



Defect Lifecycle Management across Mock, SIT, and UAT

Sivasankararao Kavuri

Data Migration Lead, USA

ABSTRACT: Large-scale system migration programs typically execute functional and technical validation across three sequential test phases: Mock cycles, System Integration Testing (SIT), and User Acceptance Testing (UAT). Each phase carries a distinct tolerance for defects, a distinct participant profile, and, in mature programs, a distinct service level agreement (SLA) governing how quickly defects must be triaged, fixed, and retested. Without a defined SLA framework, defect backlogs accumulate unevenly across phases, severity classification drifts between teams, and go-live readiness decisions are made on inconsistent evidence.

This article presents an SLA-driven defect lifecycle management framework spanning Mock, SIT, and UAT. It defines a four-tier severity and priority taxonomy, a nine-state defect lifecycle, phase-adjusted SLA response and resolution targets, an 18-metric KPI framework, and a risk-based prioritization model. The framework is illustrated throughout with sample data, including a sample defect log extract, weekly burn-down figures across a six-week SIT wave, and a three-cycle Mock-to-UAT data pattern.

The remainder of this article is organized into 18 numbered sections and two appendices, moving from defect taxonomy and phase characteristics, through SLA and workflow design, into metrics, governance cadence, risk prioritization, exit criteria, and illustrative multi-cycle data.

KEYWORDS: Defect Lifecycle Management, Mock Testing, SIT, UAT, Defect Tracking, Root Cause Analysis, Regression Testing

I. INTRODUCTION

2.1 Purpose and Context

Defect management during a migration program is frequently treated as a purely operational activity, tracked in a defect tool but not governed by an explicit service standard. As program scope grows, particularly across multi-wave SIT cycles and parallel UAT tracks, the absence of a defined SLA framework becomes a visible risk: defects sit unassigned, severity is applied inconsistently across functional teams, and steering committees receive backlog counts without a clear signal of whether the backlog trend is improving or deteriorating relative to a target.

This article addresses that gap directly. It treats defect lifecycle management as a governed process with explicit inputs (severity, priority, phase), explicit service targets (response and resolution SLA hours), and explicit outputs (KPI dashboards, escalation triggers, and exit criteria) that connect directly to go-live decision-making.

2.2 Scope and Assumptions

The framework in this article assumes a program with at least three Mock cycles, a multi-wave SIT phase of four to eight weeks, and a UAT phase of three to six weeks, which reflects a common pattern for mid-size to large system migration programs. The numeric benchmarks presented throughout, including SLA hours, defect density targets, and aging thresholds, are illustrative reference points compiled from published testing and software quality guidance rather than figures specific to any single program, and should be recalibrated against a program's own historical data where available.

- Applies to functional, technical, and data-related defects raised during Mock, SIT, and UAT.
- Assumes a centralized defect tracking tool with configurable severity, priority, and status fields.
- Assumes a defined test management structure with a test lead per functional work stream.
- Excludes production incident management, which follows a separate operational SLA framework.



III. DEFECT TAXONOMY AND CLASSIFICATION

A consistent taxonomy is the foundation of any SLA framework. Severity describes the functional impact of a defect, while priority describes the urgency with which it should be addressed relative to other open items. The two dimensions are related but not identical, and both should be captured independently in the defect record.

3.1 Severity Classification

Severity	Definition	Response SLA	Resolution SLA
Sev 1 - Critical	Complete process blocker; no workaround available	1 hour	8 business hours
Sev 2 - High	Major function impaired; workaround exists but is costly	4 hours	24 business hours
Sev 3 - Medium	Isolated function impacted; workaround is straightforward	8 business hours	3 business days
Sev 4 - Low	Cosmetic or minor usability issue	1 business day	5 business days

TABLE 1 Severity classification and associated SLA response and resolution targets.

3.2 Priority Classification

Priority	Definition	Triage Response Target
P1 - Immediate	Business-critical; blocks continued testing	Within 2 hours
P2 - High	Must be fixed within the current test cycle	Within 1 business day
P3 - Medium	Should be fixed within the next test cycle	Within 3 business days
P4 - Low	Fix opportunistically; may defer to a later release	Backlog, next release window

TABLE 2 Priority classification and triage response targets, applied independently of severity.

