Gunde Blessy Pravalika/ Email: blessy.solomondavid@gmail.com

Section A — Agile/Scrum Fundamentals

1. **In Scrum, who is primarily accountable for maximizing the value of the product?**

    - A) Scrum Master
    - B) Product Owner
    - C) Development Team
    - D) Stakeholders

    Answer: Option B

2. **Which ceremony is most appropriate to raise test environment blockers for the upcoming sprint?**

    - A) Daily Scrum
    - B) Sprint Review
    - C) Sprint Planning
    - D) Retrospective

    Answer: Option C

3. **You discover ambiguous acceptance criteria during Backlog Refinement. What two actions would you take before the next Sprint Planning to ensure testability?**

    Action 1: I will communicate with the Stakeholders, Business Analyst and Product Owners, to discuss the unclear or incomplete requirement. I will review that I have gathered all the required data and confirm with them about the final requirement

    Action 2: After gathering all the necessary data or required information I will validate that all the inputs and data are clearly included in the acceptance criteria. Before testing I will review the acceptance criteria to check whether the information provided is clear

4. **As a tester, how would you contribute in Sprint Review and Retrospective to improve product quality and team process? Provide two concrete examples for each ceremony.**

    Sprint Review: As a tester, I help to improve product quality by sharing test results, highlighting risks, and giving feedback on completed stories. I also provided the demos

to the Product Owners which helped in getting the feedback early which improved the product quality.

Example1: I usually give a quick update on test coverage and open defects, and if stakeholders raise questions or feedback, I capture them as action items or backlog updates. I shared that most test cases passed but a couple of minor bugs were still open, so stakeholders were aware of the current quality.

Example 2: While the team was demoing a feature, I noticed something from a user point of view like a confusing message or missing validation. I pointed it out during the review so we could improve it based on feedback.

Retrospective: I contribute by analyzing defect patterns, highlighting what slowed testing, sharing ideas to improve collaboration or automation, and suggesting small, actionable process improvements that help the team deliver higher quality in the next sprint

Example 1*:* In a retrospective, I mentioned that some requirements were unclear, which caused delay in the testing. I suggested having earlier clarification sessions or acceptance criteria reviews, and the team agreed to involve QA earlier."

Example 2: I noticed regression testing was taking too long each sprint, so in the retrospective I suggested automating the most repetitive test cases. This helped reduce manual effort in later sprints.

Section B — Manual Testing Basics

5. **Which option best defines exploratory testing?**
    - A) Executing pre-written automated scripts
    - B) Unscripted, learning-driven testing guided by charters
    - C) Performance load testing
    - D) Unit testing by developers

    Answer: Option B

6. **List the essential elements that a well-written test case should include. Briefly justify why each element matters.**

The essential elements that are included to write the well written test cases are below Test Case number, Story ID, Sprint, Title, Precondition, Test Steps, Expected Result, Status, Comments

- **Test Case number:** Indicates the test case number. Based on the number we can know how many test cases have been created for that story
- **Story ID:** Provides the story number which will be helpful to identify for which story this id belongs to.
- **Sprint**: Specifies which sprint the story belongs to
- **Title:** Specifies the title of the story so whenever we open, we can clearly see to which story this belongs
- **Precondition:** Specifies the setup that needs to be done before we start testing
- **Test Steps**: Describes step by step to execute the test accurately and consistently
- **Expected Result:** Describes the exact result we expect and the correct outcome once the testing is done
- **Actual Result**: Mentions what we encountered during testing
- **Status**: Provides the status of the testing whether the test is passed or failed based on the Expected result.
- **Comments**: Captures the observations or clarification that we came across during testing

7. **Explain the difference between severity and priority with a brief example for each (one or two sentences per example).**

    - **Severity**: Severity indicates how serious is the defect and how the defect is affecting the functionality
    - **Example**: If a User wants to transfer the amount to other user account and the amount doesn't reache the correct account then the functionality is breaking and leading to incorrect results
    - **Priority**: Priority indicates how urgently a defect should be fixed based on the business urgency
    - **Example**: In the bank application if there is a typo mistake in the title page it may not affect the functionality but it's breaking the customer trust and bank professionalism

