

# Exploratory Testing Charter Pack

A reusable pack of exploratory testing charters for feature discovery, risk tours, usability, integrations, data, and release confidence.

## Purpose

Use this charter pack to structure time-boxed exploratory sessions while preserving tester freedom, observation quality, and useful debriefs.

## 1. Charter Setup

- Define the mission in one sentence: explore a product area to discover risks, questions, and evidence.
- Set a time box, target environment, test data, accounts, browsers, devices, and any constraints.
- Name the primary risk focus such as functionality, usability, data, integration, performance, security, or accessibility.
- Capture notes as observations, questions, ideas, defects, and follow-up charters.

## 2. Feature Discovery Charter

- Explore the main workflow as a first-time user, an experienced user, and a user trying to recover from mistakes.
- Vary inputs, optional fields, permissions, saved states, and navigation paths.
- Look for unclear feedback, missing validation, inconsistent states, and hidden assumptions.
- Record any behavior that conflicts with requirements, expectations, or comparable product areas.

## 3. Risk Tour Charter

- Focus on the riskiest paths: revenue, compliance, security, data loss, irreversible actions, and high-traffic journeys.
- Interrupt workflows with refreshes, back navigation, duplicate submissions, timeouts, and network changes.
- Combine valid and invalid data across boundaries, roles, states, and integrations.
- Watch for silent failures, partial saves, misleading success messages, and weak recovery options.

## 4. Integration and Data Charter

- Explore how the feature behaves when upstream or downstream systems are slow, unavailable, or return unexpected data.
- Check create, read, update, delete, sync, retry, conflict, and duplicate-handling behavior.
- Use realistic data volume, missing data, special characters, localization, and old records.
- Confirm that logs, messages, and visible UI states help support teams understand failures.

## 5. Debrief Template

- Summarize what was tested, what was not tested, what was learned, and what still feels risky.
- Separate confirmed defects from questions, product suggestions, automation ideas, and future charters.
- Attach evidence such as screenshots, logs, test data, timestamps, and reproduction notes.
- Agree on next action: fix, investigate, monitor, add coverage, accept risk, or schedule another session.