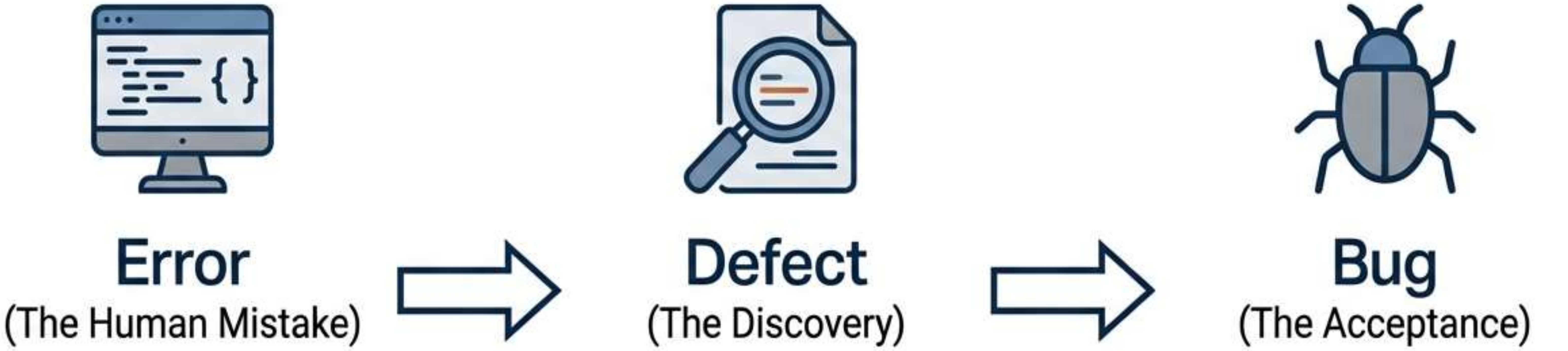
Testleaf
Always Ahead

Mastering Defect Management and the Bug Life Cycle

Operational workflows, classification logic, and decision matrices for QA professionals.

Defining the Terminology

The evolution of a code issue.



Error

(The Human Mistake)

A syntax error or logic gap made by the programmer during coding.

Defect

(The Discovery)

The variation between ‘Expected Results’ and ‘Actual Results’ identified by the Tester.