8. **Given a feature with acceptance criteria, describe how you select a minimal set of test cases to achieve risk-based coverage. Mention at least three factors that influence your selection.**

    When I receive a feature then below are the factors that I follow

- First, I will go through the Acceptance criteria, and I will prioritise the test cases based on the Business impact and core functionality, I make sure these are tested first
- Second, "Whenever a feature involves complex logic or integrations, I make sure to test and those parts more carefully since they usually have a higher chance of issues.
- Third, I look at past similar features to see where bugs usually happen, so I make sure those areas are covered and included in the test issues or cases.

Section C — Azure DevOps Basics

9. **In Azure DevOps Boards, which work item type typically holds acceptance criteria for a user-facing capability?**

    - A) Bug
    - B) Task
    - C) User Story
    - D) Epic

    Answer: Option C

10. **In an Azure DevOps Bug work item, which field/area is used to link it to the work it impacts (e.g., a User Story)?**

    - A) Area Path
    - B) Iteration Path
    - C) Related Work (linking)
    - D) Assigned To

    Answer: Option C

11. **Outline the steps to transition a User Story from New → Active → Resolved → Closed while ensuring testing gates are respected. Include at least one checkpoint for each transition.**

    a) New → Active: this is the Stage where the Story is ready, and developers are assigned to the story

    Checkpoint: The acceptance criteria should be clear, and all the required data should be provided

b) Active → Resolved In this Stage the developers has completed the work and deployed the code to QA environment and assigned to QA team

Checkpoint: Acceptance criteria is met, Code review is done, basic testing is done, code deployment is done, Build is successful without any errors, Evidence attached, comments added

c) Resolved →Closed in this stage the QA team will verify and confirms if it's working as expected according to the requirement then they will make it as Resolved and Product Owners will verify and approves and closes the User story

Checkpoint Fix should work as expected, Regression should be done, Evidence should be added, Comments should be added, Approval from Product Owners

12. **Describe how you would use Queries or Dashboards in Azure DevOps to track defect trends across a sprint. Mention at least two metrics or visualizations you would create.**

- I Use queries to filter the Bugs of the current Sprint and dashboard to monitor the defects. This helped to understand how defects are changing during the sprint
- The metrics or visualization I create are Trend Chart and Pie Chart
- Trend Chart is created to track the progress through the sprint and Pie Chart shows which areas are risky and which need to be fixed first

---

Section D — SharePoint & Microsoft 365

13. **Name one common reason permissions behave unexpectedly in SharePoint Online.**

    - A) Versioning turned off
    - B) Unique permissions have broken inheritance on a sub-site/list
    - C) Empty recycle bin
    - D) Co-authoring disabled

    Answer: Option B

14. **You're testing a custom SharePoint solution (web parts + lists/libraries). Provide three functional checks and two non-functional checks you would perform that are specific to the SharePoint/M365 context.**

The functional check that I perform are Permission, Document Library Action and Load of Custom Webparts and the non-functional check are Performance and Security

Functional Checks for SharePoint/M365

- **Permission:** First, I will check whether user is having permission to view, add, delete or edit the items
- **Document Library action:** I will validate whether we can create, edit, upload or delete the document and its data is saved correctly or not
- **Load of Custom Webparts:** I ensure that custom webparts are loading the data correctly

Non-Functional Checks for SharePoint

- **Security:** I will ensure that it's following the security standard and doesn't share the sensitive information
- **Performance:** I will verify if pages load properly when accessed by multiple users simultaneously.

