CSSE 371
Software Requirements Engineering
Exam 2, Fall 2012

Name: _____Key_____

Mailbox:_____

1. Closed book, with one hand-written, double-sided sheet of notes.
2. Your answers must be concise and address the question directly.
3. All questions in Part A should be answered in 3 to 4 sentences.
4. Please provide contextual information you think is necessary to understand your answers in Part B

Part A: / 60 Points

Question 1		**12 Points**
Question 2		**12 Points**
Question 3		**12 Points**
Question 4		**12 Points**
Question 5		**12 Points**

Part B: / 40 Points

Question 1		**15 Points**
Question 2		**15 Points**
Question 3		**10 Points**

Total: / 100 Points

Part A

1. Suppose that you are a software developer, and you are given the following requirement by a requirements engineer::

   The system displays the Basic Loss Information Page, with a form to enter basic loss information specific to the claim line.

Identify three problems of clarity in this requirement, which you would ask the requirements engineer to fix, and explain why. If you need to make any assumptions in your discussion, please state those assumptions:

Answer: The "clarified" version of this requirement, from the slides (371-Chapter23-24.ppt, slide 13), is: The clerk enters basic loss information specific to the claim line." This version overcomes these problems:

a. The original statement doesn't say you actually have to enter the data, but probably you do.

b. The original statement suggests a design which may or may not need to be followed.

c. The original statement combines system actions and user actions.

2. I have never been on a project which didn't need to reduce its scope for the first release. At the start of a project, the developers are all optimistic and think they can do anything! Given this human nature problem, what people and processes would you put in place to ensure that the customer does not end up disappointed?

Answer: These ingredients are known to help the problem of customer disappointment:

a. Have the customer prioritize requirements with the idea of matching actual release content to project difficulty, working down the list, when the latter becomes more known.
b. Use iterative development so that the customer gets early versions as the developers continue to work.
c. Put people in place whose role is to manage customer expectations.
d. Maintain a close relationship with the customer, so that they better understand technical issues encountered by the developers.
e. Have product and release management people in place to ensure the developers continue to match changing expectations of customers.

3. Leffingwell's "Supplementary Specification Checklist" asks, "Have supplementary requirements been linked to use cases where appropriate?" Ok, where would this linking be

appropriate, and where would it not be appropriate? Give an example of *each* of these two kinds of requirements – those that should be linked, and those that should not:

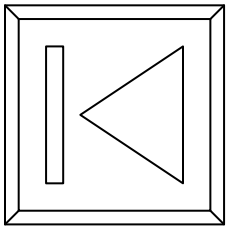
Answer: This point appears on slide 20 of "371-Chapter27-29.ppt". The two kinds of requirements typically linked to use cases are these:

a. *Functional requirements associated with multiple use cases, like screen design.*

b. *Quality attributes that apply to particular use cases, like a quality attribute for performance or for reliability, which applies just to the main activities a system does.*

Pretty much all other requirements from the supplementary spec are unrelated to specific use cases.

4. One of the ID book's examples of "virtual affordances" is this:



a. What does this particular symbol "afford" for the user?

Answer: It looks like the action to return to the beginning of some function, medium, or file, because the arrow points to some fixed point on the left. (Or, perhaps return to some other fixed point.)

b. If you used it in your system, why would most users know what it means?

Answer: They would have seen something similar before, to control another device.

5. In the experimental design of a usability test comparing two designs, you can use different participants, the same participants, or matched participants. Identify what each of these 3 choices is. If you only had 10 people for your testing, which one would you use, and why?

Answer: From "Chapter_14_ID2e_slides.ppt", Slide 14:

- *Different participants - single group of participants is allocated randomly to the experimental conditions.*
- *Same participants - all participants appear in both conditions.*
- *Matched participants - participants are matched in pairs, e.g., based on expertise, gender, etc.*

With only 10 participants, the safest strategy would be to have all participants do both designs. They could alternate as to which one was done first.

