

CS6361 Requirements Engineering page 1/8

The University of Texas at Dallas  
Computer Science Program

Test 1

October 16, 1997

Conditions: Closed book Duration: 90 minutes

State assumptions, if there is any

Please write legibly; unreadable answers are NOT answers!

Name:

\_\_\_\_\_ {Please underline last name}

Student Number:

1. \_\_\_\_\_/20

2. \_\_\_\_\_/30

3. \_\_\_\_\_/10

4. \_\_\_\_\_/20

5. \_\_\_\_\_/20

Total \_\_\_\_\_/100

1. [20 marks]

page 2/8

For each of the following ten statements, indicate whether it is true (mark T) or false (mark F).  
(No penalty for a wrong answer)

T This is Test 1 for CS6361.

F 1. Detecting a software error through verification and code inspection is the most effective way to provide a high assurance on the system performance.

F 2. Requirements validation involves verifying that a software architectural design is a "good-enough" solution to the given problem.

F 3. Scenario analysis has been proven effective in building dependable systems in the presence of the "say-do" problem.

F 4. Technical feasibility is a critical criteria for the "goodness" of an executable requirements specification language.

F 5. One major impetus towards the Unified Approach (proposed by Rumbaugh, Booch and Jacobson) has been to cope with "requirements volatility" while having a high degree of requirements traceability.

T 6. The more powerful ontological and epistemological primitives are of the particular requirements modelling language being used, the more knowlege can be captured of the application domain at hand.

T 7. The three critical problems plaguing large industrial software systems, as discovered in a Field study by Curtis and his colleagues, are: changing and conflicting requirements, thin spread of knowledge, and communication and coordination breakdowns.

T 8. A typical tradeoff between an informal requirements modelling language and a formal modelling language has to do with conceptuality vs. (mathematical) rigor.

T 9. Omissions in requirements often result from lack of communication between requirements engineers and domain experts who belong to the subject world.

T 10. If  $A$  is a logical statement,  $A \text{ OR } \neg A$  is always true.

**2. [30 marks]**

page 3/8

Circle the best answer to each of the following questions.

1. The objectives of requirements are:

1. to support system evolution
2. to support verification and validation
3. to achieve agreement regarding the requirements between system developers, customers, and end-users
4. to provide the basis for software design
5. all of the above

2. What is the *least* relevant to the technique of sampling when building a medical insurance system?

1. choosing population elements
2. determining the sample size
3. decomposing goals into subgoals
4. determining the types of data

3. The fundamental processes within requirements engineering do not include:

1. elicitation
2. optimization
3. specification
4. validation
5. none of the above

4. A good requirements specification for an Electronic Commerce System needs to provide answers to:

1. what algorithms should be used?
2. how can users' job security be enhanced?
3. what should be the response time of the system?
4. what data structures should be used?
5. how can the system be implemented?
6. none of the above

5. What is the *least* likely concern of a requirements engineer in choosing among requirements modelling notations when building a patient-monitoring system?

1. ontology of programming languages
2. ontology of the application domain
3. epistemology of the application domain
4. methodological support for using the notation

Circle the best answer to each of the following questions.

6. A requirements document for a telephony system should:

1. use a formal language
2. describe bounds on coupling and cohesion
3. state how the system ought to be verified.
4. describe typical workload on the system
5. none of the above

7. What is included in the four major domains (worlds) of information systems such as a mail tracking system?

1. subject world
2. usage world
3. development world
4. system world
5. all of the above

8. What is the most relevant to domain analysis?

1. software metrics
2. tacit knowledge
3. requirements similarity
4. sampling
5. finite automata

9. What is the *least* relevant to issues with many sources of requirements?

1. the “say-do” problem concerning tacit knowledge
2. validation of software metrics model
3. multiple viewpoints
4. traceability of requirements

10. Due to an electrical malfunction, a helicopter pilot can not determine the current position. The pilot holds a handwritten sign “WHERE AM I?” in the window. People in a tall building drew a large sign “YOU ARE IN A HELICOPTER”. Which is the most related to the problem with this large sign?

