

# Risk-Based Testing Matrix

A practical matrix for ranking product areas by likelihood, impact, detectability, customer exposure, and regression confidence.

## Purpose

Use this matrix during planning, backlog refinement, regression selection, and release readiness when the team needs to test the riskiest things first.

## 1. Define Risk Inputs

- List features, workflows, APIs, integrations, data flows, reports, permissions, and platform areas in scope.
- Capture recent defect history, production incidents, customer complaints, support tickets, and business-critical journeys.
- Identify change size, technical complexity, team familiarity, dependency count, and release timing pressure.
- Include non-functional risks such as performance, accessibility, security, compatibility, reliability, and recoverability.

## 2. Score Likelihood

- Score higher when the area changed recently, has complex logic, depends on external systems, or has weak automated coverage.
- Score higher when requirements are ambiguous, ownership is unclear, or implementation involved significant refactoring.
- Score lower when the area is stable, isolated, well-monitored, and covered by reliable automated checks.
- Record assumptions behind each score so the team can challenge them during review.

### 3. Score Impact

- Score higher for revenue, compliance, legal, safety, privacy, data-loss, customer-trust, and executive-visible workflows.
- Consider blast radius: number of users affected, frequency of use, reversibility, and operational recovery effort.
- Include internal impact when failures block support, operations, analytics, finance, or delivery teams.
- Separate cosmetic annoyance from business-critical failure so priorities stay honest.

### 4. Choose Test Response

- High likelihood and high impact areas receive deep exploratory testing, targeted regression, data variation, and stakeholder review.
- High impact but lower likelihood areas receive focused confirmation, monitoring checks, and release-risk notes.
- High likelihood but lower impact areas receive efficient sampling, automation candidates, and defect-prevention feedback.
- Low likelihood and low impact areas receive lightweight smoke checks or are deferred with explicit risk acceptance.

### 5. Review and Rebalance

- Revisit scores after defects, scope changes, late merges, environment issues, or customer feedback.
- Compare matrix priorities against actual testing time to find mismatches between stated and practiced risk.
- Use the matrix in release conversations to explain what was tested deeply, sampled, deferred, or accepted.
- Archive the matrix with release notes so future teams understand why testing choices were made.