

From practice to discipline: how to manage your load tests

1 TEST STRATEGY & DOCUMENTATION GOVERNANCE

Without a formal strategy document, performance testing exists only as a collection of scripts and tribal knowledge. The strategy layer is what makes it a repeatable and defensible.

- A formal performance test strategy document exists with scope, objectives, and methodology
- The strategy is version-controlled and reviewed at least once per release cycle or quarter
- There is a defined review and approval process with sign-off before a campaign begins
- Test plans are written per project or feature, distinguishing the what and why from the how
- Entry and exit criteria are formally documented before any test campaign starts

✔ **Done when:** A written, approved performance test strategy is accessible to all stakeholders, test plans exist for active campaigns, and no test campaign begins without documented entry criteria and a named approver.

2 RISK MANAGEMENT

Most performance failures are not surprises. A risk register forces the team to be explicit about what is untested and what the business consequence of a failure would be.

- A performance risk register exists listing known weaknesses, untested flows, and dependencies
- Risk priority is assigned using business impact × likelihood, not just technical complexity
- The risk register is reviewed and updated at the start of every test campaign
- Third-party and vendor SLA risks are identified and factored into test scope decisions
- There is a documented escalation path when performance risk is flagged before a release

✔ **Done when:** Every known performance risk has a recorded severity, an owner, and either a mitigation plan or a formal acceptance decision. The register is current, not a snapshot from six months ago that no longer reflects the system.

3 COMPLIANCE, AUDIT & REGULATORY READINESS

In regulated industries, performance testing is an evidence-producing function. Regulators, auditors, and enterprise procurement processes require proof that non-functional testing was performed, documented, and signed off.

- Performance testing artifacts (plans, results, sign-offs) are retained for the period required
- An audit trail exists for every test campaign (who ran it, when, against what environment, results)
- Regulatory frameworks relevant to your industry are mapped to performance testing requirements
- Performance evidence packages can be assembled on request without manual reconstruction
- There is a named owner responsible for ensuring compliance readiness of testing documentation

✔ **Done when:** Any auditor or enterprise customer can be provided with a complete, traceable record of performance testing activity, including scope, results, and sign-off chain — without requiring the team to reconstruct it from memory or scattered files.

From practice to discipline: how to manage your load tests

4 STAKEHOLDER ESCALATION & CONFLICT RESOLUTION

Without a formal process for navigating performance conflicts, decisions default to whoever has the most organizational authority, which is rarely the person with the most technical context.

- There is a documented escalation path for when performance findings conflict with release timelines
- A defined arbitration process exists, who has final authority when everyone disagree on severity
- Performance findings that are overridden by business decisions are recorded
- There is a process for escalating third-party or vendor performance failures
- Teams are aware of the escalation path before it is needed, not introduced during a conflict

✔ **Done when:** Every significant conflict between a performance finding and a business decision has been resolved through a defined process, with the outcome and rationale documented.

5 DEFECT & FINDING LIFECYCLE MANAGEMENT

Every finding needs to enter the same lifecycle as a functional defect: logged, triaged, assigned, retested, and closed. Without this, the same bottlenecks reappear release after release.

- Identify whether or not third party vendors will be included in your load tests
- Findings are triaged using a consistent severity framework
- There is a defined SLA for addressing critical and high-severity findings before release
- Be aware of any possible rate limits imposed by the third party vendor
- A performance debt backlog is maintained separately from the functional defect backlog

✔ **Done when:** Every performance finding from the current and prior campaigns exists as a tracked ticket with a status, an owner, and — where resolved — a recorded retest result confirming the fix held under load.

6 THIRD-PARTY & VENDOR PERFORMANCE ACCOUNTABILITY

Most systems depend on components they do not control. These dependencies can be the actual source of production performance failures, yet they are routinely excluded from test scope.

- All third-party dependencies are inventoried with their contractual performance SLAs and SLOs
- Performance expectations for external dependencies are included in contracts and requirements
- A process exists for escalating performance degradation caused by a third-party vendor
- Vendor performance is reviewed periodically against contractual SLAs
- There is a contingency plan for key external dependencies if the SLAs can't be met

✔ **Done when:** Every significant external dependency has documented SLAs and SLOs, a known coverage status in your test suite, and a named escalation path for performance failures.

From practice to discipline: how to manage your load tests

7 ENVIRONMENT ACCESS & SCHEDULING GOVERNANCE

Environment contention is one of the most common causes of delayed campaigns and quietly abandoned performance testing practices.

- There is a booking or scheduling system for shared test environments
- Conflicts between teams in for the same environment are resolved through a defined process
- Environment costs are allocated to the teams using them
- here is a minimum lead time requirement for booking shared environments
- Environment configuration drift is tracked and managed

✔ **Done when:** Any team can determine environment availability, book time, and begin a test campaign without negotiating access ad hoc. Conflicts have a resolution process, costs are visible, and environment state is known before every run.

8 BUDGET, RESOURCE & CAPACITY PLANNING

When these costs are invisible, they become the first thing cut under budget pressure, and the practice quietly degrades without anyone deciding to stop it.

- Performance testing infrastructure costs are estimated and tracked per release cycle
- Cloud and load generation costs are reviewed periodically against actual usage
- Tool license renewals and vendor contracts are reviewed annually against actual usage
- Performance testing is tracked across the full lifecycle design, execution, analysis, and reporting
- Capacity planning outputs from performance tests are shared with infra and platform teams

✔ **Done when:** Performance testing has a defined budget line, actual costs are tracked against it, and capacity planning recommendations derived from test results have a clear path to the teams that make infrastructure decisions.

9 CONTINUOUS IMPROVEMENT & MATURITY TRACKING

A performance testing practice that does not improve will drift. Without deliberate reflection, it silently degrades while appearing to still function.

- A retrospective is conducted after every major test campaign — not only after production incidents
- A maturity model or self-assessment framework is used to track improvement over time
- Retrospective action items have owners and deadlines, not just meeting notes
- The performance testing practice is formally reviewed against its own stated objectives
- The practice is benchmarked against industry standards at least annually

✔ **Done when:** Every major test campaign closes with a retrospective, action items are tracked to completion, and the team can point to measurable improvements in process, coverage, or result reliability compared to the previous cycle.