

CS510 (SOFTWARE REQUIREMENT AND SPECIFICATIONS)

FINAL TERM SOLVED QUESTIONS

SOLVED BY FATIMA ALI

Question: Define Goal model completeness criteria?

Answer:

1. A goal model is said to be complete with respect to the refinement relationship 'if and only if' every leaf goal is either an expectation, a domain property or a requirement.
2. A goal model is said to be complete with respect to the responsibility relationship 'if and only if' every requirement is placed under the responsibility of one and only one agent.

Question: Why we use KAOS Model?

Answer:

KAOS Model is used to define and document concepts of the application domain that are important concerning the recognized requirements and provide static constraints on operational system.

Question: what are the types of objects?

Answer:

There are three types of objects.

1. entities

They represent independent and passive objects. For example: elevator door, buttons, cage etc. passive means, the cant perform any operation. Independent means that their descriptions don't need refer to other objects of the model.

2. agent

They represent independent and active objects. For example: elevator company, elevator controller, passenger etc. They are active. It's meaning they can perform operations. Operations usually imply state transitions on entities.

3. associations

They are dependent and passive objects. They are Dependent because their descriptions refer to other objects. For example: the "At" association links a Cage to a Floor. An instance of that association would hold if cage 'c' is currently located on floor 'f'. To express association, we need further objects. They are also passive so they can't perform operations.

Question: Define KAOS Operational Model?

Answer: The KAOS operation model describes all the behaviors that agents need to fulfill their requirements. Behaviors are expressed in terms of operations performed by agents.

Question: How behaviors are expressed?

Answer: Behaviors are expressed in terms of operations performed by agents.

Question: how operation work on objects?

Answer: Operations work on objects as they can create objects, trigger object state transitions and activate other operations by sending an event. Operations can directly be expressed by stakeholders during the interviews. It can be identified by looking at all the existing requirements.

Question: How operations can expressed and identified?

Answer: Operations can directly be expressed by stakeholders during the interviews. It can be identified by looking at all the existing requirements.

Question: How operations, concerned objects and events are represent?

Answer:

1. Operations: Operations are represented as ovals.
2. Concerned objects: Concerned objects are connected to the operations by means of Input and Output links.
3. Events: Events are represented as those traffic signs that are used to indicate direction.

Question: Define completeness criteria?

Answer: Completeness means to be complete a process diagram. It must specify the following requirements.

1. The agents who perform the operations.
2. The input and output data for each operation.

Question: When operations are to be executed?

Answer: All operations are to be justified by the existence of some requirements through the use of operationalization links. Blue filled circle represent the operationalization links.

Question: Define KAOS Responsibility Model? How to build a responsibility diagram?

Answer: The responsibility model contains all the responsibility diagrams. It describes for each agent, the requirements and expectations that he's responsible for, or that have been assigned to him.

To build a responsibility diagram: To build a responsibility diagram, the analyst reviews the different requirements and expectations in the goal model and gives an agent to each of them. After all requirements and expectations are allocated to a responsible agent, a diagram is generated for each agent, listing all requirements and expectations that he is been assigned.

Question: Responsibilities of Elevator Company?

Answer:

1. fire proof cage
2. equipment to protect against fire available in the cage
3. emergency power available
4. elevator equipped with a break down alarm
5. button based interface provided
6. elevator cage has a door
7. cage button include a stop button
8. software secured
9. way to escape provided
10. elevator equipped with floor doors

Question: responsibilities of elevator controller?

Answer:

1. overweight conditions reported to the passenger
2. cage door closed while moving
3. elevator stopped at passenger destination
4. elevator kept on current floor, doors open, until overweight condition disappear
5. emergency condition reported
6. passenger informed of their calls status
7. stopped on calling floor
8. moving elevator stopped on the next floor in case of fire signal
9. emergency lights on, when needed
10. door remain closed when cage not stopped on a floor level
11. button command detected
12. elevator resumed
13. elevator stopped upon power failure
14. no door opening while moving
15. weight condition checked before moving
16. door locked open on the floor level in case of fire signal or alarm
17. elevator stopped

Question: define responsibility model notation.

Answer: responsibility model represented by a red filled circle.

Question: How requirement change management?

Answer: Requirements get changed during the progress of development. It is almost impossible to stop the requirements from changing. Different software development approaches tackle changing requirement in different ways. Unlike Waterfall or document driven approaches of software development, agile methodologies welcome change during the course of software development but simultaneously manage the changes in a systematic manner.