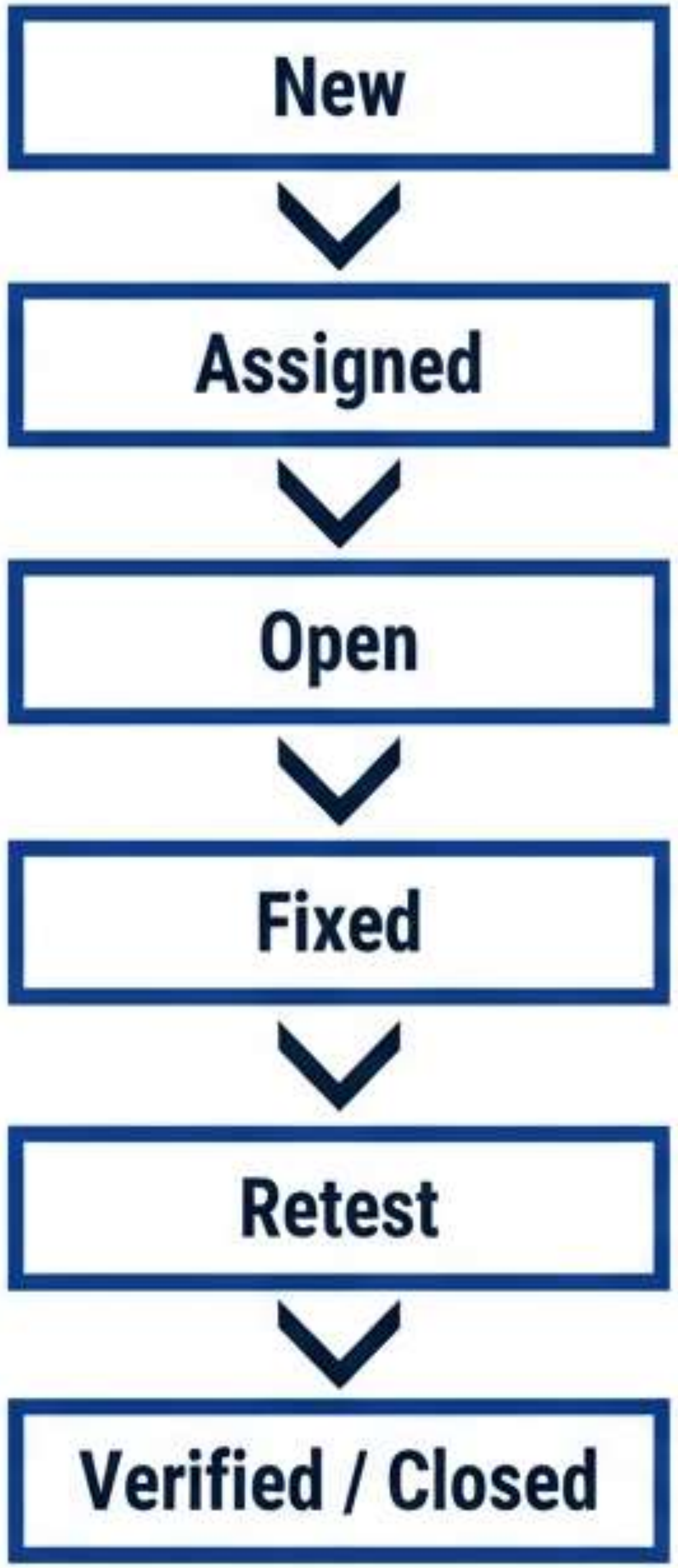
Bug

(The Acceptance)

A defect that has been accepted by the Development team to be fixed.

