Quality Engineering Risk-Based Testing & Production Debug Guidelines

Target Audience: QE members within Delivery PODs

Version: Draft – October 2025

1. Purpose

The purpose of this guideline is to define the role and responsibilities of Quality Engineer members in ensuring process release risks are identified, tested, and documented. QEs provides risk-based testing through attestation or Production Debugs (Prod Debug), depending on the impact and complexity of the change.

Objectives:
1. Ensure all process and environment changes are attested.
2. Capture and document defects systematically.
3. Validate that testing in production is conducted with due diligence, minimizing business and customer risk.
4. Record exceptions to standard testing with risk acceptance attached
5. Provide reusable documentation and test scenarios for future efficiency.

2. Scope

This POD primarily accepts tickets estimated at ≤5 days effort. Any work exceeding 5 days requires escalation and additional review. Risk/complexity scoring (future state) may further refine classification of tickets.

A little confused on scope:

Enhancement is complex and over 5 days

Pcr or cdn app ops <5 days , maybe quick turn around no e2e testing

Low medium high categorization? Complexity and repsosibilties outputs expected from qe?

What is out of scope

In Scope for QEs:
- Changes impacting process flows, system objects, or data handling.
- High-risk or customer-facing workflows.
- Financial or regulatory reporting processes.

Out of Scope:
- Environment management or infrastructure changes not linked to QE activities.
- Business-only activities outside QE remit.

3. QE Role Definition

QEs are responsible for:
- Collaborating with POD team members (Designers, POs, SMEs, BSA, And Provisioning analyst?).
- Contributing to determining if changes require an attestation or Prod Debug.
- Executing attestation testing and/or coordinating Prod Debug sessions.
- Capturing and documenting evidence, test outcomes, and defects.
- Ensuring rollback, compliance, and audit safeguards are respected.

4. Metrics & Proof of Value

Current baselines:
- Defect leakage before deployment: ~0.33–0.5 defects per release.
- Defects found during Prod Debug: ~2–3 minor issues, usually identified by designers.
- Incident MTTR: ~1–2 hours (typically resolved within the same day).

Target: Reduce defect leakage to ≤0.25 per release while maintaining MTTR within current levels.

Tracking to be expanded in future to include:
- % of tickets requiring Prod Debug vs Attestation.
- QEs effort distribution between attestation and Prod Debug.

5. Decision Framework – Attestation vs Prod Debug

5.1 Testing & Escalation Matrix

Condition	QE Action	Required Safeguards / Oversight
Minor process change (e.g., variable update, small object tweak, path adjustment < 5 days effort)	Attestation only	- Test evidence captured in ticket (screenshots/logs) - Designer assigned to defect fixes - No prod debug required
Considerable change to process/object (higher complexity but still reproducible in lower env)	Attestation + Extended validation	- Additional test scenarios reviewed - Peer review of outcomes - PO/QE sign-off

Production incident / urgent fix (high customer impact or urgent process maintenance)	Prod Debug	- Two-person rule (1 executes, 1 observes) - PA + TL present for login/logout - Rollback package secured - Documentation in JIRA - SME notified if absent
Lower environment limitation (cannot replicate in PAT/test, e.g., integrations, live data)	Prod Debug	- TL validates limitation - Clean room compliance enforced - Access time-boxed (2–4 hrs)
Inability to build/spy due to access issues in lower envs	Prod Debug	- Validate lack of access via TAO + LOB - SE/QE/BSA document session - PA present for compliance
Regulatory/compliance deadline (legal/audit obligation, no lower env available)	Prod Debug (exception)	- Must be validated by TAO - Full documentation and audit evidence required
Lower environments unavailable (e.g., shared drives, critical apps offline in PAT)	Prod Debug (exception)	- TAO/LOB validation required - Limited scope; rollback readiness confirmed

Prod Debug is required if any of the following apply:
- Critical workflows: customer-facing, financial, or regulatory.
- Environment gaps: lower environment coverage unavailable or non-representative.
- Complex changes: high system impact, dependencies, or critical new subprocesses.

Prod Debug may be:
- Proactive: validating new end-to-end workflows.
- Reactive: triggered when risk is identified late or incident severity is high.

6. Preconditions for Production Testing

Before a Prod Debug, QEs ensures:
1. Lower Environment Availability – Confirm if unavailable or inadequate for testing. Document evidence.

2. Test Data Readiness – Identify test data required and confirm availability, or if LOB can prepare test data.
3. Risk Acceptance & Approval – Secure approvals from PO/LOB acknowledging the need for Prod Debug.
4. Rollback Preparedness – Package and validate rollback plan before starting.
5. Resourcing & Oversight – Assign roles per Execution Guidelines (QE, Designer, PA, TL, L1/L2 support).

7. Execution Guidelines (QE-Focused)

- Access Requests: QE raises a Provisioning Team Incident Ticket, including duration, sessions, and justification.
- Outlook Calendar Invites: QE arranges invites for Prod Debug sessions (include PA, TL, Design Lead; extend to L2 support as required).
- Oversight:
  * Full Debug – PA and TL required at login/logout; logout mandatory to verify no unintended changes.
  * QE documents all objects/processes accessed during Prod Debug.
- Resourcing (Two-Person Rule): One executes, one observes. QE ensures this is followed.
- Credential Control: QE confirms access is restricted, credentials rotated after the debug session.
- Rollback Safeguard: QE verifies rollback package readiness.
- Compliance: QE ensures no customer data leakage.

8. During Prod Debug

QE Responsibilities:
- Execute checklist and test scenarios.
- Record successes/failures, screenshots, and time notes.
- Confirm expected vs actual outcomes with business.
- Monitor logs and session integrity with SE/SME.
- Ensure “happy path” and “unhappy path” are tested.

9. Closure Requirements

At session end, QE ensures:
1. Access Removal – Provisioning Analyst revokes all credentials at the end of the session.
2. Verification – Confirm no objects, processes, or code were modified and saved in production.
3. Audit Log Review – QE/Provisioning Analyst checks Blue Prism system audit logs.
4. Documentation – QE updates ticket/story with:
  - Actions taken
  - Findings & root cause
  - Defects identified
  - Recommendations for permanent fix

- Verification notes from audit log review
- Recommended next steps and summary sent to all relevant parties

5. Follow-Up Deployment – If changes are required, raise follow-up release request.
6. Rollback – If modifications are made in prod space, escalate and trigger rollback package per procedure.

10. Documentation Standards

- Attestation: screenshots, test evidence stored in Confluence.
- Defects: captured in JTMF, assigned to designers.
- Prod Debug Summaries: detailed time-stamped session reports.
- Happy vs Unhappy Paths: logged at ticket level for reusability.

11. Governance & Continuous Improvement

- Retrospectives: Owned by POD, conducted quarterly.
- Metrics Review: Track defect leakage, Prod Debug frequency, QE effort distribution.
- Audit Compliance: QE ensures no PII exposure, rollback readiness, and adherence to clean-room protocols.
- Future State Enhancements: risk scoring matrix, test repository consolidation, and automation of evidence capture.

12. End-State Vision

Quality Engineers are the risk gatekeepers within the POD. By combining attestation, structured Prod Debug practices, and reusable documentation, QEs ensures:
- Business continuity and compliance.
- Reduced defect leakage and faster resolution times.
- Consistent best practices for both proactive and reactive production testing.

Checklist:

Prepare test