Part B

1. (15 points) Given the following use case:

User Changes Account Preferences

ID#: UC2

Description

User uses the account editing page to change details of their account such as password and age.

Pre-Conditions

1) User is logged in
2) User is on home page

Basic Flow

1) User presses account preferences button
2) User changes the values of the fields that they would like to change [See Alt Flows 'A' & 'B']
3) User presses save changes button
4) System sends change requests to database
5) User gets confirmation of preference changes
6) User is sent to home page

Alternate Flow 'A': User Cancels Preferences Changes

1) User presses cancel during step 2 of the Basic Flow
2) User returned to home screen

Alternate Flow 'B': Data Entered is Invalid

1) User enters invalid information
2) User presented with error message with specific information about what information was invalid
3) User redirected to step 2 of Basic Flow

Post Conditions

1) Any preferences that the user wishes to change have been changed
2) The user is on the home page

To Do: Write a test case for *each* of the "scenarios" (flows) through this use case:

Answer: The question asks for one each, so there would be one for the basic flow, one for flow A and one for flow B. For example:

Test Case ID	Scenario	Description	Expected Result	Actual Result

1	Basic flow	User completes steps 1 – 3 correctly	4) System sends change requests to database 5) User gets confirmation of preference changes 6) User is sent to home page	
2	Flow A	1) User presses cancel during step 2 of the Basic Flow	2) User returned to home screen	
3	Flow B	1) User enters invalid information	2) User presented with error message with specific information about what information was invalid 3) User redirected to step 2 of Basic Flow	

2. Ch 14 of the ID book recommends collecting these types of quantitative data during usability testing. For each type of data, tell whether you collected this data in your team's testing. Whether you did or not, indicate why collecting it would be useful:

Answer: The project-related answers will vary depending on the team project. Why each would be useful:

a. Time to complete a task.

This measures difficulty, efficiency, and possibly learning time.

b. Number and type of errors per task.

This measures design errors, difficulty and difficulty of learning.

c. Number of errors per unit of time.

This is the rate of doing errors, allowing analysis of causes for slow completion time.

d. Number of navigations to online help or manuals.

This shows both use of help and also the need for going to help.

e. Number of users making a particular error.

This pinpoints errors.

f. Number of users completing task successfully.

This shows design success, especially versus errors or total task attempts.

3. Direct Manipulation in action – On the next page, we see Adobe Photoshop in action, allowing the user to change "response curves" to various levels of light and dark in the image shown. With this feature, a user can make any level of light or dark in an image become any different level they choose, by clicking on the curve shown at right, and dragging it somewhere else, as they watch the effect on the image. Each part of the change the user makes leaves a dot on the graph, as shown, and these can be undone at will, too.

   Which of the 3 "core principles of DM" described in the ID book are illustrated by this Photoshop action? Explain why on the next page, referencing the figure itself in your descriptions.

Answer: From "Chapter_2_ID2e_slides_New.ppt", slide 17, these core principles are:

1. *Continuous representation of objects and actions of interest*
2. *Physical actions and button pressing instead of issuing commands with complex syntax*
3. *Rapid reversible actions with immediate feedback on object of interest*

All 3 of these are used –

(1) The image being manipulated is continuously presented and altered.

