

ISO/IEC 25010:2023 Systems and software engineering, SQuaRE, Product quality model

Quality model for software products

40 requirements. There is no accredited certificate for this standard. As of 07/2026.

ISO/IEC 25010:2023 is a quality model, not a requirements standard with auditable clauses, and therefore cannot be certified under accreditation. The model serves as a grid for stating quality requirements completely; measurement follows ISO/IEC 25020 and evaluation follows ISO/IEC 25040, and the result is a traceable evaluation report, not a certificate. Advertising a supposed ISO 25010 certification is not permissible.

Company	Date	Prepared by
---------	------	-------------

1 Functional Suitability3

1.1

The set of functions covers all specified tasks and user goals the product is used for (Functional completeness). Gaps between documented requirements and implemented functions are known and assessed.

What proves it: Evidence comes from a traceability matrix linking requirement to test case, measurable as requirements coverage in percent. In small teams an issue tracker with one label per requirement plus one acceptance test per user goal is enough.

Open

In progress

Met

Not applicable

Evidence / justification

1.2

For the defined inputs the product delivers results with the required degree of precision (Functional correctness). Tolerances and permissible deviations are specified.

What proves it: The measurable figure is defect density per thousand lines of code or per function, backed by unit and integration tests comparing expected against actual results, plus bug statistics from the tracker. For calculation logic, keep reference data sets with expected results on file.

Open

In progress

Met

Not applicable

Evidence / justification

1.3

The functions provided support the users' tasks without unnecessary steps and fit the intended use (Functional appropriateness).

What proves it: Evidence comes from task success rate and number of steps per user task in a small usability test with five participants, logged per task. Usage data or support requests that reveal detours round out the picture.

Open

In progress

Met

Not applicable

Evidence / justification

2 Performance Efficiency3

2.1

Response, processing and throughput times stay below defined thresholds (Time behaviour). The thresholds are defined for the typical usage scenarios.

What proves it: Measurable through load time and interaction latency captured with Lighthouse or Web Vitals (LCP, INP), plus server response times from the logs at the 95th percentile. Record the per scenario thresholds in a short performance budget.

Open

In progress

Met

Not applicable

Evidence / justification

2.2

Consumption of processor time, memory, storage and network bandwidth stays within the defined limits (Resource utilization).

What proves it: Evidence comes from monitoring values for CPU, RAM and data volume under load, plus the shipped bundle size from the build report. For small businesses a measurement log from one load test against the limits of the target hardware is sufficient.

Open

In progress

Met

Not applicable

Evidence / justification

2.3 The maximum number of concurrent users, transactions or records is known and observed (Capacity).

What proves it: Evidence comes from a load test (for example k6 or JMeter) with a documented saturation point and the point at which error rates start to rise. Summarise the result in a short capacity report stating the actual limit and the planned demand.

Open In progress Met Not applicable

Evidence / justification

3 Compatibility**2****3.1 The product operates in a shared environment without impairing other products (Co-existence).**

What proves it: Evidence comes from a test run in the target environment with other software running in parallel, logging port conflicts, memory usage and file access. Containerisation with defined resource limits is the practical route for small teams.

Open In progress Met Not applicable

Evidence / justification

3.2 Data exchange with other systems works through defined interfaces and formats (Interoperability).

What proves it: Evidence comes from a versioned interface description (OpenAPI or schema file) and contract tests against that description. Verify import and export formats with real sample files and keep the results on file.

Open In progress Met Not applicable

Evidence / justification

4 Interaction Capability**8****4.1 Users can tell without outside help whether the product fits their needs (Appropriateness recognizability). Purpose and benefit are evident from the interface and the entry level texts.**

What proves it: Evidence comes from a short first impression test: participants describe the purpose of the product after 30 seconds and the hit rate is recorded. Screenshots of the start page and the onboarding screens support the case.

Open In progress Met Not applicable

Evidence / justification

4.2 New users reach the core tasks within a defined time and with a limited number of errors (Learnability).

What proves it: Measurable as time to the first successful core task and failed attempts per task, captured in a test with five new users. Document onboarding, guided tour or demo data as supporting measures.

