



University of Waterloo Final Examination

Fall

2008

Student Name	_____
Student ID Number	_____
Student UWDir Name	_____
Circle your section	8:30am:CS445 8:30am:CS645 8:30am:ECE451 1:30pm:SE463

Course Abbreviations & Numbers	CS445/CS645/ECE451/SE463
Course Title	Software Requirements Specification and Analysis
Instructor	Daniel M. Berry

Date of Exam	12 December 2008
Time Period	12:30pm–3:00pm
Duration of Exam	2.5 hours
Number of Exam Pages	25+1
Exam Type	Closed book

Marking Scheme:

Question	1	2	3	4	5	6	7	8	9	10	Total
Max	30	9	9	12	21	21	15	15	9	9	150
Score											
Grader											

Allowing 15 minutes to understand the problem on Pages 7 through 8 there are 135 minutes for 150 marks, roughly .9 minutes, or about 54 seconds, per mark; plan your time wisely!

It is suggested that you read through the entire exam first before answering any question, so that you know where to spend your time.

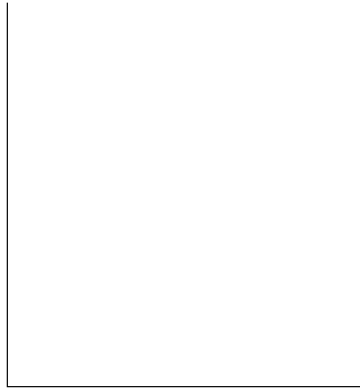
There should be sufficient space to write or draw your answers in the exam itself. If you need more paper, please request it from your proctor.

For your convenience, the last page has a duplicate of a diagram that you will need to answer several questions. You may tear this page out so that you can refer to its diagram without flipping pages.

Part I: General RE (30 marks \approx 27 minutes)

1. Some Easy Issues (30 marks \approx 27 minutes)

- a. The implication of the unit of work “person-month” that makes the term itself a myth is that the unit implies the following graph:



and that _____ and _____ can be traded off. The main reason that they

cannot be traded off is that _____ grows _____ with the

number of people in a project, while _____ grows _____ with the number of people in the project.

- b. Describe one kind of non-computer-related job for which the person-month is not a mythical unit of work.
- c. For extra credit that makes up points lost elsewhere in exam, describe one kind of software-engineering-related job for which the person-month is not a mythical unit of work.

d. According to Brooks, adding _____ to a late project makes it even

_____. **Note that the answer in the first underline in Question d is the *same* as the answer in the first underline in Question e.**

e. Burkinshaw, however observed that adding _____ to a project in progress has

the potential advantage of adding to the project _____, which is a rare commodity among people working on the project.

f. A consequence of the person-month being a myth is that in COCOMO, the formula for total months of development time, D , as a function of total person-months of effort, E is

$$D = c \times E^d$$

where d is (choose one by checking its blank line):

$\approx .33$ _____

1.00 _____

≈ 1.33 _____

g. The claim borne out by several case studies is that the _____ one spends work-

ing out requirements the _____ the subsequent implementation phase costs.

Probably the biggest single factor contributing to this effect is that an error caught during

implementation costs about _____ times to fix what the same error costs to fix if it is caught during the requirements phase. Also, an error caught during deployment or

maintenance of a system costs about _____ times to fix what the same error costs to fix if it is caught during the requirements phase.

h. Krogstie, quoting Kano, identifies three kinds of requirements:

1. _____

2. _____

3. _____

Describe for each kind of requirements how meeting it contributes to the user's satisfaction with the resulting system.

1.

2.

3.

Describe for each kind of requirements how failing meeting it contributes to the user's dissatisfaction with the resulting system.

1.

2.

3.

- i. Fill in the table to explain the distinction between a software (SW) system product and a program. In the “costs” column, express each unfilled-in cost as a function of the filled-in cost X of the program.

	written for whom	communicates with	costs
program			X
SW system			
product			
SW system product			

- j. Give three distinct reasons why we cannot estimate software development costs very well before we begin the development.

1.

2.

3.

k. The formula $D, S \vdash R$ has been called “the _____ argument” or “_____”, one of the “V”s in “V&V”. (**The last underline must be filled with a full word**) In the formula, D is _____, S is _____, and R is _____.

We view the hardware and software as building a _____ that operates in the _____ through an _____ that is shared between the _____ and the _____.

S must be written using *only* concepts that appear in the _____, while R may be written using concepts that appear also in the _____.

The parts of the world that are outside the _____ and the _____ are considered to be irrelevant to the hardware and software being built. However, sometimes things in this part of the world surprise us by turning out to be _____ to the hardware and software that was built.

