SOFTWARE ENGINEERING CS 487

HOMEWORK 1

Name: Satyam Rajput (A20537375)
Email Id: srajput3@hawk.iit.edu

Homework – 1

Consider an automated (robot) tutor which helps a student with their assignments (but avoids doing the work for the student).

1. Clarify each of the following requirements to minimize ambiguity:

   a. “Robo-tutor must explain concepts effectively.”

   Considering the **User Requirement**
   Challenges

   We know that :

   • Ambiguity
   – Clarity is difficult to achieve
   – Especially since brevity is also desirable
   – Human language is different than user language is different than system language

   In order to **minimise ambiguity** and provide clarification regarding the requirement "Robo-tutor must explain concepts effectively," the following factors should be taken into account:

   Clarity: The robo-tutor should give explanations that are easy to grasp.Adjusting explanations to the student's comprehension level is important.

Engagement: Student engagement can be achieved by using a variety of teaching strategies.

Feedback: To assess the student's comprehension, provide them with feedback channels.

Acceptance to Standards: Verify that the justifications meet the criteria and standards of the educational community.
The requirement is made more explicit and less ambiguous by addressing these issues.

b. “Robo-tutor must recognize a student's mistake and
Explain the error without ‘giving away’ the answer.

Considering **Ambiguity**, **Amalgamation** and **Confusion**
as the **User Requirement Challenges and**
Capturing Requirements

Helpful Hints:
• Establish a standard format and adhere to it
• Use language consistently
• Highlight to distinguish key elements

Error Detection: The robot tutor should be able to spot errors made by the student and give a response without giving them the actual correct answer.

Error explanation: Rather than providing the solution right away, it should describe the error and guide the student on how to recognise and approach the solution.

Socratic Questioning: Utilise this method or another that is similar to assist the student in identifying the mistake on their own.

Guidance Approaches: Use instructions, hints or models to point students in the right direction in approaching the solution without providing the answer directly.

Advancement of Critical Thinking: Facilitate the development of critical thinking and problem-solving abilities by helping them evaluate their strategy and approaches for specific tasks and recognise the error.

c. “Robo-tutor must make learning fun.”

Considering **Interface Specification** and **Requirements Engineering** such as

• Process goal
– To create and maintain a system requirements document
• Process steps
– Feasibility study
– Requirements elicitation and analysis
– Requirements specification
– Requirements validation
• Corresponding outputs

for making the robo-tutor to follow the most **optimized agile method** for the best deliverability.

Engaging Content: To make learning fun, the robo-tutor should use interactive exercises, AR/VR, and multimedia to deliver content engagingly and interactively.

Personalisation: Learning can be made more engaging and relatable by adjusting the course material to the interests and preferences of the student.

Collaborative Learning: To promote a sense of teamwork and enjoyment in learning, include group challenges or collaborative activities.

Diversity in Teaching Methods: To keep learning engaging and exciting, incorporate a range of teaching techniques, including role-playing, storytelling, and practical exercises.

2. Document a test case for each of the clarified requirements (definitively determine correctness).

Test Case1: **Error Detection and Explanation**
Requirement: The robo-tutor must be able to recognise errors that students make in their assignments or exercises and provide an explanation of the error without giving away the solution.

Example of Test 1:
Input: A student inputs 5 x 4 = 22 as the solution to a multiplication problem.
Expected Result: The robo-tutor identifies the multiplication error and highlights it in the explanation, pointing the student in the direction of the right answer without saying "5 x 4 = 20" out loud.

Test Case 2: **Engaging Content Delivery**
The robo-tutor needs to present material in an interesting and interactive way.

Example of Test 2:

Input: Using an interactive timeline and multimedia components like videos and quizzes, the robo-tutor instructs history.

Expected Result: the students demonstrate enhanced retention of the historical material, actively engage with the subject matter, and take part in the quizzes.

Test Case 3: **Adequate Support and Reinforcement**

Requirement: To inspire and involve the student, use positive reinforcement techniques like motivation, compliments, and encouragement.

Example of Test:

Input: For students who consistently score well on assignments or quizzes, the robo-tutor awards virtual badges or stars.

Expected Result: Students display increased motivation, engagement and involvement with the course content as they strive to accrue more virtual incentives.

Test Case 4: **Collaborative Learning**
Requirement: To promote teamwork and enjoyment in learning, include cooperative exercises or group challenges.

Example of Test:
Input: Students are given a group project by the robo-tutor that requires cooperation to solve a challenging issue.
Expected Result: the students work together constructively, exchanging ideas and knowledge, resulting in a fun and
insightful learning experience.

3. Propose an H-C-I “protocol” to allow a student and a robo-tutor to effectively interact.

For an effective H-C-I protocol we need to consider the following:

User Requirement Challenges
Requirements Engineering
Interface Specification
Ethnography
Requirements Discovery

The robo-tutor can effectively interact with the student, offer specific and flexible learning experiences, and promote a positive and encouraging learning environment by putting this H-C-I protocol into practice.

Introduction

Explanation of Role:The robo-tutor introduces itself and sets the context for the interaction by outlining its role.

User Input

- Clear Input Options for the User: The robo-tutor allows both text and voice input and offers the student to input queries, responses, or requests, accommodating both text and voice input.
- Encourage Student Input: By asking queries and providing feedback on the subject matter, the robo-tutor motivates the learner to participate.

Response Mechanism

- Clarity in Responses: The robo-tutor makes sure that answers are brief, understandable, and appropriate for the student's level of comprehension.
- Adaptive Feedback: By addressing mistakes, progress and offering direction, the robo-tutor offers adaptive feedback based on the student's progress.

Interactive Learning Interface

- Engaging Content Delivery: To improve understanding and engagement, the robo-tutor uses interactive exercises, multimedia, and real-world examples when delivering content.

- Prompting Interaction: In order to strengthen learning, the robo-tutor encourages the student to take part in discussions, tests, and problem-solving exercises.

Error Handling

- Constructive Error Feedback: Rather than just giving the student the answer, the robo-tutor points out mistakes made by the student and offers constructive feedback.

- Encouraging Persistence: By promoting a growth mindset, the robo-tutor helps the student keep going in spite of challenges and errors.

Monitoring Progress

- Regular updates on the student's progress are given by the robo-tutor, who points out both the student's accomplishments and areas for development.
- Goal Setting: The robo-tutor establishes learning objectives and monitors the student's progress towards achieving them.

Feedback Mechanism:

-Student Input: The robo-tutor actively take student input concerning the quality of the interaction and the learning process.

- Integration of Feedback: To enhance its functionality and customise the learning process to each student's preferences, the robo-tutor incorporates feedback from students.