Open In progress Met Not applicable

Evidence / justification

4.3 Operation is efficient and controllable, and recurring tasks can be carried out without detours (Operability).

What proves it: Evidence comes from a click path analysis per standard task (target steps against actual steps) and from keyboard only operation without a mouse. The usability test log with the abandonment rate per task completes the record.

Open In progress Met Not applicable

Evidence / justification

4.4 The product prevents operating errors or makes them reversible (User error protection). Critical actions require confirmation or can be undone.

What proves it: Evidence comes from test cases with invalid inputs and boundary values, plus proof of validation, confirmation dialogue and undo function. Evaluate faulty inputs from the production log and track their frequency as a metric.

Open In progress Met Not applicable

Evidence / justification

4.5 The interaction is designed to be appealing and keeps users motivated to stay with the task (User engagement).

What proves it: Measurable through return rate, session duration and completion rate of started processes taken from privacy friendly usage statistics. A short satisfaction survey with a rating scale is enough on top of that, also for small businesses.

Open In progress Met Not applicable

Evidence / justification

4.6 The product is usable by people with different abilities, languages and cultural backgrounds (Inclusivity).

What proves it: Evidence comes from a check against WCAG 2.2 level AA: contrast ratios, keyboard navigation, screen reader labels, tested with axe or the Lighthouse accessibility audit plus a manual sample. Demonstrate language coverage and localisation through the completeness of the translation files.

Open In progress Met Not applicable

Evidence / justification

4.7 Users receive help that matches the task when they need it, including in error situations and edge cases (User assistance).

What proves it: Evidence comes from the manual, context sensitive help texts and understandable error messages that suggest a next action. Metric: the share of support requests that existing help could have resolved, taken from the ticket system.

Open In progress Met Not applicable

Evidence / justification

4.8 The interface explains itself: states, the effects of actions and the inputs required are recognisable without prior knowledge (Self-descriptiveness).

What proves it: Evidence comes from an interface review against a checklist: meaningful labels, visible system status, progress and loading indicators, units on numeric fields. An observation log from the user test adds weight to the record.

Open In progress Met Not applicable

Evidence / justification

5 Reliability**4****5.1 In normal operation the product runs without failures caused by faults in the software (Faultlessness).**

What proves it: Measurable through defect density and mean time between failures (MTBF) from production logs, and through test coverage as a preventive metric. Error monitoring such as Sentry supplies the raw data with little effort.

Open In progress Met Not applicable

Evidence / justification

5.2 The product is reachable and usable during the agreed time windows (Availability). The availability target is defined.

What proves it: Evidence comes from uptime measurement by external monitoring, stated as availability in percent per month and compared with the agreed target. Record outages with duration and cause in a simple incident log.

Open In progress Met Not applicable

Evidence / justification

5.3 In the presence of hardware or software faults the product stays in a defined, usable state (Fault tolerance).

What proves it: Evidence comes from test cases with simulated failures (network down, database down, third party system not responding) and the documented behaviour, for example offline mode, retry or fallback display. File the results as a test log.

Open In progress Met Not applicable

Evidence / justification

5.4 After a failure, data and operating state can be restored within the defined time (Recoverability).

What proves it: The measurable figures are recovery time objective (RTO) and maximum tolerable data loss (RPO), evidenced by an actually performed restore test with date and duration. For small businesses a half yearly restore test with a short log is sufficient.

Open In progress Met Not applicable

Evidence / justification

6 Security**6****6.1 Data is accessible only to authorised people and systems (Confidentiality). Access rights are granted on a need to know basis.**[Document](#)

What proves it: Evidence comes from a role and permission concept, encryption in transit and at rest, and tests that confirm access is denied without authorisation. A header and TLS check of the application rounds this out.

Open In progress Met Not applicable

Evidence / justification

6.2 Unauthorised changes to programs and data are prevented or detected (Integrity).

What proves it: Evidence comes from checksums or signatures of the artefacts, input validation and a dependency scan (npm audit, Dependabot) showing that findings were remediated. Database constraints and migration logs further support this point.

Open In progress Met Not applicable

Evidence / justification

6.3 Actions that were carried out cannot be denied afterwards (Non-repudiation).[Document](#)

What proves it: Evidence comes from tamper resistant logs with timestamp and originator, and from signatures where the transaction has legal relevance. A central, write protected log target is enough for small businesses.