1. Inspection of a software development artifact is an effort to find as many potential _____ as possible in the artifact. The determination of whether a potential _____ is an _____, and the correction of any _____ is left to be done by the _____ of the artifact _____ the inspection is finished.

Thus, inspection of an artifact is essentially a brainstorm for _____ in the artifact, followed by having the _____ of the _____ prune the list of _____ to a smaller list of _____ that the _____ of the _____ must _____.

- m. The main difficulties with ranking a list of n requirements into a list totally ordered by priority are that:

1. _____ pairs of _____ have to be compared by _____.
2. It can be difficult to decide which of any two requirements has _____ over the other.

Triaging divides requirements into _____ groups, which are:

_____ (Use as many lines as you need to be consistent

_____ with the number you have given above!)

_____ (It is possible that some lines are left blank!)

_____.

Triaging is derived from a procedure used in hospital emergency rooms to deal with

many simultaneous medical _____ caused perhaps by a public _____.

In this triaging, patients are divided into _____ groups, those patients that will

_____ (Use as many lines as you need to be consistent

_____ with the number you have given above!)

_____ (It is possible that some lines are left blank!)

_____.

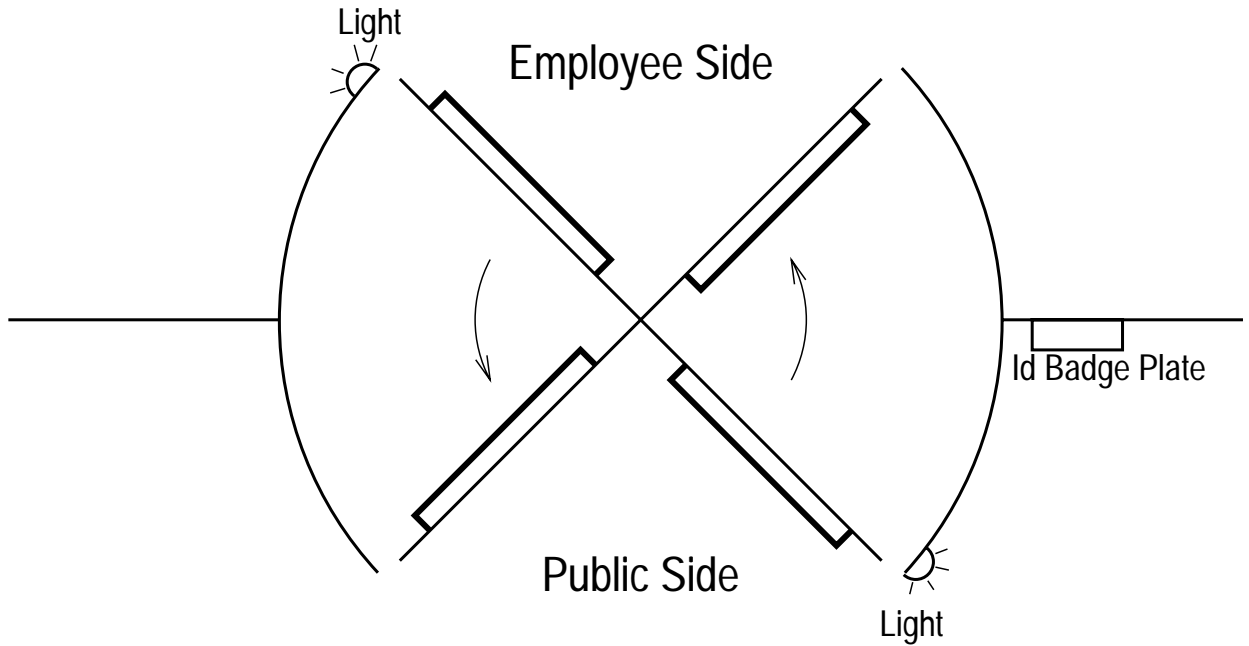
n. According to David Platt, the main reason that software sucks is that the developers, who

are programmers, develop software on the assumption that the typical _____

thinks like _____ think. Therefore, David Platt's first, last, and only law of user interface design, addressed to developers is:

“Know Thy _____, for He Is Not _____.”

Part II: Revolving Door System (72 marks \approx 64.8 minutes)



2. Initially, the revolving door is locked so that no one can go through it in either direction.

3. To go through the revolving door from the employee side to the public side, it suffices for a user, called an **EmployeeSideUser**, to simply push the handle bar on the door on her side. 4. After a very slight delay, the two lights are turned on and the door is unlocked. 5. If a user on either side pushes his or her handle bar within ten seconds, the lights are turned off and the door remains unlocked until the door finishes rotating 180° and whoever has pushed on either side has walked through. 6. Then, immediately the door is locked.

