

Testing checklist: electronic health record system

Source: <https://howtotestaproduct.com/how-to-test/ehr-system/>

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

☐ 1. release smoke tests

Done when: After each update, a test clinician opens a test patient's chart, places a lab order and sees it arrive in the lab worklist.

Risks: user-harm. Tool class: browser automation. Usually run by: QA engineer.

☐ 2. integration tests

Done when: Every order type placed in a test chart reaches its target lab, pharmacy or specialty worklist, and an order with no valid target is refused at entry with a message.

Risks: user-harm, data-corruption. Tool class: API client. Usually run by: developer.

☐ 3. end-to-end functional tests

Done when: A test patient goes from registration through orders, results, medication administration and discharge, and each step shows in the chart with the user and time.

Risks: user-harm. Tool class: browser automation. Usually run by: QA engineer.

☐ 4. regression suite

Done when: The regression set for ordering, results and medication workflows passes on each build before it reaches a live facility.

Risks: user-harm. Tool class: test management. Usually run by: QA engineer.

☐ 5. resilience and failure testing

Done when: With the main system taken offline in a test site, clinicians open the downtime view and read current medications, allergies and active orders for a test patient.

Risks: user-harm. Tool class: chaos tool. Usually run by: developer.

☐ 6. compliance checks

Done when: A reviewer confirms that a test patient's full electronic health information exports in the standard format on request and that the access audit log is reviewed on a set schedule.

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

☐ 7. data integrity and migration testing

Done when: Medication lists and allergies for a sample of test patients match field for field between the sending and receiving records, and each required quality report produces a value.

Risks: data-corruption, wrong-decisions. Tool class: data validation. Usually run by: developer.