

Sample Questions for Examination 1

Software Processes, Product Design and Engineering Design

1. Indicate whether each of the following statements is true or false:

- (1) _____ The system is never an actor in a Use Case Diagram.
- (2) _____ A "high resolution" (a.k.a., "low abstraction") Use Case Diagram will tend to have a large number of use cases for the amount of functionality in the system.
- (3) _____ A non-functional requirement is a statement that a product must have certain properties.

2. Choose the best answer to each of the following:

- (1) _____ A system consists of:
 - a. A set of objects
 - b. The attributes of objects
 - c. The relationships between objects
 - d. All of the above
- (2) _____ Which of the following is a type of UML diagram:
 - e. Activity
 - f. Context
 - g. User Interface
 - h. None of the above
- (3) _____ Requirements can be recorded using:
 - i. Natural language
 - j. Formal methods
 - k. Both of the above
 - l. Neither of the above
- (4) _____ A model is typically:
 - m. An abstraction of a target system
 - n. A composition of a target system
 - o. An inception of a target system
 - p. All of the above
- (5) _____ A requirement is:
 - q. A statement of a problem
 - r. A statement of a solution
 - s. Both of the above
 - t. Neither of the above

3. Based on an analysis of competitive products, you have determined that your new entry into the market will only be successful if a particular task involving the graphical user interface (GUI) can be completed within 3 seconds.
 - a. In the Model Human, what are the three processors that are involved in GUI tasks?
 - b. Fitt's law can be stated as follows:

$$T = a + b \cdot \log(D/S + 0.5)$$

where T is the time to acquire a target, D is the distance to the target, and S is the size of the target. Explain the significance of Fitts's law.

