SE463 Final Assessment - Answers

December 9, 2024

Question 1

a. Identify in EM1 and label with "R" the portions, if any, that constitute the R in "D, S ⊢ R". *R:* The final assessment should resemble a closed book, timed, and proctored in-person exam.

Explanation: This describes the desired outcome or requirement for the final assessment format. It focuses on *what* the assessment should be like, not *how* to achieve it. It is stated in terms of the environment (the assessment process).

b. Identify in EM1 and label with "S" the portions, if any, that constitute the S in "D, S ⊢ R". *S:*

1. We would send a pdf of the exam say 30 minutes before it is to start.
2. You would print a hard copy where you are.
3. You would write the exam with pen or pencil, while being connected to a zoom meeting with a proctor who will ascertain that you are following normal closed-book final exam rules, and in particular are not using any online resources such as classmates. The proctor will be your TA or Dan or Derek, in the case of TAs with too many teams.
4. If you are registered with AccessAbility, the time for the exam will be adjusted according to the multiplier given to us by AccessAbility.
5. When you finish or time is up, you will scan or photograph your exam and email the scan or photo to the class e-mail address or to CrowdMark.

Explanation: These points describe the proposed system's behavior to meet the requirement (R). They outline the steps involved in conducting the assessment, expressed in terms of shared phenomena (e.g., sending a PDF, printing, Zoom meeting, scanning).

c. There are many assumptions about the environment hidden in EM1 that are necessary for S to entail R. List as many unique assumptions as you can find (I found 13.). Give each unique assumption a unique label "D1", "D2", ...

- **D1:** Each student has access to a printer.
- **D2:** Each student has a reliable internet connection that will last the duration of the exam.
- **D3:** Each student has a device capable of connecting to a Zoom meeting.
- **D4:** Each student has a device with a camera for Zoom proctoring.
- **D5:** Each student has a quiet and suitable environment to take the exam.
- **D6:** Each student has the means to scan or photograph their completed exam.
- **D7:** Each student is in a time zone where the exam can be taken at a reasonable hour.
- **D8:** The proctors (TAs, Dan, or Derek) are available and have reliable internet connections.
- **D9:** The class email or CrowdMark will be functional for exam submission.

- **D10:** Each student has access to a writing utensil (pen or pencil).
- **D11:** Each student is able to print the exam within 30 minutes.
- **D12:** Students will not use any unauthorized resources during the exam.
- **D13:** All students are aware of the exam procedures and requirements.

Explanation: These are the domain knowledge (D) – guaranteed properties of the environment assumed to be true – necessary for the specification (S) to fulfill the requirement (R). If any of these assumptions are false, the proposed system may not work as intended.

d. Let n be the number of domain assumptions that you found for (c) above. Identify in ASR and label with ”Dn+1”, ”Dn+2”, ’, ..., as many portions of ASR as you can find (I found 5.) that imply some additional assumptions about the environment, not mentioned in your answer to (c) above, that are necessary for S to entail R. Each unique assumption should receive a unique label.

- **D14:** ”I'm wondering what we are supposed to do if we lose wifi connection during the zoom meeting?” (Implies: Students know how to contact the proctor or course staff in case of technical issues).
- **D15:** ”Can the 30 minutes limit be extended a little bit?” (Implies: Students have enough time between receiving the exam and starting it to print it and prepare).
- **D16:** ”...my printer isn't all that reliable...” (Implies: Students' printers are functional enough to print the exam legibly).
- **D17:** ”How much time do we have between scanning and uploading after we are done with the exam?” (Implies: There is a clearly defined and reasonable time frame for submission after the exam's writing time).
- **D18:** ”I do not have great internet and I often have to drop out of calls or meetings...” (Implies: Occasional short internet disconnections during the Zoom call are acceptable and will not lead to accusations of cheating).

Explanation: The responses in ASR highlight further assumptions that were not explicitly stated in EM1 but are necessary for the system to work smoothly. These assumptions go beyond the initial list and delve into more specific scenarios and concerns raised by students.

e. Identify in ASR and label with ”Emote” as many portions of ASR as you can find (I found 4.) that describe an emotional impact on the students who have to take the exam.

