

KillTest

高品質 より良いサービス



練習問題

<https://www.killtest.jp>

1年間無料アップデート

Exam : ACA100

Title : Appian Certified Analyst

Version : DEMO

1. The order management team receives orders via different channels, including email. Emails are accompanied by attachments containing the details of the order information. The team reviews the emails and attachments for clarity. Then, they extract the order specifics to create a customer record in their Customer Relationship Management (CRM) platform and an order in their Order Management System (OMS).

What is the most effective form of automation to help increase fulfillment efficiency in this process?

- A. Review the emails and attachments using trained artificial intelligence to extract order data information
- B. Provide a task after the order email was received asking the team to check the order status and select the next step option
- C. Prompt the order management team members with a task to create an order in the CRM platform
- D. Have an auto-reply "Thank You" email sent back thanking the customer for sending in their order request

Answer: A

Explanation:

The most effective form of automation to increase fulfillment efficiency in this process is using trained artificial intelligence (AI) to extract order data information from the emails and attachments. AI can be trained to recognize and extract key details such as customer names, order numbers, and product information directly from the content of the emails and attachments. This automation significantly reduces the manual effort required to review and input data, thereby speeding up the order processing time and reducing the chance of errors.

Reference: Appian Documentation - AI and Automation Capabilities

Note: The other options (B, C, D) provide minimal automation and would still require significant manual intervention, which does not effectively enhance fulfillment efficiency.

2. A partially created user story for an approval task has been added to the Product Backlog and you have been asked to review it.

So far, the user story includes acceptance criteria stating:

"Verify that the user can select approve or deny."

"Verify that the user can enter an approval comment."

What are two examples of additional content that need to be added to the user story before it can be considered complete?

- A. Condition upon selecting either "deny" or "approve"
- B. User story description
- C. UI mockup
- D. Truth table

Answer: A, B

Explanation:

User Story Description (B): A complete user story should include a detailed description that outlines the context, goals, and specific actions the user can perform. This helps in setting a clear understanding of what needs to be developed and why. The description provides the foundation for the user story and ensures that all stakeholders have a shared understanding of the requirement.

Reference: Appian Documentation - Writing User Stories

Condition upon Selecting Either "Deny" or "Approve" (A): To ensure that the user story is complete, you

must specify what happens when the user selects "deny" or "approve." This could include conditions such as sending notifications, triggering workflows, or updating statuses. Clearly defining these conditions ensures that the development team knows exactly what to implement and how the system should behave in different scenarios.

Reference: Appian Documentation - Acceptance Criteria Examples

3. During the "initiate" phase, what are two key benefits of successful requirements gathering?

- A. Adherence to the business stakeholders' goals
- B. Less rework after the "test" phase
- C. Fewer test cases required
- D. Better performance of the application

Answer: A, B

Explanation:

Reference: Appian Documentation - Requirements Gathering Best Practices

Less Rework After the "Test" Phase (B): Thorough and accurate requirements gathering reduces the likelihood of discovering significant issues during the testing phase. When the requirements are well-defined and understood from the start, the development team is more likely to build the application correctly the first time, resulting in less rework after testing.

Reference: Appian Documentation - Reducing Rework through Proper Planning Note: Fewer test cases required (C) and better performance of the application (D) are not direct benefits of successful requirements gathering. While better requirements can contribute to a more streamlined testing process and potentially better performance, these are not primary benefits.

4. As an Appian Analyst, which two activities are you responsible for during the "Initiate" phase of a project?

- A. Build the deployment pipeline
- B. Conduct sprint reviews
- C. Define personas
- D. Plan a requirement gathering approach

Answer: C, D

Explanation:

Reference: Appian Documentation - User Personas

Plan a Requirement Gathering Approach (D): Planning the requirement gathering approach is another crucial activity in the "Initiate" phase. This step involves determining how requirements will be collected, documented, and validated throughout the project. A well-defined approach ensures that all necessary information is captured early, setting a strong foundation for the development process.

Reference: Appian Documentation - Requirements Gathering

Note: Building the deployment pipeline (A) and conducting sprint reviews (B) are activities typically associated with later phases in the project lifecycle, such as development and delivery.

5. Review this image:

AND, OR, XOR

What is this group of BPMN symbols called?

- A. Activities

B. Gateways

C. Events

D. Flows

Answer: B

Explanation:

In BPMN (Business Process Model and Notation), the symbols AND, OR, XOR are categorized as Gateways. Gateways control the flow of processes by determining which path to follow based on certain conditions. They are essential in managing the decision points within a process, directing the flow based on the logic defined in the process model.

Reference: Appian Documentation - BPMN Gateways