3.3 Defect Lifecycle States

Nine states are recommended to give sufficient granularity for SLA measurement without introducing excessive administrative overhead for testers and fix owners.

State	Entry Criteria	Exit Criteria
New	Defect logged with severity, priority, and phase	Assigned to a triage owner
Assigned	Triage owner confirmed	Root cause analysis started
In Analysis	Root cause analysis in progress	Root cause identified and fix owner confirmed
In Progress	Fix development underway	Fix completed and unit validated
Ready for Retest	Fix deployed to test environment	Retest scheduled
Retest	Retest execution in progress	Retest passed or failed
Closed	Retest passed	No further action
Reopened	Retest failed or defect recurs	Returns to In Analysis
Deferred / Rejected	Fix deferred to later release, or defect invalid	Documented disposition, no further action in current cycle

TABLE 3 Nine-state defect lifecycle with entry and exit criteria for each state.



IV. TEST PHASE LANDSCAPE: MOCK, SIT, AND UAT

Each phase serves a different validation purpose and therefore tolerates a different defect profile. Mock cycles validate that migrated data and basic configuration behave as expected in an early, unstable environment. SIT validates end-to-end process integrity across integrated modules and interfaces. UAT validates that business users can execute their day-to-day processes against the target system with confidence.

4.1 Mock Testing Cycle Profile

Mock cycles are typically run in three iterations of one to two weeks each, focused on validating data conversion accuracy and baseline configuration. Defect volumes are highest in Mock 1 and decline sharply by Mock 3 as conversion rules and configuration stabilize.

4.2 System Integration Testing (SIT) Profile

SIT typically spans four to eight weeks across two to three waves, with each wave adding functional or interface scope. SIT defect volumes are dominated by interface and cross-module sequencing issues rather than isolated configuration errors.

4.3 User Acceptance Testing (UAT) Profile

UAT typically spans three to six weeks with business users executing end-to-end scenarios. Defect volumes are lower in absolute count than SIT but carry higher business visibility, since UAT defects are the last checkpoint before go-live and often surface usability or process-fit concerns rather than pure functional defects.

4.4 Cross-Phase Comparison

Attribute	Mock	SIT	UAT
Typical duration	1 to 2 weeks per cycle, 3 cycles	4 to 8 weeks, 2 to 3 waves	3 to 6 weeks
Primary participants	Data and configuration teams	Functional and technical testers	Business users
Defect density target (per 100 test cases)	12 to 18	6 to 10	2 to 4
SLA posture	Relaxed	Standard	Tightened
Dominant defect type	Data conversion, configuration	Interface, cross-module sequencing	Usability, process fit

TABLE 4 Comparison of Mock, SIT, and UAT phase attributes, including illustrative defect density targets.

V. SLA FRAMEWORK DESIGN

SLA targets should not be uniform across all three phases. A Sev 1 defect discovered on day two of Mock 1, in an environment known to be unstable, does not carry the same urgency as a Sev 1 defect discovered in the final week of UAT. The framework therefore applies a phase adjustment factor on top of the base severity SLA defined in Table 1.

5.1 SLA Adjustment Factors by Test Phase

Test Phase	Response SLA Multiplier	Resolution Multiplier	SLA	Rationale
Mock	2.0x base target	1.5x base target		Environment instability expected, lower business exposure
SIT	1.0x base target	1.0x base target		Standard target, full integration scope in play
UAT	0.75x base target	0.75x base target		Highest business visibility, shortest runway to go-live

TABLE.5 Phase adjustment multipliers applied to the base severity SLA targets defined in Table 1.



5.2 Escalation Matrix

Escalation is triggered by SLA breach thresholds rather than by elapsed time alone, so that a single team's aging defect does not automatically escalate to program leadership while genuine breach trends are surfaced quickly.

