**Prince Mohammad Bin Fahd University**

**College of Computer Engineering and Science**  
**Software Testing & Quality Assurance — Fall 2025**

**Homework 1 — Part 1: Bug Life Cycle**

**Kreem Al Salboukh**

**202001445**

**Bug Life Cycle (Bug Report Workflow)**

**Chart**

[New] → [Assigned] → [Open / In Progress] → [Resolved]

│ │

│ ├─(Fixed)

│ ├─(Duplicate)

│ ├─(Won't Fix)

│ ├─(Works for Me / Cannot Reproduce)

│ └─(Invalid)

│

└────────────→ [Reopened] ───────────┐

│

[Verified] ←────────────────────────────── [QA Verification] ───────────┘

│

└────────→ [Closed]

**Stages (Brief Description)**

* **New:** A tester/user files a bug report with steps to reproduce, expected vs. actual results, environment, severity, and attachments. The issue is not yet triaged.
* **Assigned:** A triager/lead verifies the report’s validity, sets priority/severity, and assigns it to a developer or team.
* **Open / In Progress:** The assignee analyzes root cause, reproduces locally, and begins implementing a fix (or prepares a decision rationale if not fixable).
* **Resolved:** Development provides an outcome and a **resolution** value:
  + **Fixed:** Code change implemented and merged to a branch/build.
  + **Duplicate:** Another bug tracks the same problem; this report points to the canonical one.
  + **Won’t Fix:** Acknowledged but intentionally not addressed (business, risk, or cost reasons).
  + **Works for Me / Cannot Reproduce:** Report cannot be reproduced with provided steps/environment.
  + **Invalid:** Not a product defect (e.g., user error, expected behavior, support request).
* **QA Verification (Verified):** QA/Reporter confirms the fix in the specified build/environment using the original steps-to-reproduce (and regression checks). If the issue persists, set **Reopened** and return to **Open/In Progress**.
* **Closed:** After successful verification (or administrative closure per project policy), the bug is archived as complete.

**Notes & Good Practices**

* Keep the bug’s **metadata** accurate (component, version, environment, severity/priority) throughout the cycle.
* Link commits, PRs, and test cases; include build numbers and affected versions.
* When **Reopening**, add fresh evidence (logs, screenshots, failing tests) and the exact build ID.

**References (examples)**

1. IEEE Std 1044-2009 — *Standard Classification for Software Anomalies*.
2. Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner’s Approach* (8th ed.). McGraw-Hill.
3. Mozilla Bugzilla Documentation — Bug workflow and resolution fields.
4. Atlassian (Jira) — Issue lifecycle and status workflow (general reference).

One page only: Keep content concise and the chart readable when printed.