- **Emote:** ”...the students will feel negative because they're being glared at by proctors in their personal spaces...”
- **Emote:** ”...stating that the exam will be held over Zoom is a form of peer pressure, which will cause some to follow through in ways that are uncomfortable to them...”
- **Emote:** ”The added stress of having to deal with technology during the exam makes it much more difficult.”
- **Emote:** ”I would be quite annoyed if I had to get one (a working printer) for this exam only...”

Explanation: These excerpts from ASR explicitly state or strongly imply negative feelings that students might experience due to the proposed exam format, such as stress, discomfort, or annoyance. This relates to the concept of user experience and satisfaction, which is an important aspect of requirements engineering, especially regarding NFRs like usability and acceptability.

f. Identify in ASR and label with "MR" as many portions of ASR as you can find (I found 1.) that describe scope-determineD (D) requirements that are clearly missing from the "R" identified in your answer to (a) above.

- **MR:** "How much time do we have between scanning and uploading after we are done with the exam?"

Explanation: This question points to a missing detail in the initial requirement (R). The original R stated that the exam would be "timed," but it did not specify the time allotted for the post-exam submission process. This is a D requirement because it specifies a detail *within* the already determined scope (a timed exam).

g. Identify in ASR and label with "NR" as many portions of ASR as you can find (I found 3.) that describe scope-determininG (G) requirements that are new and that could replace parts of, or augment, the "R" identified in your answer to (a) above.

- **NR:** "Would it not be more feasible to coordinate a timed assignment that we have a certain amount of hours to complete?"
- **NR:** "...if the exam could be online where proctor can look at the screen via zoom screen sharing..."
- **NR:** "...like to see an exam that is 'open-book', not monitored by Zoom, and is time-restricted, but to a longer period of time than a traditional exam."

Explanation: These suggestions propose fundamental changes to the nature of the exam, thus altering the project's scope. They are G requirements because they suggest changing *what* the system (the assessment) is, rather than just specifying details within an existing scope. For example, changing from a proctored exam to an open-book, time-restricted assignment is a significant scope change.

h. Instead of drawing a domain model of the world of the SE 463 final assessment as described by EM1, just list the name of all entities that you would put in the diagram. You may assume that there is a single entity named "S: Sys" that contains all system stuff that is hidden outside the interface part of the system. Thus, you should list only environment entities. Label with "E:" those entities that are in only the environment. Label with "SE:" those entities that are in the intersection of the environment and the system.

- **S: Sys**
- **E:** Student
- **E:** Printer
- **E:** Scanner or Camera
- **E:** Home or Exam Environment
- **E:** Proctor (TA, Dan, or Derek)
- **E:** AccessAbility Office
- **E:** Internet Service Provider
- **E:** Time Zone
- **SE:** Zoom
- **SE:** Computer or Device
- **SE:** Class Email or CrowdMark
- **SE:** Exam PDF
- **SE:** Pen or Pencil

- **SE:** Physical Exam Copy
- **SE:** Uploaded Exam

Explanation: This list identifies the key entities in the environment and at the interface. "E:" entities are purely external to the system, while "SE:" entities are shared between the system and the environment. The "S: Sys" represents the internal workings of the assessment delivery and collection system.

i. Identify in EM2 and label with "R'" (notice the prime) the portions, if any, that constitute the R' in "D', S' $\vdash$ R'". *R':* "...the final assessment will be open-book, timed from download, online (i.e., not on paper), and not proctored."

Explanation: This statement describes the revised requirement (R') for the final assessment. It reflects a significant change in scope compared to the original R, addressing many of the issues raised in the ASR.

j. Identify in EM2 and label with "S'" (notice the prime) the portions, if any, that constitute the S' in "D', S' $\vdash$ R'". *S':*

- "I will set a date range over which you will have 24 hours in which to do the download."

Explanation: This is the specification (S') corresponding to the new requirement (R'). It outlines how the "timed from download" aspect will be implemented, providing a 24-hour window for students to complete the assessment. It implicitly suggests that the assessment will be available for download, completed online, and submitted within that timeframe.