7. To go through the revolving door from the public side to the employee side, a user, called an **PublicSideUser**, must have a valid company id badge. 8. He must place either side of the id badge up against the badge plate on his side of the wall to the right of the door. 9. If the badge is determined to be valid, the two lights are turned on and the door is unlocked. 10. If a user on either side pushes his or her handle bar within ten seconds, the lights are turned off and the door remains unlocked until the door finishes rotating 180° and whoever has pushed on either side has walked through. 11. Then, immediately the door is locked.

12. In either case, after the door is unlocked, if no user on either side pushes a handle bar within 10 seconds, the lights are turned off and the door is locked.

13. There is no time limit on how long it takes for a user to walk through the door.

14. The space inside the revolving door is too small for more than one normal sized adult to walk through the revolving door in the same direction at the same time. **15.** However, it *is* possible for two people, on the opposite sides to walk through the revolving door at the same time.

2. Dividing the World (9 marks \approx 8.1 minutes)

Classify each sentence in the description of the **RevolvingDoor** system according to which part of the formula $D, S \mid\!\!\mid R$ it describes. That is in the underline next to each sentence number, write whichever of D , S , and R the numbered sentence is about. It is possible that a sentence is about more than one of D , S , and R .

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

10. _____

11. _____

12. _____

13. _____

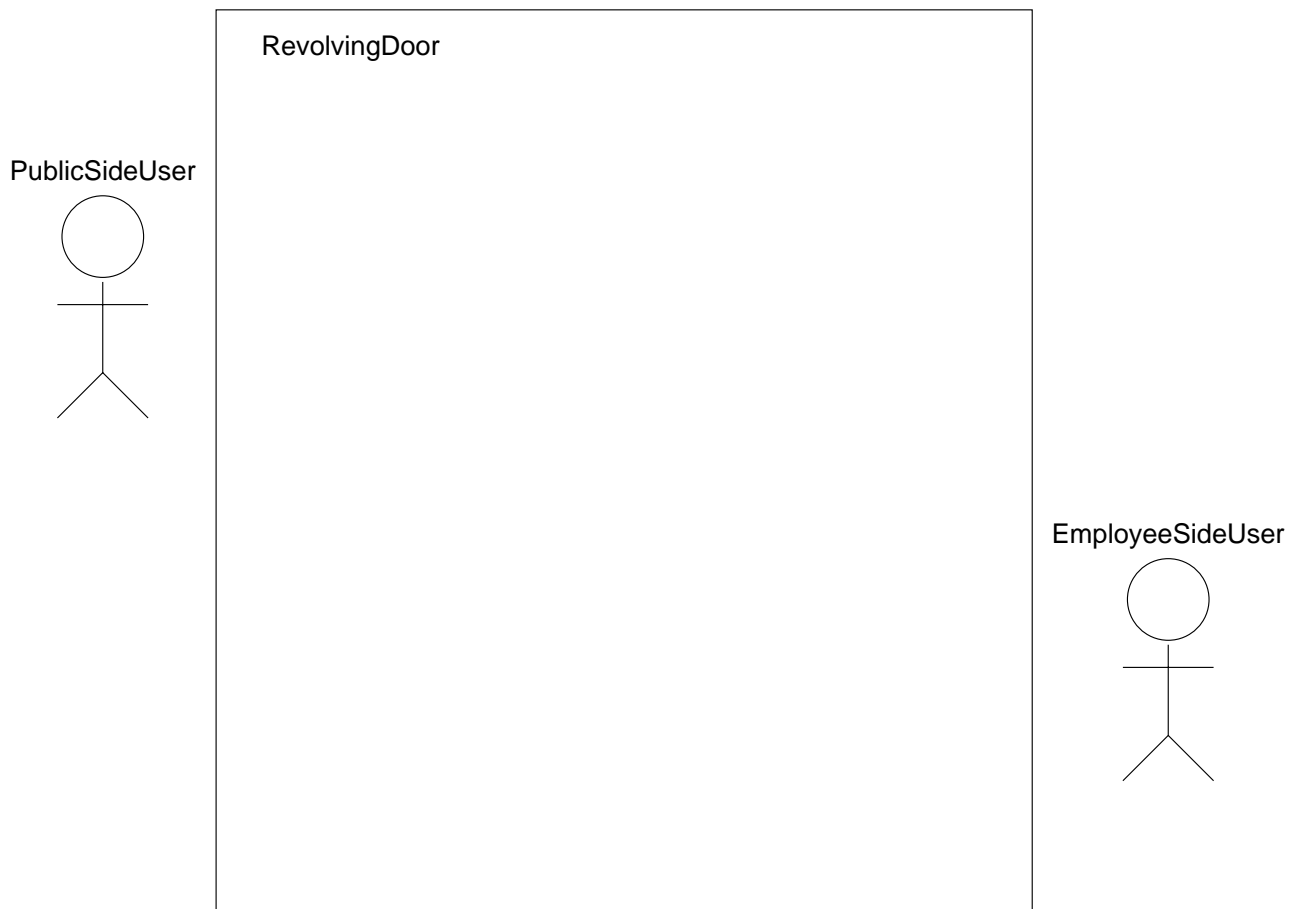
14. _____

15. _____

3. UML Use-Case Diagram (9 marks \approx 8.1 minutes)

Identify the use cases of the **RevolvingDoor** system. These should be the actions any public-side user and any employee-side user can do to the interface objects. To each use case, give a descriptive name consisting of the words of an imperative (command) sentence.

Complete the partial UML use-case diagram shown below for the **RevolvingDoor** system. It shows the two kinds of actors of the system, **PublicSideUsers** and **EmployeeSideUsers**, and the system boundary. After you have put each use case as a bubble inside the system boundary, draw a line connecting each use case to each actor that can do the use case.



4. UML Class Diagram (12 marks \approx 10.8 minutes)

Draw below a UML class diagram for the **RevolvingDoor** system. In addition, show the two kinds of users, **PublicSideUsers** and **EmployeeSideUsers**, introduced in the use case diagram. Show all multiplicities on all classes and on all associations. Please allow for an arbitrary number of users of either kind. With some additional lines, show which parts of the model are in each of the environment, the system, and the interface.

5. Nonfunctional and Other Functional Requirements (21 marks \approx 18.9 minutes)

- a. A logical requirement for the RevolvingDoor system is that it keep all on the public side who do not have a valid company id badge from entering through the revolving door.

