CSSE 371
Software Requirements Engineering
Exam 1, 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		**6 Points**
Question 2		**8 Points**
Question 3		**6 Points**
Question 4		**8 Points**
Question 5		**8 Points**
Question 6		**8 Points**
Question 7		**8 Points**
Question 8		**8 Points**

Part B: / 40 Points

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

Total: / 100 Points

Part A

1. How does defect leakage affect the software development life cycle? In particular, how are requirements a part of this leakage problem? (6 Points)

   Leakage is the way requirement errors can require scrapping and redoing of work that follows. The requirements errors "leak" into all that later work, progressively causing more rework to be required.

2. Why is it important to keep a problem statement up-to-date throughout the development life cycle? And, why do many development groups fail to do this? (8 Points)

   The problem statement describes the problem to be solved, written in a standardized format. It describes what the results of the project will be, and the benefits of those results. If it is not up to date, then some party (either the customer or the developers) are likely to think that the original plan is still in effect, and be off in a very basic way.

   Development groups fail to keep this up to date because their main focus is on the current activity, such as coding and testing, and not on fixing other things which they already did.

3. What elicitation and communication techniques can you use to avoid the "Yes, but" syndrome? (6 Points)

   Storyboarding is especially useful at dealing with "Yes, but" situations, where the customer agrees (perhaps because this is what they asked for) but really wants more or different things Storyboards are powerful visuals, and so they bring out these reactions early, when you can still do something about them (like include the customer's real desires into the project).

4. Suppose you wanted to use data flow diagrams, personas and storyboards in capturing project requirements. In what order would you develop them? Justify your answer (8 Points)

   Probably personas, storyboards, and DFD's, in that order. The reason is because this is the order from most fuzzy to most precise.

5. How does question design influence data analysis? What kind of questions would help ensure the stakeholders don't change their answers? Explain using an example (8 Points)

   The way you design a question can affect how well the data is gathered and analyzed. For instance, if you ask the question what do you think of e-bay .com and keep the question as a open-ended one, it will result in a lot of different answers and it will be very difficult to analyze the various answers you get. You need to distinguish between the various different answers you get. However, if the question was Do you like e-bay, you might have to deal with just yes, no or neither.

   Asking very specific questions, and asking follow-up questions to pin down the person being asked, is the usual way to ensure they don't change their mind. You are asking them to be as clear as possible about their answer, and as definite as possible.

6. `Wizard -of-Oz' is a commonly used technique in prototyping. What is it, and explain what can go wrong with this approach? (8 Points)

   In this approach, the user thinks they are interacting with a computer, but a developer is responding to input rather than the system.

   It is difficult to achieve for large systems and the users might mistake the delay in response to a problem with the system and might end up focusing on performance rather than on the interaction model.

7. How do throw-away prototypes end up being used as a real product? Describe a believable situation, and how you would try to prevent this? (8 points)

   The groups intend for the prototypes to be throw-away; however, they are so convincing that the customer presses them for an earlier-than-planned delivery. Or, the fault can lie with the developers, because they have to make it look like the prototype is the real thing, in order to get continued funding.

8. The theories in the social sciences are inherently weaker than they are in the physical sciences. Use one of the qualitative analysis techniques we discussed to explain this fact-of-life, in terms of how that technique develops theories about ID situations: (8 points)

   Key parts of all three theories are constructed specifically for the situation at hand, using data taken from experiments on or observations of that situation. Thus, there is no guarantee that the resulting concepts are generalizable.

Part B

1. (15 points) The following narrative describes how to use the *ls* command from a Linux terminal:

Both Supervisors and Users use this command. The command *ls* lists the contents of the current working directory, so you need to be pointing at the right directory first. Use *cd* to switch that first, if necessary.

Type *ls* at the prompt and press enter; you will see the contents of your home directory. You should see something like the following entries.

$ ls

Desktop/ Document/ HelloWorld.c

And, of course, *ls* comes with a number of flags that let you customize the ways it lists the directory contents. One of these is *-l*. The *-l* flag stands for long listing; that is, it lists the directory entries with more details. The details include the type (file or directory), owner, size, permissions, modification date, and more. The *-a* flag stands for *-all* – it shows directory entries starting with a ".". This is different from the *-author* flag, which prints the author of each file. Any of these options can be used in combination.

Write a use case that uses the above information, describing the required steps in the process of using *ls* with Linux:

Use case name: Use 'ls' command
Brief description: How this command works, and basic options for it.
Actors: Supervisors and users
Precondition: The actor is pointing at the right directory.
Main flow:

1. *User types 'ls' at the command prompt.*
2. *System responds with a display of the names of the files in the directory.*

Postconditions: The directory contents are displayed.
Alternate flows:
1a. User types 'ls –l'.
2a. System displays a 'long listing' that shows more details, including file type, owner, size, permissions, modification date, and more.
1b. User types 'ls –a' or 'ls –all'.
2b. System responds with directory entries starting with a '.' (in addition to other entries).
1c. User types 'ls –author'.
2c. System responds by adding the author of each file to the file data shown.

2. (15 points) Suppose you need to rewrite the script for "How to install Ubuntu Linux," for Rose freshmen who will be taking the course CSSE132.

The existing script has 103 steps, some of which are themselves complex, like

Step 73: Now do the emacs tutorial.

Describe the process you would use in the requirements for your rewrite of this installation script for the class. Include the following in your description:

a. Who the stakeholders are.

b. How you would gather requirements from them.

c. How you would manage those requirements.

d. What artifacts would you need to create?

e. Would you need to create a prototype? Why or why not?

A main goal of the rewrite would be to reduce the complexity of the process. This could include strategies like providing a script (if possible) or providing answers to cut and paste (if possible) or showing the exact entries to make at each point (the likely solution). It further might help provide context about Linux, to explain a little about what is going on, why the entries need to be made as they are, or what the machine is doing while you are going through this process.

However, the question here asks how you would do the requirements for such a solution, to be sure one or another solution idea was really workable!

a. Stakeholders: The students, the professor teaching the class and his/her TA's.

b. Gathering requirements: Some students could try the old process as we observed. Students who already took the course could provide their reactions to the old process (and suggestions for improvement). And the professor or the TA's might have very specific needs or interests that we could get by interviewing and/or written input. (E.g., the professor could mark desired changes in the old script.) Another stakeholder would be system administrators or others who are Ubuntu experts, and who might provide valuable tips or checks on this other input. (E.g., is there an image that students can simply download, instead of going through an installation process?)

c. Managing requirements: There could be changes that occur, such as a new version of Ubuntu coming out, or someone thinks of other things that should be in the script. So, there should be a "baseline" to the requirements, after which changes are made very carefully.

d. Artifacts: We would need examples of existing materials, notes from interviews or surveys, and any prototypes done.

e. Prototype? Probably a very good idea, so that "better or worse" studies could be done.

3. (10 points) You are starting to design a new software system which will be long-lived, and keeping its maintenance cost low is a key issue. Using the format below, do a Quality Attribute scenario for the following modifiability requirement:

   For each new software release, customer administrators will be able to make all changes to their site configurations, without assistance from the development team, in an average of under 1 hour.

   Source of stimulus: Customer administrator

   Stimulus: Changes to the configuration

   Environment: New release being installed

   Artifact: On-site system being brought-up

   Response: The administrator makes all changes to their site configuration.

   Response measure: Completed in an average of under 1 hour.