Section E — AI Tooling Mindset

15. **Give a concrete example of how you would use an AI assistant to help with testing (e.g., drafting test cases from acceptance criteria or generating exploratory charters). Include one prompt safety consideration and one human verification step.**

    - I use AI assistant to draft test cases from acceptance criteria because it will generate the detailed test cases, saves time and provides the expected result and sometimes I use AI for generating exploratory charters because it provides what to test and what to focus on
    - prompt safety consideration: We should not have to share company's confidentiality data like password or sensitive company information or any link that opens secure data of the company
    - human verification step: Need to review that the data which is provided by AI is correct and satisfying the requirement or not

16. **You have noisy, duplicate bug reports. Describe how you could use AI to cluster or summarize them, and what human guardrails you would apply before acting on the output.**

In my experience, when I encounter noisy or duplicate bug reports, I use AI to generate summaries for each cluster, highlighting key details such as steps to reproduce, affected components, and error messages. Since AI is not always perfect, I review the clusters to confirm that reports truly represent the same bug. I also cross-check screenshots and logs for accuracy. For additional clarity, I discuss any ambiguous cases with the team and perform further verification before taking action on the bug.

Section F — Practical Task 1: Bug Report (SharePoint Scenario)

Scenario: In the **Documents** library of a SharePoint site, clicking **"Upload"** sometimes a spinner that never finishes; the file is not added. This occurs predominantly for files **>** shows **50MB** when using **Microsoft Edge v120+**. Smaller files upload fine. No error toast is displayed.

17. Write a **high-quality bug report** for the scenario above, including:

- **Clear title:** Uploading large Files (file size>50mb) shows Spinner in Microsoft Edge v120+
- **Environment details (tenant/site, browser/version, OS, network conditions**) SharePoint, Edge v120, Windows, WIFI / VPN
- **Preconditions (e.g., permissions, feature flags, test data):** User need to have the permission in the SharePoint document library for uploading the large files. File size should be more than 50MB. The browser should be Edge v120+
- **Numbered steps to reproduce**
  Step 1: Navigate and open the SharePoint
  Step2: Navigate to Document Library
  Step3: Click on Upload the File button
  Step4: Select the file which is more than 50MB
  Step 5: Click on the Open button to upload
  Step 6: Verify that file should not be uploaded, and error toast should not be displayed
- **Expected** vs **Actual** results

  Expected Result: The Selected file (more than 50MB) should be uploaded successfully to the Document Library and successful message, or notification should be displayed. Uploaded File should be appeared in the library list when user navigates to the Document Library

Actual: When User trying to upload the file with more 50MB using the Edge v120 then the spinner is running continuously, and the file is not adding to the library. No error toast is displayed

- **Frequency (e.g., % or "intermittent" with notes)**

  It occurs 70% of time when we upload the files >50MB

  This is an intermittent issue which occurs only when the file is more than 50MB in size

- **Notes**: This happens only when the user trying to upload the file which is more than 50MB on Edge v120 and able to upload the smaller file
- **Attachments** (**logs, HAR, console output**, screenshots)

  We are providing the evidence or required data to the developers or team to reproduce the issue

  Screenshots: In this it shows the Spinner running continuously when we upload the file>50MB

- **Severity** and **Priority** with rationale

  Severity (High): User is unable to upload the large file, and this is blocking the critical work especially if user want to upload the large file

  Priority (P1: urgent): This bug needs to be fixed asap So user can upload the large file and continue the work

- **At least two suggested diagnostic clues (e.g., upload chunking, throttling, conditional access policies)**

  First the issue might be due to large file as its not accepting the large files and the process is getting stuck

  Second the issue might be due to low performance

- **Placeholder for Related Work link (e.g., associated User Story or Task)**

  This is a space where we add the related items to the Task

Section G — Practical Task 2: Test Cases (Custom Web Part)

Mini-feature: A custom **News** web part displays the latest **10** items from a list. It supports filtering by **Category** (single-choice field). **Pagination** appears only if **>10** items match the filter.

18. Author **three test cases** for the mini-feature:
    - **Default view:** No filter → shows 10 most recent items; pagination appears if items >10.
    - **Category filter:** Select "Announcements" → only items with Category = Announcements; pagination rule respected.
    - **Boundary:** Exactly 10 matching items → verify pagination is hidden; sorting by **Created (desc)** is correct.