Question 2

a. Which of the domain assumptions "D1", "D2", ..., "Dn", "Dn+1", "Dn+2", are not solved by S" and are thus still required to hold? The following domain assumptions are still required to hold under S":

- **D2:** Each student has a reliable internet connection that will last the duration of the exam.
- **D3:** Each student has a device capable of connecting to the internet (to access the online assessment).
- **D5:** Each student has a quiet and suitable environment to take the exam.
- **D7:** Each student is in a time zone where the exam can be taken at a reasonable hour (though the 24-hour window provides more flexibility).
- **D9:** The platform used to deliver and collect the online assessment (e.g. learn) will be functional for exam submission.
- **D13:** All students are aware of the exam procedures and requirements.
- **D18:** Occasional internet disconnections are acceptable as long as the student can complete and submit within the time window.

Explanation: S" (the online, open-book, timed-from-download format) removes the need for printers, scanners, and real-time Zoom proctoring. However, it still relies on students having internet access, a suitable device, and a proper environment. D7 is mitigated by the 24 hour window. D9 is still relevant but the platform has changed. D13 is always relevant. D18 is relevant but the acceptable length of a disconnection is now much longer.

b. Thoroughly discuss the differences between S", specifying the final assessment that you are currently taking and S, specifying the final assessment as originally conceived. Your answer should take into account 1. information in your answers to Question 1, 2. how well S and S" mitigate or not the failure of parts of D to be true in the real world, and 3. the users', i.e., the students', likely feelings during the taking of the final assessments specified by S and S". Differences between S" and S:

1. **Information from Question 1:**
   - S" directly addresses the concerns raised in ASR about the feasibility and fairness of S. It eliminates the need for printers, real-time proctoring, and the strict time constraints of the original proposal.
   - S" is a response to the identified G requirements (NR) in ASR, such as the desire for an open-book exam and a longer time window.
   - S" implicitly acknowledges that many of the original domain assumptions (D) were likely to be false in the real world (e.g., universal access to printers, reliable internet for everyone).
2. **Mitigation of D Failure:**
   - **S:**
     - Fails to mitigate: Lack of reliable internet (D2), lack of a suitable device (D3), lack of a quiet environment (D5), and issues with the time zone of some students (D7).
     - Partially mitigates: Problems with printing (D1, D11, D15, D16) are addressed by removing the need for printing, but it introduces a new potential issue - the need for a stable internet connection during the entire exam period.
     - Does not mitigate: Students cheating (D12) is made easier in an unproctored environment.
   - **S":**
     - Mitigates: Issues related to printers (D1, D11, D15, D16) are completely eliminated. Concerns about scanning and uploading (D6, D17) are also removed. The stress associated with real-time proctoring (Emote) is gone. The 24-hour window addresses time zone issues (D7) to a great extent.
     - Partially mitigates: The need for a reliable internet connection (D2) is still present but less critical than under S, as short interruptions are acceptable.
     - Does not mitigate: The potential for cheating (D12) is likely higher in an open-book, unproctored setting.
3. **Students' Likely Feelings:**
   - **S:** Students would likely feel stressed, anxious, and potentially unfairly disadvantaged due to the strict requirements and reliance on technology and a perfect environment. The "Emote" portions of ASR highlight these negative feelings.
   - **S":** Students will likely feel less stressed and more in control due to the open-book nature, the 24-hour window, and the elimination of real-time proctoring. This format is more accommodating to individual circumstances and reduces the pressure associated with technical issues.

Conclusion: S" is a significant improvement over S in terms of feasibility, fairness, and student experience. It better aligns with the principles of good requirements engineering by considering the limitations of the real-world environment, addressing user concerns, and adapting the system to better meet the needs of its stakeholders. It demonstrates the iterative nature of requirements engineering, where feedback and analysis lead to a more refined and suitable solution. However, it still has limitations, particularly regarding the potential for cheating.

