

QA vs QE vs Business Acceptance Tester

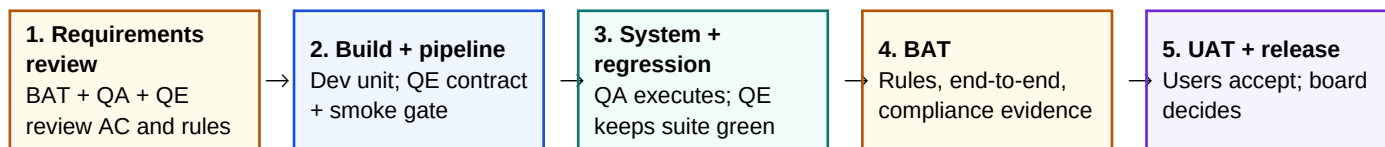
Decision Guide: who owns which quality decision, and how to find the gap on your program

ROLE 1	ROLE 2	ROLE 3
QA Analyst / Tester Question: Did we build it right? Owens: test design, system and regression execution, defect reports, test summary	Quality Engineer (QE) Question: Can we keep building it right, every commit? Owens: automation framework, CI/CD gates, test data, contract and API tests	Business Acceptance Tester Question: Did we build the right thing for the business? Owens: business-rule validation, end-to-end process, compliance evidence, BAT sign-off

Side-by-side comparison

Dimension	QA Analyst	Quality Engineer	Business Acceptance Tester
Primary goal	Detect defects against spec	Prevent defects; shorten feedback	Confirm business fit and compliance
Verification / validation	Verification	Verification at scale	Validation
When active	Refinement through release	Design through production	Requirements review, then pre-UAT
Main skills	Test design, analysis, SQL	Programming, CI/CD, APIs, test architecture	Domain rules, policy, process, data checks
Typical tools	Jira, Xray/Zephyr, TestRail, Postman	Playwright, REST Assured, Pact, GitHub Actions	Jira, Excel, SQL client, process maps
Success metric	Defects found before BAT	Change failure rate trend	Rule defects escaping to UAT/production
Counter-metric	Rejected-defect rate	Pipeline time, flaky-test rate	BAT cycle time vs plan
Blind spot	Wrong requirement tested correctly	Automates the wrong rule	Technical and non-functional risk

Hand-off flow (one release)



Feedback loop: classify every BAT, UAT and production defect by the gate that should have caught it, and feed it back to step 1 or 2.

Defect routing decision tree

Use in triage or retrospectives for any defect found in UAT or production. Stop at the first **Yes**.

Q1	Was the requirement or acceptance criterion itself wrong or missing? Yes → BAT / BA gap. Fix requirement review; add the rule to the BAT checklist.
Q2	Could an automated check at merge time have detected it (schema, contract, regression path)? Yes → QE gap. Add or fix the automated test; review quarantined flaky tests.
Q3	Did a documented test case cover it, and was it executed? Yes → QA gap. Not covered: apply the missing technique. Covered but skipped: review risk-based regression scope.
Q4	Did it appear only with production data, volume, or a downstream system? Yes → Environment / data gap. Usually QE-owned. Log as a risk if prod-like data is not permitted.

RACI summary (one A per row)

Activity	QA	QE	BAT	BA	PO	Dev
Acceptance criteria review	C	C	C	R	A	C
Automation framework	C	A/R	I	I	I	C
CI/CD quality gates	I	A	I	I	I	R
System and regression testing	A/R	R	I	I	I	C
Business-rule validation	C	I	A/R	C	I	I
Compliance evidence pack	R	C	A	C	I	I
Defect priority decision	C	C	C	I	A	C
Release go / no-go input	R	R	R	I	A	I

The three-defect escape audit

- 1 Pull the last three defects that reached UAT or production.
- 2 Run each one through the decision tree above. Record the gate owner in the Excel Defect Routing Log.
- 3 Count escapes per gate owner. The highest count is your weakest quality perspective.
- 4 Bring the tally to release planning and fund that role first: BAT time, QE automation, or QA test design.
- 5 Repeat for three releases. The trend, not one data point, should drive staffing decisions.

References: ISTQB Glossary of Testing Terms; ISTQB CTFL Syllabus v4.0; IIBA BABOK Guide v3 (Solution Evaluation); CMS-0057-F Interoperability and Prior Authorization Final Rule; HIPAA Security Rule 45 CFR 164.312(b); DORA software delivery metrics (dora.dev).