

Testing checklist: software as a medical device

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

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

☐ 1. release smoke tests

Done when: On the release build, the app pairs with a test device, shows a live reading and sounds a forced test alarm on each supported phone.

Risks: lost-actions, user-harm. Tool class: mobile automation. Usually run by: QA engineer.

☐ 2. integration tests

Done when: Each command the app sends reaches the test device once, the device confirms it, and the app shows the confirmed value rather than the requested one.

Risks: user-harm, lost-actions. Tool class: API client. Usually run by: developer.

☐ 3. regression suite

Done when: The regression set traced to each software requirement passes on the release candidate and the results are filed with the release record.

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

☐ 4. device and browser compatibility

Done when: On each supported phone model and OS version, including the latest OS beta, critical alarms sound with focus mode, low power mode and notification summaries turned on.

Risks: lost-actions, user-harm. Tool class: device cloud. Usually run by: QA engineer.

☐ 5. resilience and failure testing

Done when: When the sensor or transmitter is made to fail on the test rig, the app raises the failure alarm, and during a forced app crash loop the paired device's battery level drops no faster than with the app closed.

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

☐ 6. unit tests

Done when: Dose and threshold inputs written as ".5", "0.5", "5" and with a comma decimal each produce the intended value or a rejection, never a different dose.

Risks: user-harm. Tool class: unit framework. Usually run by: developer.

☐ 7. end-to-end functional tests

Done when: A test user goes from pairing through a full therapy day on a test rig, and every reading, alarm and delivered dose in the app matches the device log.

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

☐ 8. compliance checks

Done when: A reviewer confirms that every requirement in the release has a linked verification result, the risk file lists each change, and the cybersecurity documentation matches the shipped build.

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

☐ 9. security scanning

Done when: The scan report and software bill of materials for the app, firmware interface and cloud API list no open high-severity finding on the release date.

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

☐ 10. penetration testing

Done when: A tester within radio range or on the network cannot read therapy history or send a setting to the test device without pairing credentials.

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