Question 3

a. Does the Deliverable 5 help you deliver a fully functioning implementation? Did Deliverable 5's authors' attempt to meet the acceptance criteria of the SE 463 staff lead to a document that helps address the problems that you believe you will face? If "yes", then how does it help you; if "no", then what is missing? *Answer will vary based on the specific Deliverable 5 being evaluated.*

a. Does the Deliverable 5 help you deliver a fully functioning implementation? Did Deliverable 5's authors' attempt to meet the acceptance criteria of the SE 463 staff lead to a document that helps address the problems that you believe you will face? If "yes", then how does it help you; if "no", then what is missing? *Answer will vary based on the specific Deliverable 5 being evaluated.*

Example Answer (Hypothetical):

As an implementer, Deliverable 5 provides a reasonably good starting point but does not fully equip us to deliver a fully functioning implementation without further effort.

Helpful Aspects:

- *The **domain model** gives us a good understanding of the key entities and their relationships, which is essential for designing the system's architecture.*
- *The **use cases and scenarios** provide valuable insights into how users will interact with the system and the expected behavior in different situations. This helps us define the system's functionality and guide the implementation of specific features.*
- *The **non-functional requirements (NFRs)** outline important quality attributes that we need to consider during development, such as performance, security, and usability.*
- *The **traceability matrix** helps ensure that all requirements are addressed in the design and implementation, reducing the risk of missing important features.*

Missing or Insufficient Aspects:

- ***Detailed design specifications:*** *Deliverable 5 lacks detailed design specifications, such as class diagrams, sequence diagrams, and specific algorithms. This leaves a lot of design decisions to the implementers, which can lead to inconsistencies and potential issues later on.*
- ***Edge cases and exception handling:*** *While scenarios cover some variations, they do not exhaustively address all edge cases and exception handling scenarios. We need to perform further analysis to identify and handle these situations during implementation.*
- ***Specific technology choices:*** *Deliverable 5 does not specify the technologies to be used for implementation (e.g., programming language, frameworks, databases). This leaves us with the task of evaluating and selecting appropriate technologies, which can be time-consuming.*
- ***User interface details:*** *While the document may provide a general overview of the UI, it lacks detailed mockups or wireframes, making it difficult to implement a consistent and user-friendly interface.*

Overall Assessment:

Deliverable 5's authors made a reasonable attempt to meet the acceptance criteria, providing a solid foundation for implementation. However, it falls short in providing the level of detail needed for a seamless transition to coding. As implementers, we need to invest significant effort in detailed design, edge case analysis, technology selection, and UI design to bridge the gap between the requirements specification and a fully functioning implementation.

In conclusion, Deliverable 5 serves as a valuable high-level guide but requires substantial elaboration and refinement during the implementation phase. It helps us understand what needs to be built but leaves many how questions unanswered.

b. Write an essay explaining what your group was doing that allowed your group to have picked a scope and its G requirements and to have worked out completely all the D requirements for these G requirements. By appeal to material learned in SE 463, explain why what the group was doing helped to find and document all these requirements. Consider the following issues in your answer.

- Even though your group did not need the gift, did your group's doing the deliverables tell your group something about its Capstone project's requirements that your group did not already know? If so, what did you learn from each deliverable?

- And, did your group have to change its Capstone project's direction or scope as a result of what your group learned in doing the deliverables? If so, what was the change?
- Was doing any of the deliverables a waste of time? If so, which ones and why?

Answer will vary based on the specific group's experience.

Example Answer (Hypothetical):

Our group had already established a well-defined process for requirements elicitation and analysis before starting the SE 463 deliverables. This allowed us to pick a scope, define our G requirements, and thoroughly work out the corresponding D requirements. Our approach was heavily influenced by the principles and techniques learned in SE 463.