Question: what are the reasons for requirement changes?

Answer:

1. Lack of domain knowledge at start.
2. Change in customer prioritization.
3. Change in platform or environment.
4. Inconsistent requirements.
5. Change due to expensiveness.
6. Change due to difficult to implement.
7. changes in organization

Question: define Agile Manifesto?

Answer:

1. Instead of tools, techniques and processes, Agile focus and prefer more on individuals capabilities and expertise and their interaction.
2. Instead of formality of having large documentation, Agile concentrate to have working application.
3. Instead of contract negotiation, Agile highly contemplate to have a very close relationship with customer in terms of continuous and constant conversation
4. Instead of following a plan, Agile assures to adopt and respond frequently over change request

Question: Define Extreme Programming (XP) and there rules?

Answer: Extreme Programming is the most famous agile technique. XP use story cards for elicitation. A user story is the description that provides business value to the customer. These rules are as follows:

1. The Planning Game
2. Small Releases
3. Metaphor
4. Simple Design
5. Tests
6. Refactoring
7. Pair Programming
8. Collective Ownership
9. Continuous Integration
10. 40-hour Week
11. On-site customer
12. Coding Standard

Question: Define Scrum?

Answer:

Scrum: Scrum is another popular agile technique used to develop and manage software.

Question: Define Rational Unified Process (RUP)?

Answer: (RUP) Document Driven approach, RUP Phases, RUP Disciplines:

1. Extensive planning
2. Codified Process
3. Heavy Documentation
4. Big Design up Front

Strengths:

1. Straightforward
2. methodical and structured nature
3. Predictability, stability and high assurance

Weaknesses or disadvantages:

1. Slow adaptation to rapidly changing business requirements
2. A tendency to be over budget
3. A tendency to behind schedule
4. Failed to provide dramatic progress in productivity, simplicity and reliability

Question: Define XP and Scrum?

Answer: XP and Scrum (Agile Software Development)

Approaches:

1. Iterative and Incremental development
2. Customer collaboration
3. Frequent Delivery
4. Light and fast development
5. Light documentation

Strengths:

1. Short development cycle
2. Higher customer satisfaction
3. Low bug rates
4. Quick adaptation to rapidly changing requirements
5. Highest Priority Work, Constant Feedback, Control over Cost and Schedule.

Weaknesses:

1. Significant document reduction
2. Organizational structure
3. Not suitable for large scale systems
4. Heavy dependence on individual knowledge
5. Managed prioritization
6. Not suitable for critical safety systems
7. Frequent change effect cost and schedule

Question: Define Agile (XP and Scrum) requirements change management process?

Answer: The agile change management process handles changes in the start of each iteration of the development cycle. It may be a sprint in Scrum or iteration in XP and so on. The key stakeholders in change management process are the managers, developing team and customers or product owner. As Agile's development period is short for a particular iteration so it is understood that all the requirements cannot be implemented in one go. Therefore there is a pile or stack of requirements and the relevant stakeholders have to decide which requirements to implement in the one iteration. The prioritization is also a continuous process in agile development and the requirement stack is continuously updated as a result of update.

Question: Lifecycle of change management in agile?

Answer:

Start: In the start of each iteration, the team takes the highest priority requirement that can be completed in the specified iteration period. The requirement is now going to be well understood with the help of customer and other related stakeholders etc. Required planning, documentation or modeling can also be done at the starting stage to achieve the goal within the specified budget and time.

Middle: During development they may take help from customer as well as from other related stakeholders to have better understanding of the requirement. The goal is to build the software that best meets the requirement.

End: The working product developed can be organized and it is preferable to organize to take the feedback from the end users. The acceptance can also be run on the developed software so the necessary quality can also be ensured.

Question: What Kind of Tool Do We Need for Lifecycle of change management in agile?

Answer:

1. Word processor (Microsoft Word with templates).
2. Spreadsheet (Microsoft Excel)
3. Industrial strength or commercial RM tools (IBM/Teleological DOORS, IBM Requisite Pro, Borland Caliber RM)
4. Internal tools such as GenSpec (Hydro-Quebec)
5. Open source RM tools such as OSRMT
6. Bug tracking tools (Bugzilla)
7. Collaboration tools (TWiki)

Question: Comparison between Agile and Conventional Philosophy?

Answer:

Approaches	Conventional philosophy	agile
Changing requirement	88%	13%
Requirements Gathering Process	complete Specification	incrementally
Means of Communication	document	Onsite customer
Contracts with Clients	Strict	flexible
Attitudes towards Change	Difficult task	Welcome
Relationship with Customer	satisfactory	Close

Question: Freezing the Requirements?