Does the RevolvingDoor system always meet this requirement?

If so, explain why

If not:

1. please describe a scenario permitted by the system that violates the requirement by allowing a person on the public side through even though he may not have a valid id badge?

2. what behavior can the HAL company insist of its employees to make sure that no person on the public side without a valid id badge can come through the door?

- -
 3. Which part of $D, S \models R$, i.e., D , S , or R , is modified by the behavior change described in item 2 just above?
 4. Describe changes that can be made to S that make the behavior change described in item 2 above not needed to ensure that the requirement is met.
- b. Which part of $D, S \models R$, i.e., D , S , or R , contains the idea that all and only HAL employees receive valid HAL id badges.
- c. What should the RevolvingDoor system do about the fact that the statement “All and only HAL employees receive valid HAL id badges.” cannot possibly be true?

d. Describe a way that “All HAL employees receive valid HAL id badges.” can be not true.

e. Describe a way that “only HAL employees receive valid HAL id badges.” can be not true.

6. UML Scenario Specification (21 marks \approx 18.9 minutes)

Draw below a UML sequence diagram for a scenario involving one **PublicSideUser**, one **EmployeeSideUser**, the **Door**, the **Lights**, the **IdCardChecker** and the **DoorLightController**. In this scenario, the **EmployeeSideUser** pushes on the door at precisely the same time that the **PublicSideUser** lays his valid id badge up against the id badge plate. However, it is the **EmployeeSideUser** that pushes the door after it is unlocked and before the 10 seconds are up. Both users walk through the revolving door at the same time, finishing at the same time. Carry the scenario through to the door getting locked after both users have walked through.

Part 3: Ambiguity (15 marks \approx 13.5 minutes)

7. Only, Also, and Plural (15 marks \approx 13.5 minutes)

With a drawn line, match each text on the left side with exactly one sentence on the right side, namely one that fits in place of the X in the text. Since the number of items on the two sides are not equal, the correspondence is not 1–1!

X The rest of the people work after lunch.	Only I sleep after lunch.
X I do not work after lunch.	I only sleep after lunch.
Jeff sleeps after lunch. X	I sleep only after lunch.
X The rest of the people sleep after dinner.	I sleep after only lunch.
I sleep before lunch. X	
X I do not sleep before lunch.	
I sleep after dinner. X	Also I sleep after lunch.
I brush my teeth after lunch. X	I also sleep after lunch.
X I do not sleep after dinner.	I sleep also after lunch.
X The rest of the people sleep before lunch.	I sleep after also lunch.

Assuming that there are 100 male members, for each sentence below, indicate the all possible numbers of wives that came, according to the strict meaning of the sentence and *not* assuming that each man has only one wife by law.

