

Testing checklist: accounting and invoicing SaaS

Source: <https://howtotestaproduct.com/how-to-test/accounting-saas/>

Generated: 2026-10-07. Licence: CC BY 4.0.

☐ 1. release smoke tests

Done when: After each release, a test company creates and sends an invoice, records the payment and sees the trial balance still in balance.

Risks: data-corruption. Tool class: browser automation. Usually run by: QA engineer.

☐ 2. penetration testing

Done when: An external tester cannot log in by guessing common passwords, and cannot read another company's ledger by changing an organisation ID.

Risks: data-breach, fraud-abuse. Tool class: proxy. Usually run by: external lab.

☐ 3. end-to-end functional tests

Done when: A bill entered by a user for a supplier that same user created cannot be paid until a second user approves it.

Risks: fraud-abuse. Tool class: browser automation. Usually run by: QA engineer.

☐ 4. data integrity and migration testing

Done when: Importing the same bank statement twice leaves one copy of each transaction, and the VAT report totals equal the sum of the posted ledger lines.

Risks: data-corruption, regulatory-penalty. Tool class: data validation. Usually run by: QA engineer.

☐ 5. compliance checks

Done when: A reviewer traces a filed VAT return back to ledger entries with no manual export step, and finds edited or deleted entries kept in the audit log.

Risks: regulatory-penalty, data-corruption. Tool class: test management. Usually run by: product manager.

☐ 6. contract tests

Done when: Each VAT return submission in the tax authority sandbox is accepted, and the fraud prevention header check reports no missing field.

Risks: regulatory-penalty. Tool class: contract tool. Usually run by: developer.

☐ 7. integration tests

Done when: A transaction posted in the sandbox bank appears once in the bank feed with the same date, amount and payee.

Risks: dependency-failure, data-corruption. Tool class: API client. Usually run by: developer.

☐ 8. unit tests

Done when: VAT, sales tax and rounding on invoice lines match a reviewed table for each tax rate and currency offered, including credit notes.

Risks: data-corruption. Tool class: unit framework. Usually run by: developer.

☐ 9. security scanning

Done when: The latest scan of the web app and public API shows no open finding that exposes another company's ledger or payroll data.

Risks: data-breach. Tool class: DAST scanner. Usually run by: security specialist.

☐ 10. resilience and failure testing

Done when: When the bank feed provider is cut off in a test environment, the account shows a visible stale-feed warning, and the catch-up import after recovery creates no duplicates.

Risks: dependency-failure. Tool class: chaos tool. Usually run by: developer.