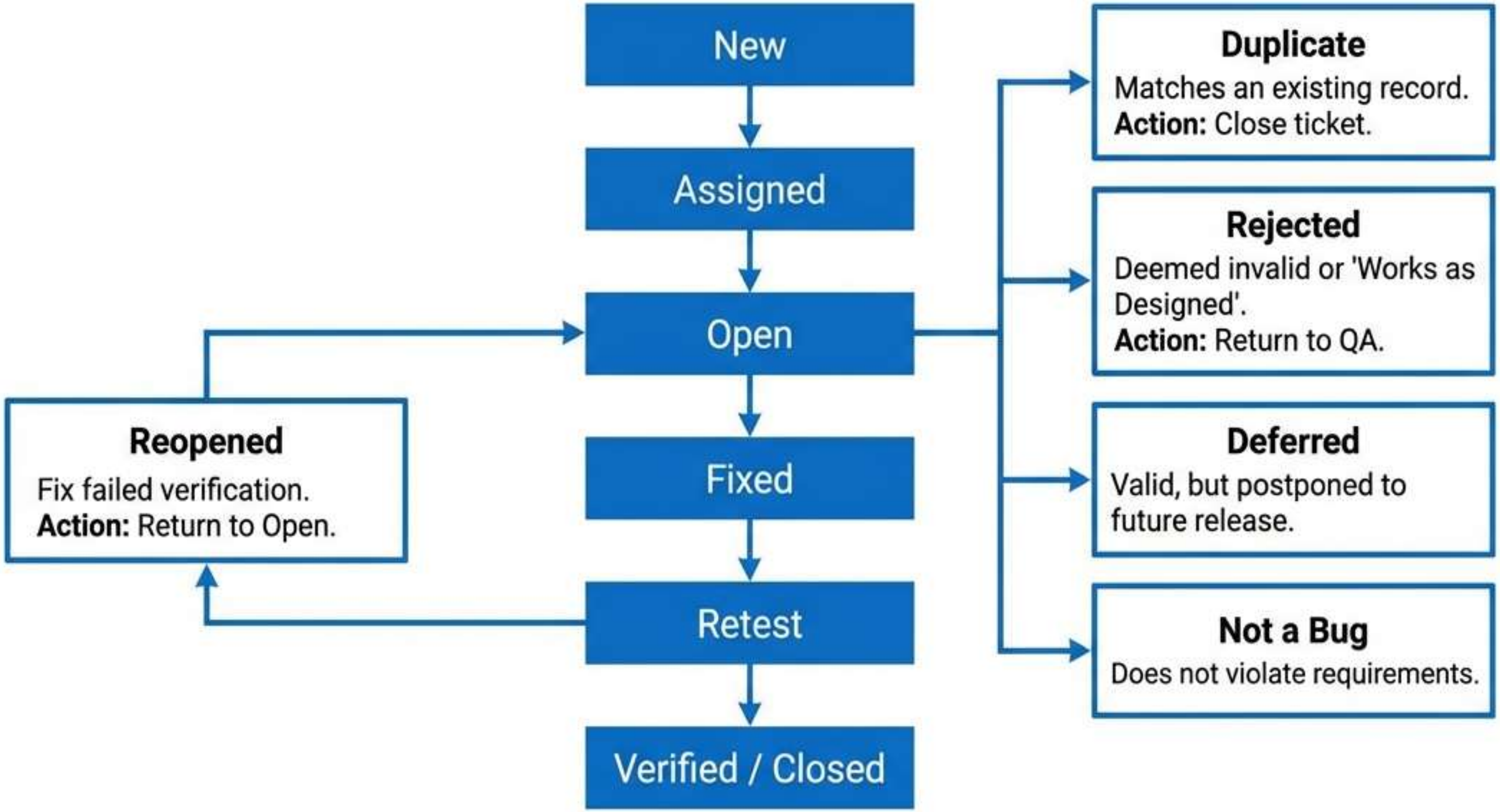
The Standard Bug Life Cycle Workflow

The 'Happy Path' from discovery to resolution



- **New**: Defect is identified and logged.
- **Assigned**: Lead assigns defect to a developer.
- **Open**: Developer begins analysis and fixing.
- **Fixed**: Code is rectified and pushed to QA.
- **Retest**: QA validates the specific fix.
- **Verified / Closed**: Fix confirmed; ticket closed.

Handling Exceptions and Non-Linear States



Duplicate
Matches an existing record.
Action: Close ticket.

Rejected
Deemed invalid or 'Works as Designed'.
Action: Return to QA.

Deferred
Valid, but postponed to future release.

Not a Bug
Does not violate requirements.

Reopened
Fix failed verification.
Action: Return to Open.

Classification Logic: Severity vs Priority

Testleaf
Always Ahead

SEVERITY

Technical Impact

How broken is the system?

A parameter indicating the impact on software functionality.

- Critical, Major, Moderate, Minor

PRIORITY

Business Urgency

How fast must it be fixed?

Decides the scheduling order for fixes based on business needs.

- Critical, High, Medium, Low

The Severity and Priority Matrix

Severity (Technical Impact)	High	Low
High	**Immediate Showstoppers.** **Examples:** System crash, Payment gateway failure, Data breach.	**High tech impact, low user frequency.** **Examples:** Legacy browser crash, Annual report crash.
Low	**Minor tech issue, high brand damage.** **Examples:** Company logo missing, Slogan spelling error.	**Cosmetic or minor annoyance.** **Examples:** Typo in ‘About Us’, Spacing inconsistency.

Priority (Business Urgency)