Our Process:

1. **Iterative Requirements Elicitation:** *We began by conducting brainstorming sessions and interviews with stakeholders (including ourselves as potential users) to identify initial G requirements. We used techniques like use case analysis, as taught in SE 463, to understand the system's intended functionality from the user's perspective.*
2. **Domain Modeling:** *We developed a comprehensive domain model (class diagram) to represent the key concepts and their relationships in the problem domain. This helped us clarify our understanding of the system's environment and identify potential D requirements related to data constraints and business rules. This aligns with the SE 463 emphasis on domain modeling as a crucial step in requirements engineering.*
3. **Scenario Analysis:** *For each G requirement, we developed detailed scenarios, including typical scenarios, alternatives, and exceptions. This process, emphasized in SE 463, helped us uncover numerous D requirements related to specific user interactions, error handling, and edge cases. We used the concept of a "domain ignoramus" to challenge our assumptions and ensure we weren't overlooking any important details.*
4. **Non-Functional Requirements (NFRs):** *We identified and prioritized NFRs early on, using techniques like the quality grid discussed in SE 463. This helped us define constraints on the system's performance, security, usability, and other quality attributes.*
5. **Zave-Jackson Validation Formula (ZJVF):** *We used the ZJVF (D, S $\vdash$ R) as a guiding principle to ensure that our specification (S) and domain knowledge (D) were sufficient to satisfy the requirements (R). This iterative process helped us refine our requirements and identify any gaps or inconsistencies.*

Impact of Deliverables:

Even though our process was already aligned with SE 463 principles, completing the deliverables provided further insights:

- **Deliverable 1 (Abstract):** *Forced us to articulate the project's core value proposition concisely, which helped us refine our understanding of the target users and their needs. It also helped us identify what belongs in the interface (INTF).*
- **Deliverable 2 (Domain Model):** *Validated our existing domain model and highlighted a few minor inconsistencies that we subsequently corrected.*
- **Deliverable 3 (Use Cases and Scenarios):** *Provided an opportunity to revisit our scenarios and ensure they were comprehensive and aligned with the domain model. We identified a few missing scenarios related to edge cases.*
- **Deliverable 4 (Non-Functional Requirements):** *Reinforced the importance of NFRs and prompted us to further refine our fitness criteria for certain quality attributes.*
- **Deliverable 5 (SRS):** *Helped us organize our requirements in a structured manner and ensure that we had addressed all aspects of the IEEE SRS standard. It also forced us to think about traceability and how we would demonstrate that our implementation met the requirements.*

Changes to Project Direction:

As a result of doing the deliverables, we made one minor adjustment to our project scope. We initially planned to support a specific payment gateway, but through scenario analysis in Deliverable 3, we realized that supporting multiple payment options would significantly improve usability. This added a few D requirements related to payment integration but did not fundamentally alter the project's core functionality.

Waste of Time?:

None of the deliverables were a complete waste of time. While some deliverables reinforced existing knowledge or practices, they all provided value in terms of validation, refinement, or structured documentation. The most valuable deliverables were Deliverable 3 (Use Cases and Scenarios) and Deliverable 5 (SRS), as they directly contributed to the completeness and clarity of our requirements specification.

Conclusion:

Our group's proactive approach to requirements engineering, informed by SE 463 principles, allowed us to effectively manage the interplay between G and D requirements. The SE 463 deliverables, while not drastically changing our project's direction, provided valuable opportunities for validation, refinement, and structured documentation. This experience reinforced the importance of a systematic and iterative approach to requirements engineering in ensuring project success.

c. Did Dan succeed in giving the gift to your group? Regardless of your answer, write an essay answering this question. Consider the following issues in your answer.

- In doing the deliverables, what did your group do that it had not done before?
- What aspects of the development of your group's Capstone project were impacted by your group's doing the deliverables?
- How different do you think the current status your group's Capstone project would be today if your group had not done the deliverables?
- What did your group learn about its Capstone project's requirements from each deliverable?
- What was the most essential deliverable for the development of your group's Capstone project? and the least essential?
- Was doing any of the deliverables a waste of time? If so, which ones and why?

Answer will vary based on the specific group's experience.

Example Answer (Hypothetical):

Whether or not Dan succeeded in giving the 'gift of time' is debatable, but the SE 463 deliverables undoubtedly impacted our Capstone project, primarily by forcing a more structured and rigorous approach to requirements engineering than we might have otherwise taken.