4. Briefly define the following:

Design Pattern

Niche Market

Software Engineering Design

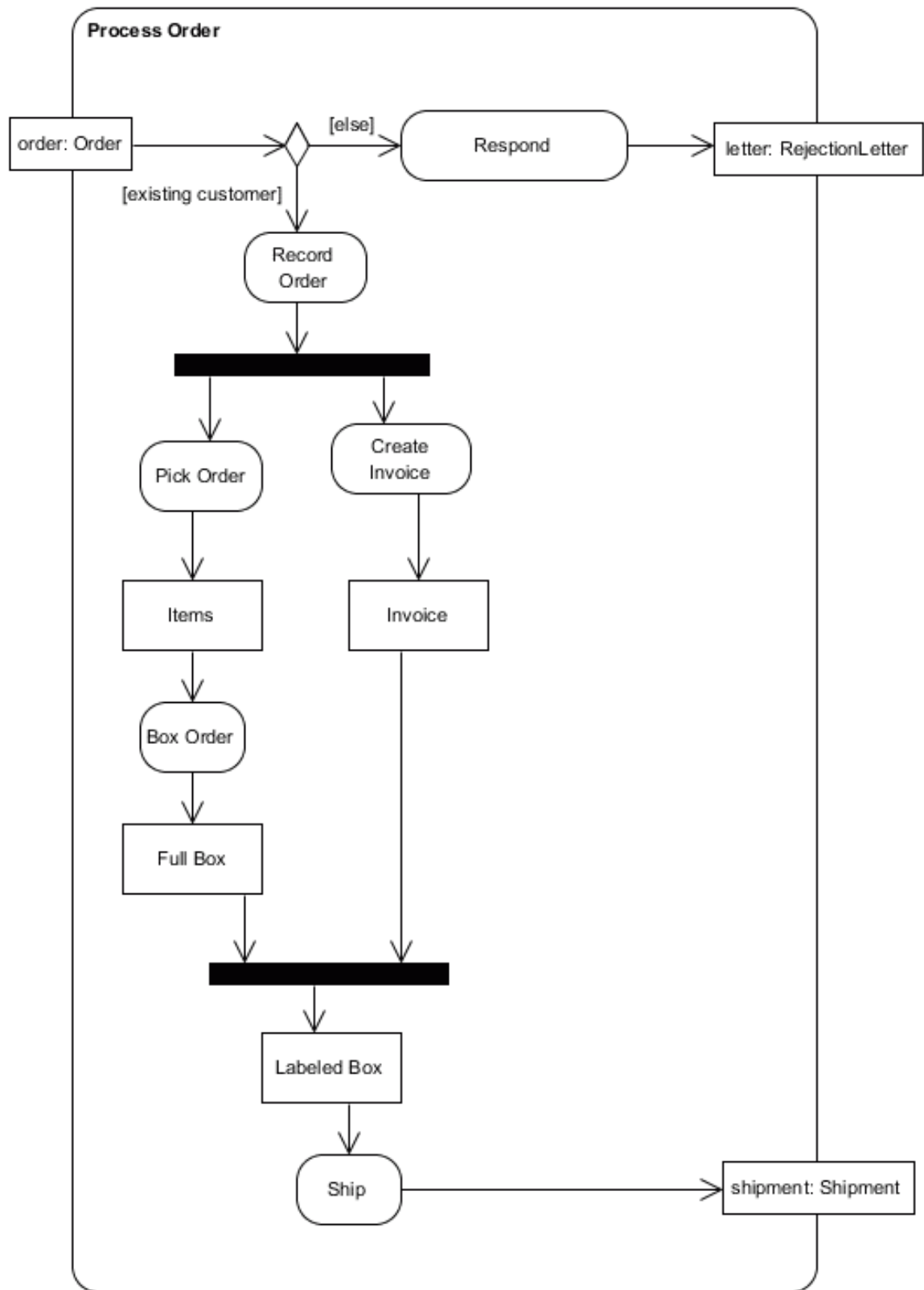
Software Product Design

User Interface

5. Software engineering considers both technical matters and non-technical matters. List two technical matters and two non-technical matters that are within the domain of software engineering.
6. Indicate the best match for each of the following:

- | | |
|---|-----------------------|
| _____ The smallest parts of a system | a. Atoms |
| _____ A model of how a system changes over time | b. Descriptive Model |
| _____ A prioritized list of features/requirements | c. Dynamic Model |
| _____ A "heavyweight" software process | d. Prescriptive Model |
| _____ A model of what should happen | e. Product Backlog |
| _____ Are we building the right project? | f. Scrum |
| _____ Are we building the product right? | g. Static Model |
| | h. Validation |
| | i. Verification |
| | j. Waterfall Process |

7. Is Scrum an iterative software process? Is it an incremental software process? Justify your answers.
8. Consider the following UML Activity Diagram.



- a. Identify all of the activities in this diagram.
 - b. Identify all of the object/data nodes in this diagram.
 - c. Identify all of the actions in this diagram.
 - d. Identify all of the decision nodes in this diagram.
 - e. Identify all of the fork nodes in this diagram.
 - f. Identify all of the join nodes in this diagram.
 - g. Identify a control flow in this diagram.
 - h. Identify a data flow in this diagram.
 - i. Can "Pick Order" and "Create Invoice" occur at the same time?
 - j. Can "Record Order" and "Ship" occur at the same time?
9. Discuss the ways in which products can be categorized.
 10. Discuss the differences between needs and requirements (as we have used these terms in this course).
 11. Explain the *principle of least astonishment* as it relates to user interface design.
 12. Explain why (using what you know about human vision) you should not use red text on a blue background.
 13. One could use UML Activity Diagrams instead of Use Case Descriptions to provide the details of how actors interact with the system. Explain why (natural language) Use Case Descriptions are more appropriate than Activity Diagrams for this purpose during the product design phase. (Hint: Think about how they are used and who uses them.)
 14. Though many people mistakenly omit them from Activity Diagrams, explain why it is important to include Merge Nodes and not just have multiple Flows enter an Action Node directly.
 15. Create a Use Case Diagram (in UML) for the following explanation of the workings of an Internet auction system:
 1. Types of users
 1. Anyone
 1. Anyone may use the search features of the system.
 2. Anyone may look at an auction's information.
 2. Members

1. Only members may bid or place items for sale.
2. All members must register with the system.
3. Members must supply their name and a valid e-mail address.
4. After registering, the system will create an account for the member.
 1. A password will be mailed to the e-mail address specified.
5. Members must log in to bid or place an item for sale.
6. Members who forget their password can have it re-mailed to them.