1. All members brought their wives.

2. All members brought their wife.

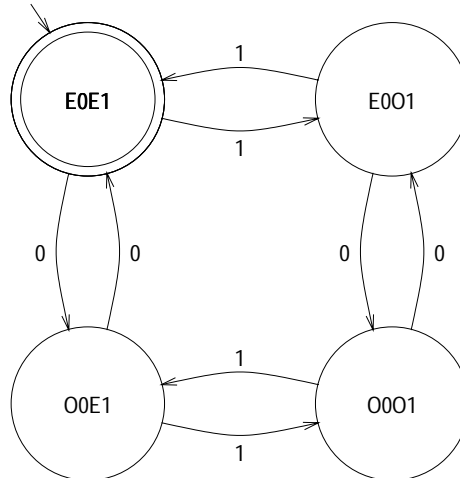
Rewrite All members brought their wives. in singular form (Hint: use Each member as the subject of the sentence.) to make it clear that each member brought exactly one wife. You will get the most credit if you use exactly the same number of words as in the original sentence.

Rewrite All members brought their wives. in singular form (Hint: use Each member as the subject of the sentence.) to make it clear that each member may have brought more than one wife. You will get the most credit if you use exactly the same number of words as in the original sentence.

Part 4: Specification Notations (33 marks \approx 29.7 minutes)

8. Finite State Machines (15 marks \approx 13.5 minutes)

Consider the finite state machine, *FSM* shown below:



Recalling that the state with the little arrow pointing to it from nowhere is the initial state and any state with a double circle outline is an accepting state, check off the blank line next to an input below if and only if the input is accepted by *FSM*.

- _____ (empty string)
- _____ 0
- _____ 1
- _____ 10
- _____ 11
- _____ 100
- _____ 101
- _____ 110
- _____ 111
- _____ 1000
- _____ 1010
- _____ 10111100101100
- _____ 011101000101010

Describe in words the language that is accepted by *FSM*.

FSM accepts each string s consisting of only 0s and 1s such that s ...

9. Linear Temporal Logic (9 marks \approx 8.1 minutes)

Below are a bunch of formulae in linear temporal logic. Among them are formulae that describe part of the behavior of the finite state machine *FSM* introduced in the previous question. Specifically, among them is a description of the circumstances in which *FSM* stays in state **E0E1** and descriptions of the circumstances in which *FSM* moves out of state **E0E1**. There are some other irrelevant formulae among the formulae. Put a check mark in the line preceding each of the *three* relevant formulae.

_____ $\square (E0E1 \rightarrow (E0E1 \ U (0 \vee 1)))$

_____ $\square (E0E1 \rightarrow (E0E1 \ W (0 \vee 1)))$

_____ $\diamond (E0E1 \rightarrow (E0E1 \ W (0 \vee 1)))$

_____ $((E0E1 \wedge 1) \rightarrow \bigcirc E0O1)$

_____ $\square ((E0E1 \wedge 1) \rightarrow E0O1)$

_____ $\square ((E0E1 \wedge 1) \rightarrow \bigcirc E0O1)$

_____ $\square ((E0E1 \vee 0) \rightarrow \bigcirc O0E1)$

_____ $\square ((E0E1 \wedge 0) \rightarrow \bigcirc O0E1)$

_____ $\square ((E0E1 \wedge 0) \rightarrow \bigcirc \bigcirc O0E1)$

_____ $\square ((E0E1 \wedge 0) \rightarrow \bigcirc O0O1)$

10. Decision Tables (9 marks ≈ 8.1 Minutes)

Consider the decision table below that describes how a clerk is to behave when a customer buys a product using any of several methods of payments, according to the cost of the product, the familiarity of the customer to the clerk, and the method of payment.

the item costs less than \$100.00	Y	Y	N	N
the customer pays by cheque	Y	N	Y	N
the customer is not known	N	N	–	–
ring up the sale immediately	X			
check the customer in the local database		X		
call the supervisor			X	
check the credit card in the credit card database				X

Express the meaning of this decision table using a sequence of conditional statements. You will find in each conditional statement, parts already filled out and parts not filled out. Fill out the non-filled out parts correctly.

IF the item costs less than \$100.00 AND
 the customer pays by cheque AND
 the customer is known

THEN _____.

IF _____ AND

_____ AND

THEN check the customer in the local database.

IF the item costs \$100.00 or more AND
 the customer pays by cheque

THEN _____.

IF _____ AND

THEN check the credit card in the credit card database.

In the last two conditional statements, whether the customer is known or not known

_____.