Breach Condition	Escalation Level	Notified Role	Response Window
Any Sev 1 breach	Level 3 - Program	Program Manager, Test Manager	Immediate, same business day
2 or more Sev 2 breaches in a week	Level 2 - Workstream	Workstream Lead, Test Lead	Within 1 business day
SLA compliance below 80 percent for a phase	Level 2 - Workstream	Test Manager	Within 2 business days
Any single defect open beyond 2x resolution SLA	Level 1 - Team	Fix Owner, Team Lead	Next daily triage call

TABLE.6 .Escalation matrix mapping SLA breach conditions to escalation level and notified role.

VI. DEFECT LIFECYCLE WORKFLOW AND GOVERNANCE ROLES

6.1 State Transition Sequence

The following sequence describes the expected path of a defect through the nine lifecycle states defined in Table 3, under normal conditions.

1. Tester logs the defect as New, assigning severity, priority, phase, and functional module.
2. Triage owner reviews within the response SLA and moves the defect to Assigned.
3. Fix owner performs root cause analysis, moving the defect to In Analysis.
4. Fix owner develops and unit-validates the fix, moving the defect to In Progress and then Ready for Retest.
5. Test lead schedules and executes retest, moving the defect to Retest.
6. If retest passes, the defect moves to Closed; if retest fails, the defect moves to Reopened and returns to In Analysis.
7. If a fix is not feasible within the current cycle, the defect moves to Deferred with a documented business disposition.

6.2 Roles and Responsibility Matrix

Activity	Business Analyst	Test Lead	Fix Owner	Test Manager
Defect logging	Consulted	Responsible	Informed	Informed
Severity and priority confirmation	Consulted	Accountable	Consulted	Informed
Root cause analysis	Informed	Consulted	Responsible	Informed
Fix development	Informed	Informed	Responsible	Informed
Retest execution and closure	Consulted	Responsible	Informed	Accountable
SLA breach escalation	Informed	Responsible	Informed	Accountable

TABLE.7.Responsibility matrix (RACI-style) for defect lifecycle activities across core program roles.

VII. DEFECT INTAKE, TRIAGE, AND SAMPLE RECORDS

7.1 Triage Cadence and Intake SLA

A daily triage call, held within one hour of the prior day's test execution close, is the primary intake control for new defects. During SIT and UAT, this cadence should increase to twice daily when the open Sev 1 or Sev 2 count exceeds five, ensuring that severity and fix ownership are confirmed well within the response SLA windows defined in Table 1.

7.2 Sample Defect Log Extract

The extract below illustrates a representative slice of an active defect log during a SIT wave, with sample identifiers and status values used for illustration only.



Defect ID	Phase	Module	Severity	Status	Age (days)
DEF-1042	SIT Wave 2	Order Management	Sev 2	In Progress	2
DEF-1043	SIT Wave 2	Finance Interface	Sev 1	Retest	1
DEF-1047	SIT Wave 2	Inventory	Sev 3	Assigned	4
DEF-1051	SIT Wave 2	Procurement	Sev 2	Ready for Retest	3
DEF-1058	SIT Wave 2	Master Data	Sev 4	New	1
DEF-1063	SIT Wave 2	Order Management	Sev 3	Reopened	6
DEF-1071	SIT Wave 2	Finance Interface	Sev 2	Closed	5
DEF-1079	SIT Wave 2	Reporting	Sev 4	Deferred	9

TABLE.8. Sample defect log extract for a single SIT wave, illustrating field structure and status distribution.

VIII. METRICS AND KPI FRAMEWORK

The framework tracks 18 discrete metrics across five groupings: volume, aging, SLA compliance, root cause distribution, and defect density. Each grouping is illustrated below with sample figures drawn from a representative SIT wave and UAT cycle.

8.1 Volume Metrics

Phase	Defects Raised	Defects Closed	Defects Open (End of Period)
Mock 1	184	151	33
Mock 2	97	92	5
Mock 3	41	40	1
SIT Wave 1	212	179	33
SIT Wave 2	168	151	17
UAT Cycle 1	58	52	6

TABLE.9. Volume metrics by phase, showing defects raised, closed, and open at period end.