For each test case, include:

- Preconditions
- Steps
- Test data
- Expected results
- Postconditions

- **Test Case 1: Default view: No filter → shows 10 most recent items; pagination appears if items >10.**

  Preconditions:

  The New Web part contains more than10 items in the list

  No Category filter is applied

  Steps:

  Step1: Navigate and Open the Page containing Web Part

  Step2: Verify that no filter is applied

  Step3: Verify whether Web part displays the most recent items

  Step 4: Verify that pagination button appears if the items are more than 10

  Step 5: Click the pagination and able to navigate to the next page

  Test Data:

  - File size= 13
  - Created data should be unique

Expected results

- The Web part display the 10 Recent files
- Pagination appears when the files are more than 10
- After clicking on pagination button remaining set of items are displayed on the next page

Postconditions

- The New Web Part and other items remains unchanged
- The Pagination button appears as the total items are more than 10
- Filters are not applied

- **Test Case2 : Category filter: Select "Announcements" → only items with Category = Announcements; pagination rule respected.**

  Precondition:

  - The New Web Part need to contain files with multiple categories including Announcement
  - No Category filter is applied prior
  - Need to have 10 files with Announcement item

  Steps:

  Step1: Navigate and Open the Page containing Web Part

  Step2: Verify that no filter is applied

  Step3: Upload the Files with multiple categories

  Step4: Select the filter as Announcement

  Step5: Verify that items with Announcement category are displayed

  Step 6: Verify that pagination button appears if the items are more than 10

  Step 7: Click the pagination and able to navigate to the next page

  Step 8: Verify that remaining items with other categories are displayed on the next page

  Test Data:

  - File size= 15
  - Need to contain File with Announcement Category = 10

- Other category file should be 5
- Created data should be unique

Expected Result:

- Items with Announcement category are displayed
- First page shows only the 10 most recent Announcements items
- Pagination appears because file is containing more than 10 items
- Clicking the Pagination button the Next page displays remaining items including announcement item
- After clicking on pagination button remaining set of items are displayed on the next page

Postconditions

- The New Web Part and other items remains unchanged
- Filters are applied and the system can reset for the next test
- The Pagination button appears as the total items are more than 10

- **Testcase 3: Boundary: Exactly 10 matching items → verify pagination is hidden; sorting by Created (desc) is correct**

Precondition:

- The New Web Part need to contain exactly 10 matching items
- No Category filter is applied prior
- Need to have 10 items with unique created data

Steps:

Step1: Navigate and Open the Page containing Web Part

Step2: Apply filter as per the requirement

Step3: Verify that the Web part displays the 10 matching items

Step 4: Check that the pagination button is hidden because it has 10 items

Step 5: Verify that the Items are sorted in descending order based on the created data

Test Data:

- File size= 10
- Category can be anything
- Created date should be unique because based on this the item will be sorted ascendingly

Expected results

- The Web part displays all the 10 Items
- Pagination is hidden because the items are ≤10
- Items are correctly sorted by Created date descending

Postconditions

- The New Web Part and other items remains unchanged
- The Pagination button is hidden
- Items are sorted according to the Created date

Section H — Optional Self-Assessment (No Points)

19. Rate your comfort (1–5) in: manual testing, Azure DevOps, SharePoint, AI tools.

- Manual testing – 4/5
- Azure DevOps – 3.5/5
- SharePoint – 3/5
- AI tools – 3/5

20. Describe one time you improved test documentation quality during a sprint. What changed and what was the outcome?

- During one sprint I have noticed that the Test documentation was incomplete, and few columns were missing because of this the test cases were incomplete, and few scenarios were missing and was unclear
- This led to misunderstanding and delay in the testing
- I have introduced new test case template which improved the test case format and added clear steps
- Due to new change, they were under stable to everyone who ever referred the test cases, better communication, faster testing and less confusion