Answer: unlike traditional approaches like waterfall model where we do not have the privilege to request requirement changes after the start of development cycle, in agile the customer or the product owner have an opportunity to add, modify or remove any requirement from the requirement stack. In fact Agile has left a process in every technique to accept change in an organized way for next iteration instead of forcing to the new requirement. XP and OpenUP

allows accepting change to an extent during the development iteration however recommends suggesting the requirement for next iteration.

Question: Requirements Prioritization?

Answer: Requirements Prioritization

1. Nature of Agile development lifecycle demands
2. feedback after every development iteration
3. Stakeholders status in the beginning
4. agile development framework lets the stakeholders to re-prioritize the requirements
5. Factors like market uncertainty, technical uncertainty, project duration and project budget that demands that the requirements should be analyzed.

Question: define Value Oriented Prioritization?

Answer: Value Oriented Prioritization:

Requirement prioritization was named as “Value Oriented Prioritization”. This technique suggests that the company or the stakeholders should identify the business value areas like Sales, Marketing, Strategic, and Customer Retention etc. A positive numeric value should be associated with each of these business value areas. Then, values are also given to each requirement after negotiation and consultation with all the relevant stakeholders. With identifying the business value areas, the related risks should also be identified and a negative value should also be given to each risk.

Question: how to evaluate requirement?

Answer: Evaluation: The process of systematically collecting data that informs us about what it is like for a particular or group of users to use a product for a particular task in a certain type of environment.

Question: Why, what, where, and when to evaluate?

Answer:

Why: to check that users can use the product and that they like it.

What: a conceptual model, early prototypes of a new system and later, more complete prototypes.

Where: in natural and laboratory settings.

When: throughout design; finished products can be evaluated to collect information to inform new products.

Question: define usability?

Answer: Usability:

It has been defined by the International Standards Organization (ISO) as:

The extent to which the product can be used by specified users to achieve specified goals with

1. effectiveness
2. efficiency
3. satisfaction

Question: how usability evaluation?

Answer: Usability Evaluation

Metrics used to measure Usability:

1. Time to complete a task
2. Fraction of task completed
3. Fraction of task completed in a given time
4. Number of errors
5. Time spent on errors

Question: Name the Usability Evaluation Methods?

Answer:

1. Testing
2. Inspection
3. Inquiry

Question: What is traceability?

Answer: Traceability is the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor - successor or master subordinate relationship to one another.

Question: What is Requirements Traceability?

Answer: The requirements traceability is the ability to define and follow the life of a requirement, in both direction i.e. a forward and backward direction.

Question: Why is requirement Traceability?

Answer: We need requirement traceability for the following reasons:

1. To finding missing requirements
2. To finding unnecessary requirements
3. To certification and compliance
4. To change impact analysis

5. To maintenance
6. For project tracking etc.

Question: Classification of Requirement Traceability

Answer:

1. Backward-from traceability

Links requirements to their sources i.e. documents or people.

2. Forward-from traceability

Links requirements to design and implementation components.

3. Backward-to traceability

Links design and implementation components back to requirements.

4. Forward-to traceability

Links requirements back to their sources.

Question: Categories of requirement traceability?

Answer: Requirements-sources traceability

Links the requirement and their sources which specified the requirement.

Requirements-rationale traceability

Links the requirement with a description of why that requirement has been specified.

Requirements-requirements traceability

Links requirements with other requirements which are, in some way, dependent on them.

Requirements-architecture traceability

Links requirements with the sub-systems where these requirements are implemented.

Requirements-design traceability

Links requirements with specific hardware or software components in the system, which are used to implement the requirement.

Requirements-interface traceability

Links requirements with the interfaces of external systems, which are used in the provision of the requirements.

Question: How is tracing performed?

Answer: Each element (requirement, design attribute, test, etc.) is given a unique identifier. Linkages done manually and managed by a CASE tool.

Question: define Traceability matrix? Where it is used?

Answer: A traceability matrix is a document that co-relates any two-baseline documents that require a many to many relationship to check the completeness of the relationship.

It is used to track the requirements and to check the current project requirements are met.

Question: what are the Types of Traceability Matrix?

Answer: Types of Traceability Matrix:

1. Forward traceability
2. Backward or reverse traceability
3. Bi-directional traceability (Forward + Backward)

Question: What are the Advantage of Requirement Traceability Matrix?