8.2 Aging Metrics

Aging Bucket	Open Defect Count	Percentage of Open Backlog
0 to 2 days	38	44 percent
3 to 5 days	27	31 percent
6 to 10 days	15	17 percent
11 or more days	7	8 percent

TABLE.10. Aging distribution of the open defect backlog at a representative SIT wave checkpoint.



8.3 SLA Compliance Metrics

Severity	Resolution SLA Target	Actual Resolution	Average	SLA Compliance
Sev 1	8 business hours	7.1 business hours		94 percent
Sev 2	24 business hours	22.4 business hours		89 percent
Sev 3	3 business days	3.4 business days		82 percent
Sev 4	5 business days	4.1 business days		91 percent

TABLE.11.SLA compliance by severity for a representative SIT wave, comparing target and actual average resolution time.

8.4 Root Cause Distribution

Root Cause Category	Defect Count	Percentage of Total	Cumulative Percentage
Data conversion / mapping error	94	28 percent	28 percent
Configuration gap	76	22 percent	50 percent
Interface / integration defect	61	18 percent	68 percent
Custom code defect	48	14 percent	82 percent
Test data or environment issue	34	10 percent	92 percent
Requirement or design ambiguity	27	8 percent	100 percent

TABLE.12.Root cause Pareto distribution across a combined Mock and SIT data set of 340 defects.

8.5 Defect Density Benchmarks

Phase	Test Cases Executed	Defects Found	Density per 100 Test Cases
Mock (combined 3 cycles)	2,140	322	15.0
SIT (combined 2 waves)	3,860	380	9.8
UAT Cycle 1	1,210	58	4.8

TABLE.13.Defect density benchmarks by phase, calculated per 100 test cases executed.

IX. WEEKLY TREND REPORTING AND BURN-DOWN TRACKING

Weekly burn-down tracking gives the program a leading indicator of whether the open backlog is trending toward the exit criteria defined in Section 12, rather than relying solely on a point-in-time snapshot at the end of a phase.

Week	Opened	Closed	Net Open (End of Week)	SLA Breaches
Week 1	68	41	27	6
Week 2	54	49	32	5
Week 3	47	58	21	3
Week 4	39	44	16	2
Week 5	28	33	11	1
Week 6	14	22	3	0

TABLE.14 .Sample six-week burn-down trend for a SIT wave, showing convergence of the open backlog toward zero.



TREND A consistent week-over-week decline in net open defects, combined with a falling SLA breach count, is a stronger go-live signal than the absolute open count alone.

X. GOVERNANCE CADENCE AND REPORTING STRUCTURE

Three nested governance forums give the program both operational responsiveness and executive visibility without duplicating discussion at every level.

Forum	Frequency	Typical Attendees	Primary Focus
Daily defect triage	Daily, 30 minutes	Test leads, fix owners	New defect intake, severity confirmation, blocker identification
Weekly quality council	Weekly, 60 minutes	Test manager, workstream leads	SLA compliance review, aging backlog, escalation follow-up
Steering committee update	Biweekly or at phase milestones	Program manager, sponsors, test manager	Exit criteria status, go-live readiness, risk decisions

TABLE.15. Nested governance cadence for defect lifecycle oversight across daily, weekly, and milestone reporting.

XI. RISK-BASED DEFECT PRIORITIZATION MODEL

Severity and priority alone do not always capture the relative business risk of a defect. A simple risk score, calculated as the product of business impact, frequency of occurrence, and a severity weight, gives triage teams an additional, quantified input for sequencing fix effort when multiple defects compete for limited developer capacity.

FORMULA Risk Score = Business Impact (1 to 5) x Frequency of Occurrence (1 to 5) x Severity Weight (Sev 1 = 4, Sev 2 = 3, Sev 3 = 2, Sev 4 = 1)

Defect ID	Business Impact (1-5)	Frequency (1-5)	Severity Weight	Risk Score
DEF-1043	5	4	4	80
DEF-1051	4	3	3	36
DEF-1047	3	2	2	12
DEF-1058	2	1	1	2

TABLE.16 .Sample risk score calculation for four defects from the log extract in Table 8.

