

SE3821 Software Requirements and Specification

Dr. Walter Schilling

Fall, 2013

You may use 1 8.5 x 11 inch sheet of paper with notes and other supporting material for the exam.

1 Exam Scenario

On the exam, questions may be based upon the following scenario. In preparing for the exam, it would be wise to read this scenario and think of the requirements related questions (scenarios, diagrams, etc.) that might be asked. The scenario will be printed on the exam as well.

S and H Landscaping company, founded by two MSOE software engineering professors, is looking to hire a contracting team to design a software application for them. As a business, S and H does landscaping tasks. In the summer time, depending on the services subscribed to by the customer, the company will cut grass on a weekly basis as well as apply fertilizer using one of three plans. The basic plans fertilizes 4 times during the growing season, the silver plan fertilizes 6 times throughout the growing season, and the “gold and green plan” fertilizes 8 times per year plus includes lawn aeration. Contracting with S and H requires the user to agree to a weather station being installed on the premise which will monitor real time weather conditions for the customer’s location.

Landscaping has many coordinated activities which must be managed. For example, certain fertilizers will be ineffective if it rains within 24 hours of application. Thus, S and H, in its dedication to quality, will not schedule fertilizer to be applied if it is going to rain within 24 hours. Fertilizer may, however, be applied if it has rained in the last 24 hours. Weather forecasts are obtained from the National Weather Service.

To comply with local laws, the system will automatically generate a Blackberry e-mail and send it to the customer 24 hours in advance of all fertilizer applications.

The cutting of grass is also impacted by weather as well. Grass should not be cut for 24 hours if more than .05 inches of rain falls or if the temperate drops below 45 degrees Fahrenheit. Grass will be scheduled based upon rainfall indications from the climate monitoring system as well as temperature readings from the climate monitoring system.

In the winter time, S and H will also plow snow for people. Depending on the contract, a snow plow will be dispatched (or scheduled) to clear a driveway and salt the driveway if more than 0.5 inches of snow falls in a 24 hour period. Additionally, a snow plow will be dispatched if the weather forecast calls for freezing rain or if freezing rain has been detected by the weather station.

In order to finance the operation, S and H Landscaping has established a relationship with Pay Pal. Pay Pal will receive bills from the company to be charged to the customers. Bills will automatically be generated monthly by the system based upon the customer’s subscription.

2 Detailed Outcomes for the Exam

1. Week #1

(a) Lecture #1 Introduction

- i. Explain the importance of consistent requirements in a software system.

(b) Lecture #2 What are Requirements

- i. Understand the relationship between MSOE courses and development activities
- ii. List reasons why software fails to be successful.
- iii. Explain the key purpose for the requirements activity.
- iv. Compare and contrast constraints with requirements.
- v. List common sources for constraints.

- (c) Lecture #3 What are Requirements - Part 2
 - i. Explain the steps of the Volere Requirements Process
 - ii. Explain what happens during Project Blastoff
 - iii. Explain the concept of a context diagram
 - iv. Explain the concept of requirements trawling
 - v. Explain how prototypes can be used in the requirements trawling process.
 - vi. Define rationale.
 - vii. Define fit criterion.

2. Week #2

- (a) Lecture #1 Project Blastoff
 - i. Explain the rabbit, horse, and elephant analogy to software development
 - ii. Define the trinity of scope, stakeholders, and goals
 - iii. Define stakeholders and determine the stakeholders for a project from a project description
 - iv. Identify different sponsors for a project
 - v. Explain the concept of a persona
- (b) Lecture #2 Goals and Planned Achievements
 - i. Develop a persona for a project user.
 - ii. Define the goals for a project using a purpose, advantage, and measurement technique
 - iii. Define a set of constraints for a project
 - iv. Explain how one would estimate the cost of a software development project.
- (c) Lecture #3 - No Class due to travel by the professor.

3. Week #3

- (a) Lecture #1 Business Use Cases and Use Case Diagrams
 - i. Construct a context diagram for a given problem.
 - ii. Define business events based on a problem.
 - iii. Explain the relationship between work and business events.
 - iv. Explain the difference between business events and time triggered business events
 - v. Explain the relationship between input and output and business events.
 - vi. Explain the relationship between business use cases and product use cases
- (b) Lecture #2 Use Cases and Use Case Diagrams
 - i. List the 14 types of UML diagrams
 - ii. Define Actor
 - iii. Define a use case
 - iv. Interpret the meaning of a use case diagram.
 - v. Explain the relationship between Use Case Diagrams and Use Case Scenarios
- (c) Lecture #3 Trawling for Requirements - Part 1
 - i. Define Business Event
 - ii. Define time triggered business events
 - iii. Translate a context diagram into a list of business events and input and output views
 - iv. Explain the relationship between work response and use cases.