What We Did Differently:

- **Formal Documentation:** *Before the SE 463 deliverables, our requirements were documented more informally, primarily through meeting notes and discussions. The deliverables forced us to create a formal SRS, adhering to the IEEE standard, which improved the clarity and organization of our requirements.*
- **Systematic Scenario Analysis:** *While we had considered user interactions, the emphasis on scenarios (typical, alternatives, and exceptions) in Deliverable 3 pushed us to perform a more systematic and thorough analysis of potential use cases, edge cases, and error handling.*
- **NFR Prioritization:** *We had identified NFRs, but Deliverable 4 prompted us to prioritize them using a quality grid, which helped us make more informed trade-off decisions during design and implementation.*
- **Traceability:** *The concept of traceability, emphasized throughout the deliverables, was not something we had explicitly considered before. Deliverable 5 encouraged us to think about how we would demonstrate that our implementation met the requirements.*

Impact on Capstone Project:

- **Improved Requirements Clarity:** *The deliverables forced us to articulate our requirements more precisely and comprehensively, reducing ambiguity and improving communication within the team.*
- **Early Identification of Issues:** *The structured approach, particularly scenario analysis, helped us identify potential problems and inconsistencies early on, saving us time and effort during later stages.*
- **Better Design Decisions:** *The focus on NFRs and their prioritization influenced our design choices, leading to a system that better met the desired quality attributes.*
- **Enhanced Traceability:** *While we didn't fully implement a traceability matrix, the deliverables made us more aware of the importance of demonstrating how our implementation satisfied the requirements.*

Current Status Without Deliverables:

If we had not done the deliverables, our Capstone project would likely be less well-defined from a requirements perspective. We might have encountered more issues during implementation due to ambiguities or overlooked requirements. Our design might not have adequately addressed NFRs, and we would have had a harder time demonstrating that our system met the intended goals. We may have had to go back and rewrite portions of our code had we not done the deliverables.

Lessons from Each Deliverable:

- **Deliverable 1 (Abstract):** *Taught us to concisely articulate the project's value proposition and target users.*
- **Deliverable 2 (Domain Model):** *Validated our understanding of the key concepts and their relationships.*
- **Deliverable 3 (Use Cases and Scenarios):** *Emphasized the importance of systematic scenario analysis for uncovering D requirements and edge cases.*
- **Deliverable 4 (Non-Functional Requirements):** *Reinforced the need to prioritize NFRs and define measurable fitness criteria.*
- **Deliverable 5 (SRS):** *Provided a structured framework for organizing and documenting our requirements, emphasizing traceability.*

Most and Least Essential:

- **Most Essential:** *Deliverable 3 (Use Cases and Scenarios) was the most essential, as it directly contributed to the completeness and detail of our requirements specification by forcing a thorough examination of user interactions and potential issues.*
- **Least Essential:** *Deliverable 2 (Domain Model) was arguably the least essential, as our team had already developed a solid understanding of the domain. However, it still served as a valuable validation exercise.*

Waste of Time?:

None of the deliverables were a complete waste of time. While some had a more significant impact than others, they all contributed to a more rigorous and structured approach to requirements engineering. The process, though demanding, ultimately improved the quality and clarity of our Capstone project's requirements.

Conclusion:

While the 'gift of time' might be subjective, the SE 463 deliverables provided a valuable framework and set of tools that enhanced our Capstone project's requirements engineering process. They forced us to be more systematic, thorough, and detail-oriented, leading to a better-defined and ultimately more successful project. The experience underscored the importance of the principles and techniques taught in SE 463 for real-world software development.