1. enterprise requirements
2. system functional requirements
3. system non-functional requirements

3. [10 marks]

page 5/8

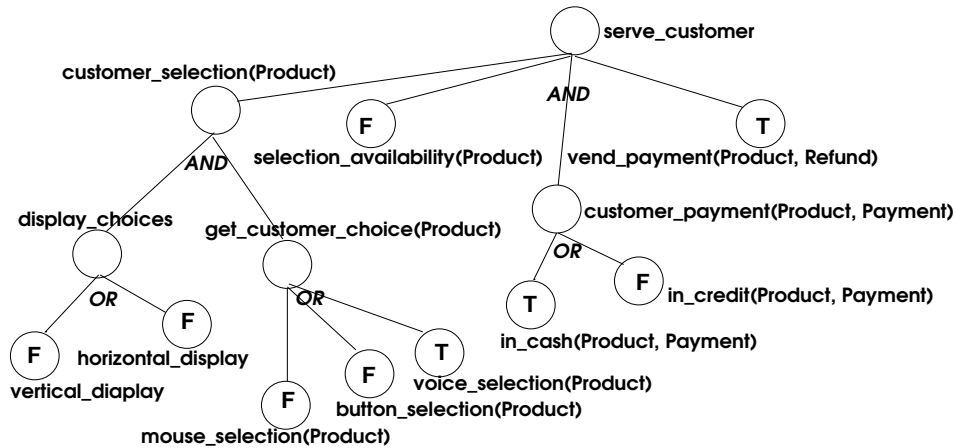
1. Describe briefly what the main causes of a high degree of “requirements volatility” are.

Requirements change because the problem being solved changes.  
because people’s perception changed.  
because some involved persons were not contacted or  
because some involved persons were contacted but  
not in an appropriate manner.

2. Describe briefly what the major issues are in relying on “ethnomethodology” during requirements elicitation.

- . ethical, legal implications, if video-taping is done without notif.
- . observation in an artificial setting, if people are aware of video  
(=> try to observe in a maximum natural setting,  
with minimum interruption)
- . can be too time-consuming to analyze the recording  
(=> use it for critical tasks.)

1. Consider the following AND/OR decompositions of goals into subgoals. Fill in all empty circles with either a "T" or "F".



2. Describe what part of the graph can correspond to design.

All the OR decompositions in the graph.

3. Describe briefly how this kind of graph can help validate requirements.

This kind of graph results from taking a goal-directed approach, where everything is done to meet the goals and justified in terms of the goals. As a result, requirements can be traced back to top-level goals. Hence, traceability is enhanced and justifiability is improved, and consequently requirements validation becomes easier.

4. Describe briefly how an executable specification can be generated from this kind of graph.

Using Prolog, each AND decomposition,  $a \leftarrow \text{AND}(c, d)$ , can be expressed as

$a :- c, d.$

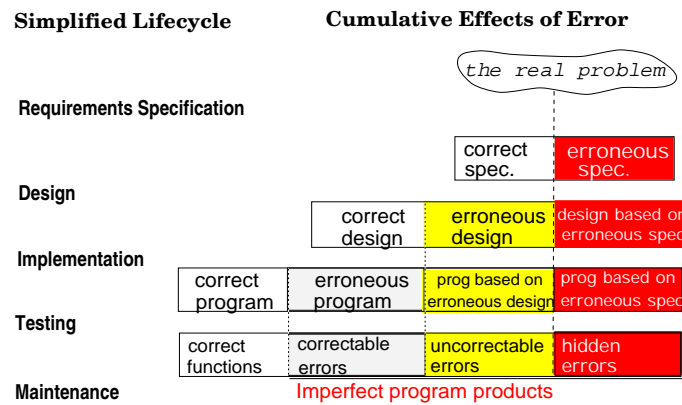
Each OR decomposition,  $a \leftarrow \text{OR}(c, d)$ , can be expressed as

$a :- c.$

$a :- d.$



2. Using error propagation in software lifecycle, briefly explain why it is important to detect a software error as early as possible.



If a software error is detected at a late stage, it means that time and efforts have already been spent on the development of intermediate products.

Also, the later in the development cycle that a software error is detected, the more expensive it will be to repair. In other words, the error has to be discovered in the requirement and the right intermediate products have to be developed/reworked for the corrected part of the requirements.

Therefore, from the viewpoint of both cost and effort, it is important to detect an error as early as possible.

3. In the following, each upper character denotes a requirements statement. Prove that the set of statements is inconsistent.

A AND D	-----	(1)
$\neg C$	-----	(2)
$A \rightarrow B$	-----	(3)
$B \rightarrow C$	-----	(4)
$D \vee A$	-----	(5)

A from (1) --- (6)

B from (3) and (6) --- (7)

C from (4) and (7) --- (8)

--

[] from (2) and (8)