- Risk score above 60: treat as immediate fix priority regardless of remaining SLA window.
- Risk score 25 to 60: schedule fix within the current cycle, monitor SLA closely.
- Risk score below 25: standard SLA governance applies without additional escalation.

XII. EXIT CRITERIA AND GO-LIVE READINESS GOVERNANCE

Exit criteria translate the metrics defined in Section 8 into explicit go or no-go thresholds for each phase, removing ambiguity from the readiness decision.



Criterion	SIT Exit Threshold	UAT / Go-Live Threshold
Open Sev 1 defects	0	0
Open Sev 2 defects	5 or fewer	2 or fewer
Open Sev 3 defects	20 or fewer	10 or fewer
SLA compliance (Sev 1 and Sev 2 combined)	85 percent or higher	95 percent or higher
Regression test pass rate	95 percent or higher	98 percent or higher
Defect find rate (new defects per day, final week)	Declining trend for 2 consecutive weeks	Fewer than 3 new Sev 1/2 per day

TABLE.17.Exit criteria thresholds for SIT completion and UAT / go-live readiness decisions.**XIII. ILLUSTRATIVE MULTI-CYCLE DATA PATTERN**

The table below consolidates illustrative defect counts across the full Mock-to-UAT sequence for a single mid-size migration program, showing the expected downward trend in both volume and density as the program progresses toward go-live.

Cycle	Defects Raised	Sev 1	Sev 2	Sev 3	Sev 4	SLA Compliance
Mock 1	184	6	34	88	56	64 percent
Mock 2	97	2	15	47	33	74 percent
Mock 3	41	0	6	21	14	81 percent
SIT Wave 1	212	9	48	102	53	79 percent
SIT Wave 2	168	4	31	84	49	88 percent
UAT Cycle 1	58	1	9	31	17	93 percent

TABLE.18.Consolidated illustrative defect data across a full Mock-to-UAT sequence, showing declining volume and improving SLA compliance.

The pattern reflects two consistent behaviors observed across migration programs. First, severity mix shifts toward lower-severity defects as the program advances, since foundational data and configuration issues are resolved earliest. Second, SLA compliance improves steadily even as absolute defect count declines more slowly in percentage terms, reflecting a maturing triage and fix process rather than simply a smaller remaining scope.

XIV. COMMON CHALLENGES AND MITIGATIONS

Challenge	Typical Cause	Recommended Mitigation
Inconsistent severity classification across teams	Lack of shared severity definitions	Publish and train on the severity matrix in Table 1 before Mock 1 begins
Defects reopened repeatedly	Insufficient root cause analysis before fix	Require documented root cause before a defect can move to In Progress
SLA breaches concentrated in one module	Fix owner capacity constraint	Rebalance fix ownership or add capacity for the affected module
Backlog appears stable but risk is rising	Volume metrics tracked without severity mix	Track severity mix trend alongside raw open count
Late-stage UAT defects tied to process fit, not code	Insufficient business involvement during SIT	Include business representatives in select SIT scenarios

TABLE.19.Common defect lifecycle challenges observed across migration programs, with recommended mitigations.



XV. BEST PRACTICES

- Publish the severity, priority, and SLA matrices before the first Mock cycle begins, not after issues arise.
- Apply phase-adjusted SLA multipliers rather than a single fixed SLA across all three test phases.
- Require root cause documentation before any defect can move to In Progress.
- Track severity mix and aging trend, not just raw open and closed counts.
- Use the risk-based prioritization model when developer capacity is constrained, not as a replacement for severity.
- Hold a daily triage call throughout SIT and UAT, increasing to twice daily above defined thresholds.
- Tie phase exit criteria explicitly to the metrics defined in the KPI framework, not to a subjective assessment.
- Review root cause distribution at each phase boundary to identify systemic issues rather than isolated defects.
- Retain the full defect data set after go-live as a benchmark input for the next migration program.