2. Auctions

1. An auction involves an item, a seller, and zero or more bidders.
2. Items and sellers
 1. Sellers put up items for auction
 2. The item must include a name, a closing time, and a minimum bid.
 3. The item may include a description and a picture.
 4. Sellers may have any number of auctions active at one time.
3. Bids and bidders
 1. Any member may bid in any auction.
 2. Bids may be placed at any time before the closing time.
 3. A bid must be at least the minimum bid, and higher than any bid so far.
4. Ending an auction.
 1. No matter how an auction ends, it is immediately removed from the list of active auctions.
 2. If no bids are placed before the closing time, the auction is closed unsuccessfully and the seller notified by e-mail.
 3. If at least one legal bid has been placed before the closing time, the auction is closed successfully. The winning bidder and the seller are both mailed each other's contact information and the winning bid.

4. The seller may cancel the auction up to 24 hours before the closing time. All bidders on this auction will be mailed a notice of the cancellation.

3. Searching

1. Any user may search through active auctions by keyword.
2. All active auctions with name or description containing the keyword are presented to the user.
 1. The auctions are sorted by closing time in chronological order.
 2. The user may click a link to go directly to the auction.

16. The following Use Case Description was written during the design of a card-based payment system:

1. Use Case:

2. PayForProductWithCard

3.

4. Actors:

5. User, Bank, Cashier

6.

7. Trigger:

8. User inserts Card

9.

10. Precondition:

11. The Cashier has taken the
product from the User

12. and entered the price of the
product into the System.

13.

14. Basic Flow:

15.

16. 1. The System reads
 information off of the Card

17.

18. 2. The System prompts the
 User to enter the type of Card

19. (presenting the options
 "Credit" and "Debit" and "Cancel").

20.

21. 3. The System displays the
 amount of the payment

22. (in dollars and cents).

23.

24. 4. The System prompts the
 User to approve the payment (presenting

25. the options "Yes" and
 "No").

26.

27. 5. If the User approves the
 payment a transaction is sent to the

28. Bank and the Cashier is notified to give the product to the User.

29.

30. Extensions:

31.

32. 5a. If the User does not approve the payment, the Cashier

33. is notified to keep the product and the System is reset.

Identify all of the problems in this use case description and write an improved version.

17. Draw the UML Class Diagram for the Composite Pattern. Explain all of the relationships in the diagram.
18. I've developed an email system called `JMUmble`. In this system, arriving messages are handled by a `PostOffice` object. Depending on how the system is configured at runtime, one or more objects might need to know when a message arrives. I have currently implemented several such classes: `ScreenFlasher` (which makes the entire screen flash -- you always know when a message has arrived), `PopularityTimer` (which starts a clock that show the amount of time since the most recent message arrived), and `Mumbler` (which uses speech generation to read the name of the person that sent the message -- this is where the system got its name). Use the Observer Pattern to develop a class model of this system (in UML). You do not need to include the attributes of each class, only the operations/methods. Include comments that describe each operation/method.
19. Compare and contrast the "Pipe and Filter" architectural style with the "Layered" architectural style.
20. Explain how one distinguishes the client from the server in a client-server architecture.

Comprehensive Questions

21. In order for the grammar checker in your word processor to work correctly, many different modules must work correctly.
 - a. What kind of UML diagram would you use to describe the static structure of such a system?

- b. What kind of UML diagram would you use to describe the dynamic behavior of such a system?
- c. How should you verify such a system? (Note: You must discuss both unit testing and system testing. You may also need to discuss other aspects of verification.)

22. Indicate the best match for each of the following:

- | | |
|--|--|
| _____ Creating a requirements specification | a. Product Design - Analysis Phase |
| _____ Defining components of the user interface | b. Product Design - Resolution Phase |
| _____ Selection of an architectural style | c. Engineering Design - Analysis Phase |
| _____ Soliciting user needs | d. Engineering Design - Resolution Phase |
| _____ Selection of a design pattern | |
| _____ Characterizing users based on how they will interact with the system | |

Tools

23. Choose the best answer to each of the following:

- (1) _____ Visual Studio Code is:
- a. A Java documentation tool.
 - b. A WWW browser.
 - c. An Integrated Development Environment
 - d. All of the above
 - e. None of the above
- (2) _____ Lucid Chart is:
- f. A compiler for Java.
 - g. A debugging tool for Java.
 - h. An Integrated Development Environment
 - i. A GUI Sketch/Wireframe Tool
 - j. All of the above
 - k. None of the above