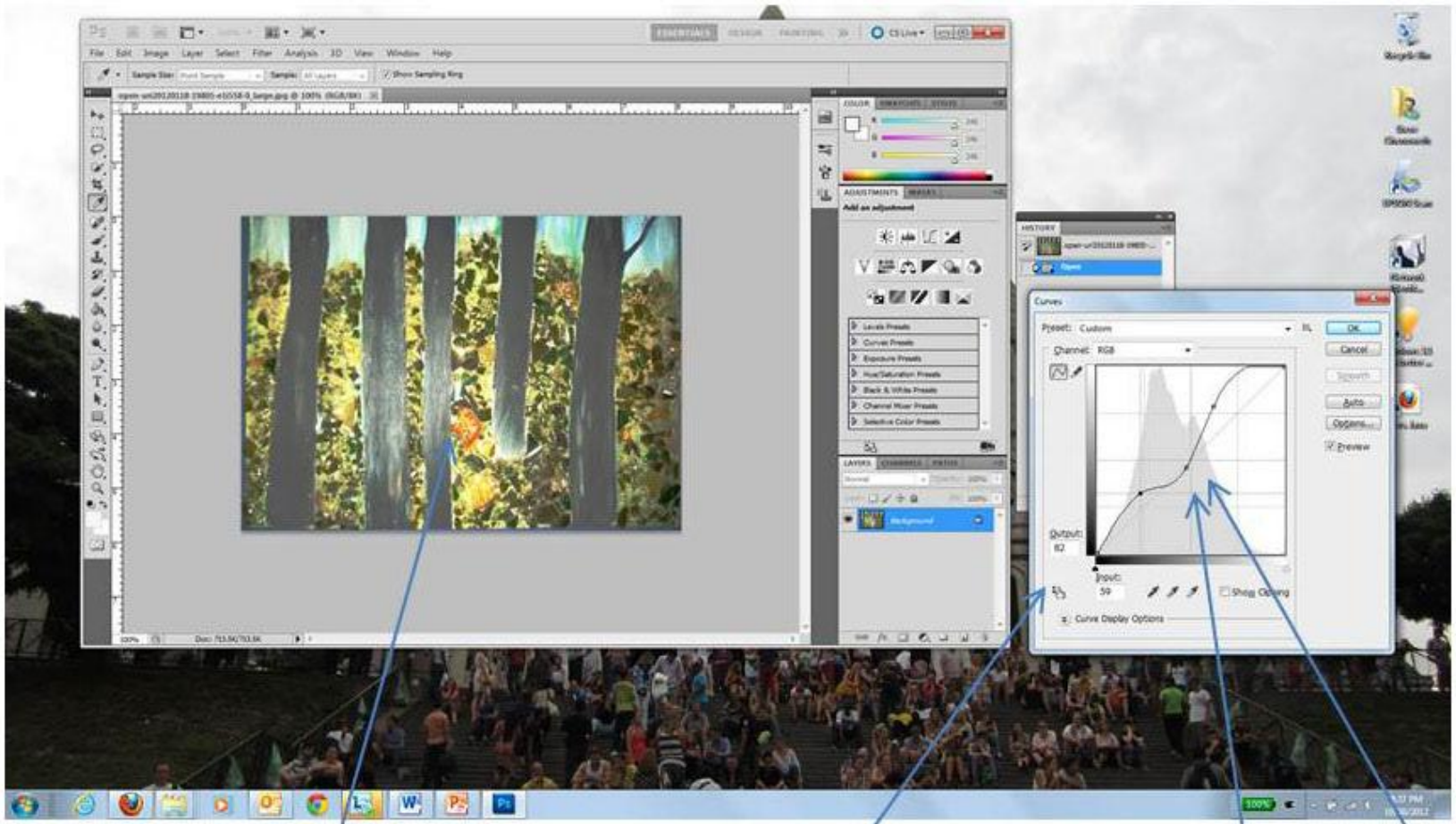
(2) The manipulation is done by dragging on the curve.

(3) The undo function is stated as possible. This is done by dragging on the curve to reposition it, among other ways.

I also considered other advantages of DM that were listed on the following slide, as possible "core principles," namely:

- *Novices can learn the basic functionality quickly*
- *Experienced users can work extremely rapidly to carry out a wide range of tasks, even defining new functions*
- *Intermittent users can retain operational concepts over time*
- *Error messages rarely needed*
- *Immediate feedback*
- *Users gain confidence and mastery and feel in control*

The figure below on the page shows the image with the answers marked on it:



1. Continuous representation of objects of interest

1. Continuous representation of actions of interest

2. Dragging the curve is a physical action vs typing a command.

3. Same action can be undone by Cntrl-Z or dragging it back.