Three-Point Estimation QA Test Strategy
Applying Risk-Adjusted Forecasting to Plan, Execute, and Monitor QA Effectively

---

Purpose

This document outlines a comprehensive QA Test Strategy aligned to the Three-Point Estimation tool and its corresponding Business Requirements Document (BRD). It is intended to guide QA teams in planning test activities using TE-based estimates, increasing accuracy in resource allocation, and ensuring full lifecycle coverage.

Strategic Objectives

- Embed estimation logic directly into test planning
- Enable capacity-based forecasting for sprints and regression windows
- Create traceability between estimated effort and actual execution
- Minimize surprises through what-if analysis and risk-based prioritization

---

QA Activities Aligned to Estimation

QA Phase	Example Tasks	Estimation Approach
Planning	Create test strategy, map features to coverage	TE-based forecasts for planning effort
Test Design	Author and review test cases	TE per test case type/module
Test Execution	Execute manual/automated tests, log results	Use TE to allocate time per cycle
Defect Management	Triage, reproduce, validate fixes	TE for defect lifecycle estimates
Regression & UAT	Perform regression cycles, stakeholder sign-off	Use TE to model rework/contingency

Reporting	Create dashboards, summary reports	Timebox effort using TE

Estimation Inputs

- Use O/M/P values for:
    - Test Case Writing (positive, negative, edge)
    - Smoke and sanity testing
    - Regression and retests
    - Bug triage and validation cycles
- Capture historical data when available for M (most likely)
- Use exploratory sessions or spikes to define P (pessimistic)

Test Case Estimation Sample

Test Type	O (min)	M (min)	P (min)	TE (min)
API Functional Tests	20	45	90	47.5
UI Field Validations	15	30	60	32.5
Regression Scenarios	45	90	180	95.0
Bug Retest & Reverify	10	20	40	21.7

Capacity Forecasting

1. **Calculate Total Estimated Time per Phase**
2. **Adjust Based on Available QA Resources**
    - Factor in part-time vs. full-time testers
    - Consider skill alignment (automation vs. manual)

3. **Use What-If Simulation**
    - Evaluate impact of losing/gaining resources
    - Adjust cycle plans or test scope as needed

Risk-Based Testing with Estimation

- Prioritize features with:
    - High complexity (wider O–P gap)
    - Customer visibility/impact
    - Dependencies or integration risk
- Deprioritize or batch low-risk areas with tight O/M/P range

---

Reporting & Feedback Loop

- Compare actual vs. estimated TE per task
- Use discrepancies to:
    - Tune future M values
    - Identify knowledge or tooling gaps
    - Inform sprint retrospectives

Traceability to BRD and Tool

- Align test coverage directly to BRD feature list
- Use tool to calculate TE per task and map to QA cycle plan

- Document test strategy inside tool output when exporting

Final Thoughts

This strategy empowers QA leaders to align with delivery teams using a shared estimation model. It promotes visibility, mitigates risk, and fosters better planning. When paired with actual execution data and feedback, it evolves into a predictive engine for QA excellence.