**Study Material: Understanding Defect Reports**

**1. Introduction to Defect Reports**

A **defect report**, also known as a **bug report**, is a document that records the occurrence, nature, and status of a defect in a software product. When software does not function as expected, a defect report is written to notify developers about the issue so they can fix it.

**2. Test Process Overview**

A software testing process consists of three main steps:

| **Step** | **Description** |
| --- | --- |
| **Test Planning** | Creating a plan for the testing process. |
| **Test Design** | Writing test cases and adding them to test suites for execution. |
| **Test Execution** | Running test cases and comparing expected vs. actual results. |

If there is a mismatch between the expected and actual results, a defect report is created and assigned to a developer for resolution.

**3. What is a Defect?**

A **defect**, also known as a **bug** or **fault**, is an imperfection in software where it does not meet its expected requirements or specifications.

**Example of a Defect**

* **Expected Result**: A login page should display an error message if a user leaves the password field empty.
* **Actual Result**: No error message appears when the password field is left empty.
* **Conclusion**: Since the actual result does not match the expected result, a defect report must be created.

**4. Components of a Defect Report**

A defect report includes several key fields to describe the issue effectively:

**4.1 Bug Report Title**

The title should clearly describe the problem. The best format is:

[Section Name] → [Short Description]

**Example:**

Login Page → No error message appears when password field is empty.

**4.2 Steps to Reproduce**

These steps describe how to replicate the defect. Each step must be clear and precise.

| **Step** | **Action** |
| --- | --- |
| 1 | Open www.example.com/login. |
| 2 | Enter a valid username but leave the password field empty. |
| 3 | Click on the "Login" button. |
| 4 | Observe that no error message appears. |

**4.3 Expected Result**

What should happen if the software is functioning correctly?

An error message should appear: "Password field cannot be empty."

**4.4 Actual Result**

What actually happens when the steps are executed?

No error message appears, and the login process does not proceed.

**4.5 Environment Details**

Specifying the environment helps developers reproduce and fix the defect.

| **Device** | **OS Version** | **Browser** |
| --- | --- | --- |
| Laptop | Windows 10 | Chrome v98 |
| Mobile | Android 11 | Safari |

**4.6 Reproducibility Rate**

Describes how often the bug occurs.

Reproducibility: 5/5 (Happens every time the steps are followed.)

**4.7 Attachments (Screenshots/Videos)**

* **Static Defects**: Use screenshots.
* **Dynamic Defects** (e.g., animations, loading delays): Use video recordings.
* Ensure the full screen is captured, including the URL if applicable.

**5. Bug Life Cycle**

Once a defect is reported, it goes through various stages:

1. **New** – A tester reports a bug.
2. **Assigned** – The bug is assigned to a developer.
3. **In Progress** – The developer works on fixing it.
4. **Fixed** – The developer resolves the issue.
5. **Retest** – The tester verifies if the fix works.
6. **Closed** – If the fix is correct, the bug is closed.
7. **Reopened** – If the fix is incorrect, the bug is reopened.

**Diagram: Bug Life Cycle**

New → Assigned → In Progress → Fixed → Retest → (Closed / Reopened)

**6. Types of Defects**

Different types of defects may occur in software:

| **Type of Defect** | **Description** |
| --- | --- |
| **Functional Defect** | A feature does not work as expected. |
| **Performance Defect** | The system is slow or unresponsive. |
| **UI Defect** | Issues with design, layout, or responsiveness. |
| **Security Defect** | Vulnerabilities that compromise security. |
| **Compatibility Defect** | Issues on specific devices or browsers. |

**7. Conclusion**

A well-written defect report ensures effective communication between testers and developers, leading to faster issue resolution and better software quality. Following a structured approach, including clear titles, detailed steps, and proper documentation, helps streamline the defect management process.

**Introduction to Defect Reporting and Bug Tracking**

**What is a Defect?**

A defect, also known as a bug, is an error, flaw, or inconsistency in a software application that causes it to behave unexpectedly. Defects can impact functionality, user interface, content, performance, and overall user experience.

