluation criteria

Evaluation criteria should be created to reflect how you will evaluate your designs and the solution on completion, these must be related to your SRS that should state the following:

- Functional requirements
- Non functional requirements
- Efficiency of the solution
- Effectiveness of the solution

3

Evaluation criteria - examples

- Error messages are meaningful (effectiveness)
- Searching data should be fast (efficiency)
- Design is appealing (NF req)
- The database stores data efficiently
  (with few blanks and limited character lengths, no duplicated data, use of relational tables if needed)
- Manipulation of data (calculations, etc) are fast
- Searching for a product category is fast (eg use of CASE vs IF)
- Calculations are accurate (effectiveness, Func reqmt)
- Records are organised so they are easy for Admin to view, search, sort at the conclusion of an event

See page 61 - 68 of your text, for ideas about what to evaluate.

4

Design folio

5

- Describe two possible solutions in a minimal way, then use your evaluation criteria to assess them.

6

Two design solutions = Design Folio

Design Folio requires you to complete some of the following to describe their two (or three) design ideas. You can use some of these, for one or the other, or both.

You can 'describe' the differences in your 2 different solutions designs using any of these:

- pseudocode (for calculations, search or sort)
- data dictionary (of variable or field names)
- object descriptions
- application architecture, describe data storage, technical requirements, security etc
- Sitemap (for website) / storyboard (for animation/game)
- Interface/screen mockups
- Describe your solution, a paragraph of so, in terms of:
  - Application architecture
  - Data storage (csv file VS database)
  - other systems that need to link in
  - the devices it'll be suitable for, etc.
- No IPO required

7

So what could your design folio look like?

Some possible ways to depict your 2 solutions,
and the differences between them,
are on the next slides.

8

So what could your design folio look like?

Design 1:

- Data dictionary for a bookings table, storing date as separate day, month, year fields
- Use of icons for the menu, in interface design
- Pseudocode for calculation / sort using CASE
- Workflow – users needs to go through Step1, Step 2 and Step 3.

Design 2:

- Data dictionary for a bookings table, storing date in a single date fields
- Use of words for the menu, in interface design
- Pseudocode for calculation / sort using nested IF statement
- Workflow – user can jump from Step 1 to Step 3 or cancel the process.

9

So what could your design folio look like?

Design 1:

- Interface shows a prompt, asking the user fro confirmation "Are you sure these details are correct?" before saving/storing the data.
- Interface shows menu across the top, a submenu in a horizontal line below it
- Use of Function, to identify the user type (customer or administrator) and display the relevant menu

Design 2:

- Sitemap shows the data being saved/stored without asking the user for confirmation.
- Interface shows menu down the left-hand side. Submenus slide out over the page.
- Use of two different pages, to display the relevant page to the user type (customer or administrator)

10

So what could your design folio look like?

Design 1:

- Reason for choice of data store - CSV
- What devices it'll work on, and how the data store & device choice impacts the application architecture
- Pseudocode for a search algorithm
- Layout diagram for the search page
- Sitemap

Design 2:

- Reason for choice of data store - database
- What devices it'll work on, and how the data store & device choice impacts the application architecture
- Colour scheme
- Layout diagram for the search page
- Sitemap

11

So what could your design folio look like?

Try to choose tools that represent
the INTERFACE, PROCESSING and DATA STORE layers of your solution.
This means using 3 tools for each Design Idea.

12

Ways to differentiate between your two designs: (p66 of text)

Interface & workflow:

Responsiveness – feedback to user when they click to do a task (e.g. edit, delete):
- "Do you really want to do this?" OR
- just do the task, OR
- no response at all (asking questions may be inefficient and annoying for experienced user)

Familiar e.g.–
- like other sites, with a Logo (top left) menu across top … OR
- unique because more efficient and tailored to workflow … OR
- minimal or no menu, steps you through common tasks (everything is a wizard)

Forgiving –
- Undo ? Redo ? OR
- auto backup? OR
- Regular saves?

13

Ways to differentiate between your two designs: (p66 of text)

Programming
- Objects (fully OOP)
- vs
- Functions(conventional main routine + functions)
- vs
- Monolithic Code (ugly patched together, spaghetti code with no individual functions)

Or refer to more specific pseudocode, use of CASE instead of nested IFs.

The above can speak to a maintainability (NFR) evaluation

14

Things to remember

- That you need to evaluate your 2 designs using the written evaluation criteria
- That you'll use them again to evaluate your finished solution.

Consider this when choosing how to represent the differences between the 2 solutions you've designed

TIP:
Try to choose tools that represent the INTERFACE, PROCESSING and DATA STORE layers of your solution. This means using at least 3 tools for each Design Idea.

15