XVI. FUTURE OUTLOOK

As of mid-2020, migration programs are increasingly executing test phases with distributed, remote teams rather than co-located war rooms, a shift that places additional weight on structured SLA governance and dashboard-driven reporting, since informal in-person escalation is less available. Programs are also showing growing interest in linking defect lifecycle data directly to test management and requirements traceability tooling, so that root cause distribution can be traced back to specific requirements or configuration decisions rather than analyzed in isolation. Continued investment in phase-adjusted SLA models and risk-based prioritization is expected as programs seek more predictable go-live readiness signals.

XVII. CONCLUSION

Defect lifecycle management across Mock, SIT, and UAT benefits from the same discipline applied to any other governed program process: clear taxonomy, explicit service targets, a defined workflow, and measurable exit criteria. The SLA-driven framework presented in this article, spanning a four-tier severity model, phase-adjusted SLA targets, an 18-metric KPI framework, and a risk-based prioritization model, gives migration programs a repeatable structure for managing defect volume and business risk consistently across all three test phases.

Programs that adopt this structure early, ideally before the first Mock cycle begins, are better positioned to make go-live decisions based on consistent, comparable evidence rather than on backlog counts that vary in meaning from team to team.

REFERENCES

- 1) Black, R. (2009). *Managing the Testing Process* (3rd ed.). Wiley.
- 2) Capgemini, Sogeti, & Micro Focus. (2018). *World Quality Report 2018-19*. Capgemini Group.
- 3) Gartner, Inc. (2017). *Best Practices for IT Service Level Agreements*. Gartner Research.
- 4) Humphrey, W. S. (1995). *A Discipline for Software Engineering*. Addison-Wesley.
- 5) IEEE. (2009). *IEEE Std 1044-2009, IEEE Standard Classification for Software Anomalies*. Institute of Electrical and Electronics Engineers.
- 6) ISO/IEC. (2011). *ISO/IEC 25010:2011, Systems and Software Quality Requirements and Evaluation (SQuaRE)*. International Organization for Standardization.
- 7) International Software Testing Qualifications Board. (2018). *Certified Tester Foundation Level Syllabus*. ISTQB.
- 8) Jones, C. (1997). *Software Quality: Analysis and Guidelines for Success*. International Thomson Computer Press.
- 9) Jones, C. (2008). *Applied Software Measurement: Global Analysis of Productivity and Quality* (3rd ed.). McGraw-Hill.
- 10) Kan, S. H. (2002). *Metrics and Models in Software Quality Engineering* (2nd ed.). Addison-Wesley.
- 11) Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
- 12) Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (6th ed.). Project Management Institute.



Appendix A: Sample Defect Record Template

The field structure below reflects the recommended minimum data captured for every defect record to support the metrics defined in Section 8.

Field	Description
Defect ID	Unique identifier assigned automatically by the defect tracking tool
Phase	Mock cycle, SIT wave, or UAT cycle in which the defect was found
Module / Functional Area	Business process or technical module affected
Severity	Sev 1 through Sev 4, per Table 1
Priority	P1 through P4, per Table 2
Status	Current lifecycle state, per Table 3
Root Cause Category	Assigned once root cause analysis is complete, per Table 12 categories
Fix Owner	Role or team accountable for the fix
Date Raised / Date Closed	Used to calculate age and SLA compliance
Risk Score	Calculated per the formula in Section 11, where applicable

TABLE.20.Recommended minimum field structure for a defect record

Appendix B: Glossary of Key Terms

Term	Definition
SLA	Service Level Agreement; a defined target for response or resolution time.
Mock Cycle	An early test cycle validating data conversion and baseline configuration.
SIT	System Integration Testing; validates end-to-end process integrity across integrated modules.
UAT	User Acceptance Testing; validates business user execution of target-state processes.
Severity	The functional impact classification of a defect.
Priority	The urgency classification of a defect relative to other open items.
Defect Density	The number of defects found per unit of test coverage, commonly per 100 test cases.
Burn-Down	A trend view of open items over time, typically tracked weekly.
RACI	Responsible, Accountable, Consulted, Informed; a responsibility assignment framework.
Exit Criteria	Explicit, measurable thresholds that must be met to complete a test phase.
Root Cause Category	A classification of the underlying cause of a defect, used for Pareto analysis.

TABLE.21.Glossary of key terms referenced throughout this article