**DAY 4 - Assignment 1 MRIGANKA PATRA**

**Emp ID: 2415820**

**Email: mriganka.patra@cognizant.com**

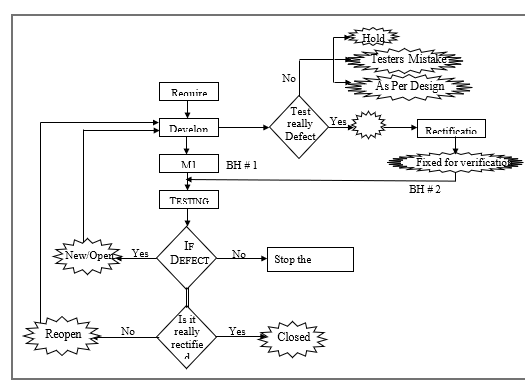
**Tasks:**

1. Define the Defect Life Cycle and illustrate its stages.

In a software development process, the bug has a life cycle. The bug should go through the life cycle to be closed. A specific life cycle ensures that the process is standardized. The bug attains different stages in the life cycle. It is as follows,

1. New vs. Open:
   1. When a defect is first identified by the tester, the initial status is set to "New".
   2. Once the developer accepts the defect, the status is changed to "Open" to indicate that the defect is being addressed.
2. Assigned
   1. The defect is assigned to a developer (or a relevant team member) for analysis and fixing.
3. Reopen and Closed:
   1. After the developer fixes the defect, the tester verifies the fix in the next build.
   2. If the fix is successful, the tester sets the status to "Closed".
   3. If the fix is not successful, the tester sets the status back to "Reopen" for further investigation and rectification.
4. Fixed for Verification, Fixed, and Rectified:
   1. When the developer fixes a defect, the status is set to "Fixed for Verification" to indicate that the fix is ready for testing.
   2. Once the tester verifies the fix, the status is updated to "Fixed" or "Rectified" to confirm the successful resolution of the defect.
5. Hold:
   1. The "Hold" status is used when the developer is unsure whether to accept or reject the reported defect.
   2. This allows for further investigation and decision-making before proceeding with the defect resolution process.
6. Testers Mistake/Error, Rejected:
   1. If the developer determines that the reported defect is not a genuine issue, they can set the status to "Testers Mistake/Error" or "Rejected".
   2. This indicates that the reported defect is not a valid defect and does not require further action.
7. As Per Design:
   1. In rare cases, the developer may determine that the reported issue is not a defect but rather a feature that aligns with the design specifications.
   2. In such cases, the developer can set the status to "As Per Design" to indicate that the behavior is intentional and not a defect**.**
8. **Reopen**  
   If during retesting the defect is still reproducible (fix is unsuccessful), it is moved back to **Reopen**, and the cycle continues.
9. **Rejected / Invalid**   
   The defect is not valid — either it is not reproducible, is by design, or is a duplicate — it is marked **Rejected.**

2. Create a diagram or flow chart of the defect life cycle.



3. List the essential fields required when logging a defect in a tracking tool.

|  |  |
| --- | --- |
| **Field** | **Purpose** |
| Defect ID (auto-generated) | Unique identifier for the defect |
| Title / Summary | Short, clear description of the defect |
| Description | Detailed explanation of the defect, including how to reproduce it |
| Steps to Reproduce | Precise steps the tester followed to encounter the defect |
| Expected Result | What the system was expected to do |
| Actual Result | What the system actually did |
| Severity | Impact of the defect on the system (e.g., Critical, Major, Minor) |
| Priority | Business urgency of fixing the defect (e.g., High, Medium, Low) |
| Environment | System/OS/browser version where the defect was found |
| Version / Build | Software version/build number where the defect was detected |
| Attachments | Screenshots, logs, videos to help developers understand the defect |
| Assigned To | Developer or team responsible for resolving the defect |
| Status | Current status (e.g., New, In Progress, Fixed, Reopened, Closed) |
| Reporter | Person who reported the defect |
| Date Created / Modified | Date/time of defect creation and updates |

4. Describe the importance of peer reviews in the defect prevention process.

**Peer Review:** A process where team members review each other’s work (code, design, test cases) before it is integrated.

**Importance:**

* **Early Detection of Defects:**
  + Helps catch defects early in the development cycle, before they reach testing or production.
* **Knowledge Sharing:**
  + Promotes understanding of code, design, and standards across the team.
* **Improved Code Quality:**
  + Ensures adherence to coding best practices and architectural guidelines.
* **Reduced Rework:**
  + Early identification and correction of issues save time and effort during later stages.
* **Team Collaboration:**
  + Encourages collaborative ownership of code and better team cohesion.
* **Defect Prevention:**
  + By identifying potential design flaws, logic errors, and poor practices before code execution, peer reviews act as a first line of defense against defects.

5. Choose a bug tracking tool (e.g., Jira, Bugzilla) and simulate the lifecycle of a sample defect.

Sample Defect: Submit button does not respond on Checkout page.

Lifecycle Simulation:

|  |  |
| --- | --- |
| Stage | Action |
| New / Open | QA tester discovers that clicking the Submit button on the Checkout page has no effect. Logs a defect in Jira with severity Critical and priority High. |
| Assigned | The defect is assigned to a frontend developer for investigation. |
| In Progress | Developer analyzes the defect, finds a missing event listener in the JavaScript code. Starts fixing the issue. |
| Resolved / Fixed | Developer commits the fix, updates Jira status to Resolved, and tags the defect with the fixed version. Adds notes about the root cause. |
| Ready for Retest | QA tester retests the Checkout page on the updated build. |
| Closed | QA verifies that the Submit button works as expected. Updates the defect status to Closed. |
| Reopened (optional) | If the issue persists or recurs, QA can reopen the defect for further investigation. |