**Types of Defects**

**1. Functional Defects**

Functional defects occur when a specific feature of an application does not work as intended.

**Example:** If the "Forgot Password" button on a login page does not redirect the user to the password reset page, it is a functional defect.

**Diagram:**

Expected Behavior:

Click "Forgot Password" -> Redirect to Reset Page

Actual Behavior:

Click "Forgot Password" -> Nothing Happens (Defect)

**2. UI (User Interface) or Visual Defects**

UI defects affect the appearance of an application, including misalignment of elements, incorrect font sizes, or improper spacing.

**Example:** A button labeled "Add to Cart" appears misaligned on an e-commerce website.

| **Defect Type** | **Example** |
| --- | --- |
| Misalignment | Button text not centered |
| Overlapping Text | Text covering another UI element |
| Incorrect Colors | A button color does not match the theme |

**3. Content Defects**

Content defects include spelling errors, missing translations, or duplicate text in an application.

**Example:** If an Arabic version of an application displays an English word instead of its Arabic translation, it is a content defect.

**4. Performance Defects**

Performance defects occur when an application is slow or does not respond as expected.

**Example:** If a webpage takes too long to load or displays a "Loading..." message indefinitely, it is a performance defect.

**Diagram:**

Expected Load Time: < 3 seconds

Actual Load Time: > 10 seconds (Defect)

**5. Suggestions**

Suggestions are improvements that testers recommend for better usability but are not actual defects.

**Example:** A tester suggests increasing the font size of placeholder text in an input field for better readability.

**Defect Report Format**

A defect report provides structured information about an issue to help developers fix it efficiently.

**Example Defect Report**

| **Field** | **Details** |
| --- | --- |
| **Title** | Forgot Password button is not working |
| **Steps to Reproduce** | 1. Open the login page 2. Click "Forgot Password" button |
| **Expected Result** | User is redirected to the password reset page |
| **Actual Result** | Nothing happens when clicking the button |
| **Environment** | Windows 10, Chrome 108 |
| **Priority** | High |
| **Defect Type** | Functional Defect |
| **Screenshot** | (Attach Screenshot) |

**Defect Tracking Tools**

Testers use specialized tools to report and track defects efficiently. Some popular defect tracking tools include:

* **Jira**
* **Trello**
* **Bugzilla**
* **Redmine**

These tools help teams collaborate, prioritize, and resolve issues systematically.

**Conclusion**

Understanding defect types and how to report them effectively is essential in software testing. Well-documented defect reports help developers identify and resolve issues faster, improving the software's quality and user experience.

**Study Material: Taking and Annotating Screenshots for Defect Reporting**

**Introduction**

When reporting defects in software testing, providing a screenshot of the issue helps developers quickly understand the problem. This study material explains simple ways to take and annotate screenshots for defect reports.

**Methods for Taking Screenshots**

There are multiple ways to capture screenshots on a Windows computer. The two most common methods are:

**1. Using the Snipping Tool**

The **Snipping Tool** is a built-in Windows application that allows users to capture a specific area of the screen.

**Steps to use the Snipping Tool:**

1. Open the **Snipping Tool** from the Start menu.
2. Click **New** to start a new screenshot.
3. Drag and select the area you want to capture.
4. You can use the **Pen** or **Highlighter** tools to annotate the image.
5. Click **Save** to store the image as a .JPG or .PNG file.

📌 *Tip: Pin the Snipping Tool to your taskbar for quick access.*

**2. Using Print Screen (PrtScr) and Paint**

The **Print Screen (PrtScr)** button on your keyboard captures the entire screen, which can then be pasted into an image editor like Paint.

**Steps to use Print Screen and Paint:**

1. Press the **PrtScr** button on your keyboard.
2. Open **Paint** (a built-in Windows program).
3. Press **Ctrl + V** or click **Paste** to insert the screenshot.
4. Use the tools in Paint to edit or highlight the issue:
   * Select **Rectangle Tool** → Choose **Red** color → Draw around the defect.
   * Use the **Text Tool (A)** to add a description.
   * Use **Crop Tool** to remove unnecessary parts of the image.
5. Click **Save As** and choose .JPG or .PNG format.