4. Week #4

- (a) Lecture #1 Trawling for Requirements
 - i. Explain what is meant by the concept of trawling for requirements.
 - ii. Explain where requirements come from
 - iii. List the roles of the requirements analyst.
 - iv. Define unconscious requirements
 - v. Define undreamed of requirements
 - vi. Explain the concept of a brown cow model

- (b) Lecture #2 Trawling for Requirements - Part 2
 - i. Define the concept of an apprenticeship
 - ii. Explain how observation can be used to determine requirements
 - iii. Explain the concept of a mind map
 - iv. Construct a mind map for a system
 - v. Explain how a mind map may help to determine requirements for a product
 - vi. Derive requirements for a system based upon a persona.
 - vii. Explain how to construct a quick and dirty process model.
- (c) Lecture #3 Trawling for Requirements - Part 3
 - i. Explain the essence of requirements gathering
 - ii. List guidelines for making interviews more effective
 - iii. List guidelines for holding effective interviews
 - iv. Derive requirements for a system based upon a persona.
 - v. Critique a trawling techniques, and select the appropriate method based upon the projects needs.

5. Week #5

- (a) Lecture #1 Scenarios and Requirements
 - i. Explain why we use scenarios as part of the requirements process
 - ii. Define Scenario
 - iii. Explain how to translate a story into a scenario.
 - iv. Construct a use case scenario from a story.
 - v. List the symbols used on an activity diagram.
 - vi. Explain how one can draw an activity diagram from a use case scenario.
 - vii. Explain the concept of a Misuse Case and a Negative Scenario
- (b) Lecture #2 Understanding the Real Problem
 - i. Explain why capturing the essence of the problem is important to rabbit projects
 - ii. Critique a requirement as to whether it captures the essence of the problem or a solution to a problem.
 - iii. Explain the concept of a swimlane
 - iv. Explain how removing swimlanes can help to capture the essence of the problem.
 - v. Explain the risk of not innovating
 - vi. Define systemic thinking
 - vii. Define value demand and failure demand
 - viii. Explain how to run an innovation workshop
- (c) Lecture #3 Midterm Exam

6. Week #6

- (a) Lecture #1 Oral Presentations
- (b) Lecture #2 Oral Presentations
- (c) Lecture #3 No class

7. Week #7

- (a) Lecture #1 Starting the Solution
 - i. Explain what types of projects would benefit from different trawling techniques.
 - ii. List problems that may occur with iterative development.
- (b) Lecture #2 Functional Requirements
 - i. Define functional requirements
 - ii. Define Scenario
 - iii. Explain how to translate a story into a scenario.
 - iv. Construct a use case scenario from a story.
 - v. List the symbols used on an activity diagram.
 - vi. Explain how one can draw an activity diagram from a use case scenario.

- (c) Lecture #3 Documenting the Requirements
 - i. Document software requirements using snow cards
 - ii. Define Fit Criteria
 - iii. Define Rationale for a requirement
 - iv. Construct appropriate descriptions, rationale, and fit criteria for a software requirement.
 - v. Explain the steps of requirements analysis immediately adjacent to writing the requirements
8. Week #8
- (a) Lecture #1 Fit Criteria and Rationale
 - i. Define Fit Criteria
 - ii. Define Rationale for a requirement
 - iii. Define fit criterion for a set of requirements
 - iv. Explain different techniques for expressing fit criteria.
 - v. Explain how fit criteria can be expressed for implementation constraints.
 - (b) Lecture #2 Non-Functional Requirements
 - i. Define non-functional requirements
 - ii. Explain how non-functional requirements may cause additional functional requirements to be assigned.
 - iii. Explain the relationship between non-functional requirements and use cases
 - iv. List the types of non-functional requirements
 - (c) Lecture #3 Quality Gateway
 - i. Understand the importance of quality
 - ii. Define Requirements Creep
 - iii. Define requirements leakage
 - iv. Define gold plating in terms of requirements
 - v. Identify requirements within a specification which may be gold plated.
9. Week #9
- (a) Lecture #1 Requirements and Iterative Development
 - i. Explain how prioritization occurs in iterative development
 - ii. Explain what happens as business needs are analyzed
 - iii. Define user stories
 - iv. Define development backlog
 - (b) Lecture #2 Requirements Reuse
 - i. Explain how requirements can be reused.
 - ii. Explain when requirements reuse should be considered.
 - iii. Given a problem, think of requirements which can be reused.
 - iv. Explain how a requirements pattern can be documented.
 - (c) Lecture #3 Requirements Completeness
 - i. Explain why reviewing the requirements specification tends to be an iterative process
 - ii. Explain, at a high level, the concept of a Fagan Inspection
 - iii. Show how business events can be checked for completeness.
 - iv. Explain the meaning of a CRUD check and how it can be used to check requirements scenarios.
 - v. Explain when requirements should be prioritized and list prioritization criteria.
 - vi. List categories of risk to a software project
 - vii. Explain how risk can be quantitatively assessed for a project.
10. Week #10
- (a) Lecture #1 Putting it all together
 - i. Critique a requirements document based on the materials taught in class
 - ii. Effectively communicate issues with requirements documents
 - iii. Determine if a requirements document is useful to the development team.
 - (b) Lecture #2 Course Evaluations
 - (c) Lecture #3 Exam review and Evaluation
 - i. Evaluate the exam status.