Table of Contents

[1. Bug Triage Process & Severity Classification 2](#_Toc210286396)

[1.1 Bug Severity Classification 2](#_Toc210286397)

[1.1.1 Critical (P0/Severity 1) 2](#_Toc210286398)

[1.1.2 High (P1/Severity 2) 2](#_Toc210286399)

[1.1.3 Medium (P2/Severity 3) 3](#_Toc210286400)

[1.1.4 Low (P3/Severity 4) 4](#_Toc210286401)

[1.2 Bug Triage Process 4](#_Toc210286402)

[Step 1: Bug Submission 4](#_Toc210286403)

[Step 2: Initial Validation (QA Lead) 5](#_Toc210286404)

[Step 3: Severity & Priority Assignment 5](#_Toc210286405)

[Step 4: Triage Meeting (Daily/Weekly) 5](#_Toc210286406)

[Step 6: Verification & Closure 6](#_Toc210286407)

[1.2.1 Bug Triage Best Practices 6](#_Toc210286408)

[1.2.2 Sample Triage Decision Matrix 7](#_Toc210286409)

[1.2.3 Bug Status Workflow 8](#_Toc210286410)

[1.2.4 Roles & Responsibilities 8](#_Toc210286411)

# 1. Bug Triage Process & Severity Classification

## 1.1 Bug Severity Classification

### 1.1.1 Critical (P0/Severity 1)

**Definition:** Complete system failure or data loss that affects all users in production.

**Criteria:**

* Application crashes or becomes completely unavailable
* Data corruption or loss
* Security vulnerabilities exposing sensitive data
* Payment processing failures
* Complete loss of core functionality

**Response Time:** Immediate (within 1 hour)  
**Resolution Target:** 4-8 hours  
**Examples:**

* Production server down
* Database failure
* Critical security breach
* Payment gateway not working

### 1.1.2 High (P1/Severity 2)

**Definition:** Major functionality is broken, affecting many users or critical business operations.

**Criteria:**

* Core feature not working but workaround exists
* Significant performance degradation
* Affects large number of users
* Blocking issue for important workflows
* No workaround for key functionality

**Response Time:** Within 4 hours  
**Resolution Target:** 24-48 hours  
**Examples:**

* Login issues affecting subset of users
* Report generation failing
* Search functionality broken
* Email notifications not sending

### 1.1.3 Medium (P2/Severity 3)

**Definition:** Functionality issue with moderate impact, workaround available.

**Criteria:**

* Feature not working as expected but alternatives exist
* Affects some users or non-critical features
* Workaround is reasonable
* UI/UX issues affecting usability
* Minor performance issues

**Response Time:** Within 24 hours  
**Resolution Target:** 1-2 weeks  
**Examples:**

* Filter not working in reports
* Incorrect data display in secondary features
* Broken links in help documentation
* Form validation errors

### 1.1.4 Low (P3/Severity 4)

**Definition:** Minor issues with minimal impact on functionality.

**Criteria:**

* Cosmetic issues (UI alignment, colors, fonts)
* Minor inconveniences
* Enhancement requests
* Rare edge cases
* Documentation errors

**Response Time:** Within 1 week  
**Resolution Target:** Future release/backlog  
**Examples:**

* Text alignment issues
* Minor UI inconsistencies
* Tooltip text errors
* Feature enhancement requests

## 1.2 Bug Triage Process

### Step 1: Bug Submission

* Bug reported via bug tracking system (Jira, Azure DevOps, etc.)
* Required information:
  + Title and description
  + Steps to reproduce
  + Expected vs actual behavior
  + Environment details (browser, OS, version)
  + Screenshots/videos
  + Logs/error messages

### Step 2: Initial Validation (QA Lead)

* Verify bug is reproducible
* Check for duplicates
* Validate if it's a bug or feature request
* Add missing information if needed
* **Action:** Accept or Reject/Need More Info

### Step 3: Severity & Priority Assignment

* **QA Lead** assigns initial severity based on classification
* **Product Owner** reviews and adjusts priority based on:
  + Business impact
  + Number of users affected
  + Customer complaints
  + Release timeline
  + Available resources

### Step 4: Triage Meeting (Daily/Weekly)

**Attendees:** QA Lead, Development Lead, Product Owner, Scrum Master

**Agenda:**

* Review new bugs since last meeting
* Validate severity and priority
* Assign ownership to development team
* Identify dependencies
* Estimate effort
* Decide on fix timeline

**Outcomes:**

* Bug assigned to sprint/backlog
* Clear owner identified
* Target resolution date set

**Step 5: Assignment & Tracking**

* Bug assigned to developer
* Status updates tracked:
  + **New** → **In Progress** → **Ready for Testing** → **Closed**
* Regular status updates in daily standups

### Step 6: Verification & Closure

* QA verifies the fix in test environment
* Regression testing performed
* If passed: Mark as **Resolved**
* If failed: Reopen with details
* Deploy to production
* Final verification in production
* Close the bug

## 1.2.1 Bug Triage Best Practices

**1. Clear Communication**

* Use standardized bug templates
* Maintain clear, concise descriptions
* Include all relevant details upfront

**2. Regular Triage Cadence**

* **Critical/High:** Daily triage
* **Medium/Low:** Weekly triage
* Emergency triage for production issues

**3. Escalation Path**

**Critical Issues:**

1. QA immediately notifies Development Lead & Product Owner
2. Emergency war room if needed
3. Status updates every 2 hours

**4. Metrics to Track**

* Number of bugs by severity
* Average time to resolution by severity
* Reopen rate
* Bug backlog age
* Bugs found in production vs testing

**5. Documentation**

* Maintain triage meeting notes
* Document decisions and rationale
* Keep bug status updated
* Link related bugs/features

**6. Automation**

* Auto-assign based on component/module
* Automated notifications for SLA breaches
* Dashboard for bug metrics
* Auto-close stale bugs (after warning)

## 1.2.2 Sample Triage Decision Matrix

| **Bug Type** | **Users Affected** | **Workaround** | **Severity** | **Priority** |
| --- | --- | --- | --- | --- |
| Production down | All | No | Critical | P0 |
| Core feature broken | Many | No | High | P1 |
| Core feature broken | Many | Yes | Medium | P2 |
| Secondary feature | Few | Yes | Medium | P2 |
| UI issue | Any | N/A | Low | P3 |
| Enhancement | Any | N/A | Low | P3 |

## 1.2.3 Bug Status Workflow

NEW → OPEN → IN PROGRESS → READY FOR QA → RESOLVED → CLOSED

↓ ↓

DUPLICATE/INVALID REOPENED

↓ ↓

CLOSED IN PROGRESS

## 1.2.4 Roles & Responsibilities

**QA Lead:**

* Initial bug validation
* Severity assignment
* Facilitate triage meetings
* Verify fixes

**Product Owner:**

* Priority assignment based on business needs
* Final decision on fix timeline
* Stakeholder communication

**Development Lead:**

* Technical assessment
* Developer assignment
* Effort estimation
* Code review oversight

**Developer:**

* Bug investigation
* Fix implementation
* Unit testing
* Status updates

**Scrum Master:**

* Facilitate triage process
* Remove blockers
* Track metrics