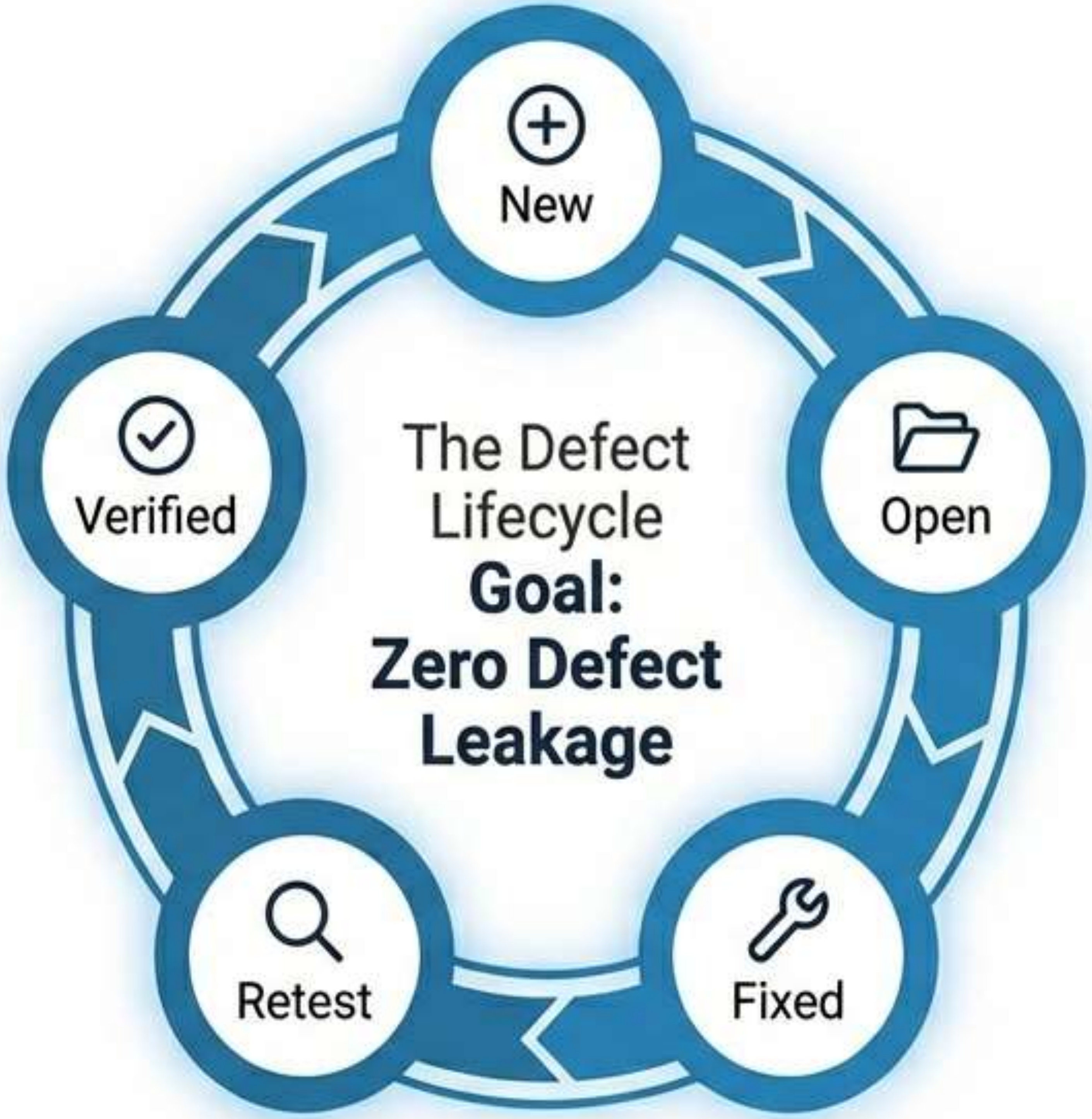
Best Practices for Defect Management

- **Objective Reporting**: Clearly differentiate between technical failure (Severity) and user experience issues (Priority).
- **Contextual Priority**: Always weigh technical severity against business value (e.g., Brand Logos).
- **Traceability**: Ensure every bug is linked to a requirement to avoid ‘Not a Bug’ rejections.
- **Clear Evidence**: Provide steps to reproduce to prevent bugs being returned as ‘Cannot Reproduce’.
- **Managed Backlogs**: Use the ‘Deferred’ state strategically, not as a hiding place for difficult bugs.

Summary & Key Takeaways

Severity measures functionality impact.

Standard Workflow: Linear Happy Path.



The Defect Lifecycle

Goal: Zero Defect Leakage

Priority measures implementation urgency.

Exceptions: Deferred, Rejected, Duplicate.