Question 4

a. What evidence from SE 463 materials would you remind M of to show her the flaw in her reasoning? I would remind M of the following evidence from SE 463 materials:

- **Pair Programming Benefits (Source: Lecture Notes/Textbook):** *Pair programming, as practiced by Bob and Alice, has been shown to improve code quality, reduce defects, and enhance knowledge sharing. Studies indicate that the time "lost" due to two people working on one task is often offset by the fewer errors and better design resulting from the collaborative effort. This aligns with the concept of "two heads are better than one," especially in complex problem-solving.*
- **Early Defect Detection (Source: Lecture Notes on D Requirements):** *Alice's detection of the potential division-by-zero error is a prime example of the benefits of early defect detection. Finding and fixing errors in the coding phase is significantly less expensive than dealing with them later in the development lifecycle (maintenance or, worse, after deployment). This relates to the high cost of fixing D requirement defects found late.*
- **Informal Code Review (Source: Lecture Notes/Textbook):** *Pair programming can be seen as a form of continuous, informal code review. Alice is effectively reviewing Bob's code as he writes it, leading to immediate feedback and error correction. This is more effective than traditional, formal code reviews that happen later in the process.*

Explanation: M's reasoning is flawed because she is only considering the immediate time spent on coding and not the potential long-term benefits of pair programming in terms of reduced defects, improved code quality, and knowledge transfer. SE 463 emphasizes the importance of early defect detection and the value of collaborative development practices, both of which are exemplified in this scenario.

b. Pair programming can be thought of as conducting concurrently, both programming and what other activity that was covered among the topics of SE 463? Pair programming can be thought of as conducting concurrently both programming and **informal but continuous code (or specification) review**.

Explanation: As one programmer (the driver) writes code, the other (the navigator) is actively reviewing the code, looking for errors, suggesting improvements, and ensuring that the code aligns with the requirements. This is a form of continuous inspection, a quality assurance technique discussed in SE 463.

c. If Alice had not found the defect and Bob had delivered the code as he was writing it before Alice's comment, the resulting defect would be due to the failure to take into account what kind of requirement? D or G? Explain your answer. The resulting defect would be due to the failure to take into account a **D (scope-determined) requirement**.

Explanation: The requirement to handle the case where "numberOfCourses" is zero is a detail *within* the already defined scope of calculating the GPA. It's not a new feature or a change to the overall goal of the module (which is to calculate the GPA), but rather a specific condition or constraint that needs to be handled within that scope. D requirements are about the specifics of "how" a system should function within a defined scope, while G requirements determine the scope itself. In this case, the scope is to calculate the GPA; the D requirement is to handle all possible inputs correctly, including the edge case of zero courses.

Question 5

a. 1. There's only enough for two. 2. There's enough only for two.

- **NS:** There's not enough for three.
- **Agreement:** NS agrees with sentence 2.

Explanation: Sentence 1 implies that there might be enough for more than two, but it's not certain. Sentence 2 states definitively that there is only enough for two and no more. The new sentence (NS) clarifies that there isn't enough for three, which aligns with the meaning of sentence 2.

b. 1. There's only enough for two. 3. There's enough for only two.

- **NS:** There is definitely enough for one.
- **Agreement:** NS agrees with sentence 1.

Explanation: Sentence 1 implies that there is at least enough for two, but could be more. Sentence 3 states that there is enough for two, and no more. The new sentence (NS) clarifies that there is at least enough for one, aligning with the meaning of sentence 1.

c. 1. There's only enough for two. 4. There's enough for two only.

- **NS:** There is definitely enough for one.
- **Agreement:** NS agrees with sentence 1.

Explanation: Sentence 1 implies that there is at least enough for two, but could be more. Sentence 4, similar to sentence 3, states that there is enough for two, and no more. The new sentence (NS) clarifies that there is at least enough for one, aligning with the meaning of sentence 1.

d. 2. There's enough only for two. 3. There's enough for only two.

- **Indistinguishable**

Explanation: Sentences 2 and 3 have the same meaning. They both state that there is enough for two people, and explicitly not enough for more than two. The placement of "only" does not change the core meaning in this case.

e. 2. There's enough only for two. 4. There's enough for two only.

- **Indistinguishable**

Explanation: Similar to the previous pair, sentences 2 and 4 are indistinguishable in meaning. They both convey that there's sufficient quantity for two and no more.

f. 3. There's enough for only two. 4. There's enough for two only.

- **Indistinguishable**

Explanation: Sentences 3 and 4 are also indistinguishable. They have the same meaning as sentences 2 and 4, emphasizing that the quantity is sufficient for two and no more.