📌 *Tip: Pin Paint to your taskbar for easy access.*

**Example of a Defect Screenshot Annotation**

Here’s an example of how a properly annotated screenshot should look:

| **Screenshot** | **Explanation** |
| --- | --- |
|  | The **Add to Cart** buttons are not aligned properly. A red rectangle highlights the issue, and a text note explains the problem. |

**Attaching Screenshots to Defect Reports**

Once you have captured and annotated the screenshot, attach it to your defect report along with the following details:

* **Defect Title**: A short, descriptive title (e.g., *“Login -> Forgot Password button not working”*).
* **Steps to Reproduce**: Clear instructions on how to replicate the issue.
* **Expected Result**: What should happen.
* **Actual Result**: What actually happens.
* **Priority Level**: How urgent the issue is (High/Medium/Low).
* **Environment**: Where the defect occurs (e.g., Windows 10, Chrome Browser).

**Conclusion**

Providing clear and well-annotated screenshots significantly improves the efficiency of defect reporting. Using the **Snipping Tool** or **Print Screen with Paint** ensures that developers can quickly identify and resolve issues.

**Study Material: Recording and Attaching Videos for Defect Reporting**

**1. Introduction**

When reporting a software defect, sometimes a screenshot is not enough to explain the issue. Recording a video helps developers understand the problem more clearly. This document explains how to record a video of a defect using a simple, free tool called **Screen-O-Matic**.

**2. Why Use Video for Defect Reporting?**

* **Clear Explanation**: Videos show step-by-step actions leading to the issue.
* **Better Communication**: Developers can see exactly what happens.
* **Time-Saving**: No need for long written explanations.

**3. Available Tools for Screen Recording**

| **Tool Name** | **Cost** | **Features** |
| --- | --- | --- |
| Camtasia | Paid | Advanced editing, high memory usage |
| Adobe Premiere | Paid | Professional editing, high memory usage |
| **Screen-O-Matic** | **Free** | Browser-based, easy to use, low memory |

Among these, **Screen-O-Matic** is recommended because it is free and does not require installation.

**4. How to Use Screen-O-Matic**

**Step 1: Open Screen-O-Matic**

1. Open a web browser and go to: <https://screencast-o-matic.com/>
2. Click on **"Start recording for free"**.
3. Click **"Launch Free Recorder"** (No download required).

**Step 2: Configure Recording**

1. Choose **Full Screen** or select an area of the screen.
2. Set **Microphone Off** (if voice recording is not needed).
3. Click **Record** to start capturing the screen.

**Step 3: Capture the Defect**

1. Perform the actions that lead to the defect.
2. Highlight areas by moving the cursor.
3. Click **Pause** to stop recording.

**Step 4: Save the Video**

1. Click **Done** → **Save as Video File**.
2. Choose a location (e.g., Desktop).
3. Rename the file (e.g., "Cart\_Bug1").

**Step 5: Attach the Video to the Defect Report**

1. Open the defect reporting tool.
2. Upload the saved video.
3. Add a description explaining the issue.

**5. Benefits of Screen-O-Matic**

* **Small File Size**: Videos are around 2MB, making uploads easy.
* **Cursor Highlighting**: Shows movements and clicks clearly.
* **Fast Processing**: Saves time compared to professional video editing tools.

**6. Example Scenario**

**Issue: "Add to Cart" button does not update properly.**

1. Start Screen-O-Matic and begin recording.
2. Navigate to the website and add an item to the cart.
3. Click "Update Basket" (notice the issue where it doesn’t appear).
4. Stop recording and save the file.
5. Upload the video to the defect report.

**7. Conclusion**

Recording videos for defect reporting is a simple yet effective way to communicate issues. **Screen-O-Matic** is an easy-to-use tool that helps testers document bugs quickly and efficiently. By following these steps, you can improve the clarity of your defect reports and assist developers in fixing issues faster.