Answer:

1. It confirms 100% test coverage.
2. It highlights any requirements missing or document inconsistencies.
3. It shows the overall defects or execution status with a focus on business requirements.

Question: what are the requirement traceability tools?

Answer:

1. CASE Tools
2. Caliber-RM
3. DOORS

Question: what are the characteristics of CASE Tools?

Answer: Characteristics of CASE Tools:

1. Hypertext linking
2. Unique identifiers
3. Syntactical similarity coefficients

CASE Tools problems:

Hypertext linking and syntactical similarity does not consider context.

Unique identifiers do not show requirement information.

Choosing architecture view and classification schemas will always be manual.

Question: Caliber-RM

Answer: Caliber-RM:

1. Centralized repository.
2. Requirements traceability across the lifecycle.
3. Impact analysis.

Question: DOORS

Answer: Doors is Telelogic. It capture, link, trace, and manage. It is used for large applications.

Question: What is requirement document?

Answer:

1. Software requirements specifications or SRS.
2. The requirements document is a formal document used to communicate the requirements to customers, engineers and managers.

Requirements documents includes:

Answer:

1. The services and functions which the system should provide.
2. The constraints under which the system must operate.
3. Overall properties of the system.

Users of the requirements documents:

Answer:

1. System customers
2. Managers

3. System engineers
4. System test engineers
5. System maintenance engineers

Question: what are the characteristics of good SRS?

Answer: Characteristics of good SRS

1. Correct
2. Unambiguous
3. Complete
4. Consistent
5. Verifiable
6. Modifiable
7. Traceable

Question: Verification and Validation?

Answer: Verification and Validation

Verification: Process in which we check a product against its specifications for example White box, black box testing.

Validation: Process in which we check expectations of the users who will be using it. For example Inspection, Formal Technical Review.

Question: Is it possible software without any defects?

Answer: No, it is almost impossible to develop software without having any defect. Software and defects during software development. It is impossible to build a product without presence of any defects and these two cannot be separated.

Question: Define Black box testing?

Answer: In this type of testing, a component or system is treated as a black box and it is tested for the required behavior. This type of testing is not concerned with how the inputs are changed into outputs. As the system's internal execution details are not visible to the tester. He gives inputs using an interface that the system provides and tests the output. If the outputs match with the expected results then the system is fine else there a defect is found.

Question: Structural testing (white box)?

Answer: Structural testing (white box):

It is opposite to black box testing, in structural or white box testing we look inside the system and evaluate that what it consists of and how is it executed. The inner of a system consists of design and structure of code and its documentation etc. So, in white box testing we consider these internal structures of the program and devise test cases that can test structure.

Question: Define Defect Removal Process?

Answer: Defect Removal Process

Following steps I take to check the presence of the defects:

1. I will run the scenario as described in the bug report and try to reproduce the defect.
2. If the defect is reproduced in the development environment, then I will identify the root cause and fix it and then send the patch to the testing team along with a bug resolution report.

Question: Cyclomatic complexity?

Answer: Cyclomatic complexity

E = Number of Edges

N = Number of Nodes

$V(G) = E - N + 2$

Current paper 2017-feb final term

Question:

E = Number of Edges = 8

N = Number of Nodes = 6

Using the above measures, calculate the cyclomatic complexity, $V(G)$ of Graph G .

Answer:

$V(G) = E - N + 2$

$V(G) = 4$

Question:

Mention at least four users of Software requirement specification.

Question:
Mention one core purpose of KAOS responsibility diagram.

Question:
List four types of models in KAOS.

Question:
What steps would you take to check the presence of the defects and what you do to eliminate those defects?

Question:
Mention at least three parts of “Introduction” section of Software requirement specification template (standard 830-1993 by IEEE)

Question:
In context of KAOS operation model, mention the symbols for following notations:

1) Operations

2) Events

Answer:

1. oval
2. pentagon

Question:
Consider that our goal is “To satisfy a request for a given service”. Now, in context of KAOS goal model terminologies, what appropriate name you will give to this goal from the following list? Also mention the reason behind it.

- 1) Satisfy service request
- 2) Service request satisfied

Question:
In context of KAOS operation mode, specify the completeness criteria.

Question:
You have read about two main types of testing: Black box testing and white box testing. Differentiate between these two types.

Question:
Requirements traceability is categorized into different categories. You are required identify and explain three of these categories

Question:

In KAOS object model, what do the following symbols represent?

1) Parallelogram

2) Yellow circle

Answer:

1. represent the requirement
2. represent the expectations