Open In progress Met Not applicable

Evidence / justification

6.4 Actions in the system can be traced unambiguously to a responsible person or service (Accountability).[Document](#)

What proves it: Evidence comes from audit logs carrying a user identifier for every security relevant action, combined with a ban on shared accounts. Spot check: trace one transaction end to end in the log and document the result.

Open In progress Met Not applicable

Evidence / justification

6.5 The identity of users and systems is verified reliably (Authenticity).

What proves it: Evidence comes from the authentication method, the password policy and two factor authentication for administrative access, and from the validation of tokens and certificates. Test cases with expired and forged tokens demonstrate effectiveness.

Open In progress Met Not applicable

Evidence / justification

6.6 The product withstands attacks and remains operational while under attack (Resistance).

What proves it: Evidence comes from a vulnerability scan, static code analysis (SAST) and, where proportionate, a penetration test with a findings list and remediation status. For small businesses an automated scan in the pipeline plus a yearly short external test is the practical route.

Open In progress Met Not applicable

Evidence / justification

7 Maintainability					5
7.1	The product consists of separated components whose modification affects as few other components as possible (Modularity).				
What proves it: Measurable through coupling and cohesion from a static analysis and through dependency graphs (madge, dependency-cruiser) showing that no cycles exist. Capture the module boundaries in a brief architecture sketch.					
OpenIn progressMetNot applicable					
Evidence / justification					
7.2	Components are designed so that they can be reused in more than one context (Reusability).				
What proves it: Evidence comes from shared libraries or components with their own versioning and a documented interface. Metric: the share of reused components and the level of code duplication from the static analysis.					
OpenIn progressMetNot applicable					
Evidence / justification					
7.3	Causes of failures and the impact of planned changes can be determined efficiently (Analysability).				
What proves it: Evidence comes from structured logging, error monitoring and readable code metrics such as cyclomatic complexity from SonarQube or ESLint. Metric: mean time to root cause from the ticket system.					
OpenIn progressMetNot applicable					
Evidence / justification					
7.4	Changes can be made without introducing new defects and without degrading quality (Modifiability).				
What proves it: Evidence comes from regression tests in the pipeline and the change failure rate, meaning the share of defects introduced by changes. Mandatory code review and version control with traceable commits support this point.					
OpenIn progressMetNot applicable					
Evidence / justification					
7.5	Effective test criteria and tests can be established and executed for the product and its components (Testability).				
What proves it: Measurable through test coverage in percent (lines and branches) from the coverage report and through the runtime of the test suite. An automated test run on every commit is the evidence that it works.					
OpenIn progressMetNot applicable					
Evidence / justification					
8 Flexibility					4
8.1	The product can be adapted to different or changing environments without having to be rebuilt (Adaptability).				
What proves it: Evidence comes from configuration held outside the code (environment variables, configuration files) and from tests across several target environments, browsers or operating systems in a test matrix. File the result as a compatibility table.					
OpenIn progressMetNot applicable					
Evidence / justification					
8.2	The product handles growing or shrinking volumes of users and data without an unacceptable drop in performance (Scalability).				
What proves it: Evidence comes from load tests with a rising number of users, tracking response time and error rate, complemented by the scaling strategy (horizontal or vertical). Metric: response time at single, double and fivefold load.					
OpenIn progressMetNot applicable					
Evidence / justification					

8.3 The product can be installed, updated and removed reliably in the target environment (Installability).

What proves it: Evidence comes from a reproducible installation guide and an automated installation test on a clean system, measured as success rate and time required. Run uninstall and rollback once as well and log the outcome.

Open In progress Met Not applicable

Evidence / justification

8.4 The product can replace another product used for the same purpose while data and workflows are preserved (Replaceability).

What proves it: Evidence comes from import and export functions in open formats and from a migration test with real legacy data, checked for completeness and data fidelity. Keep the data model and the migration scripts under version control.

Open In progress Met Not applicable

Evidence / justification

9 Safety**5****9.1 The safe operating range is defined, and the product detects and handles any departure from that range (Operational constraint).**

What proves it: Evidence comes from documented operating limits (threshold values, permissible states) and tests that check the behaviour outside those limits. If the software carries no potential for harm to people or property, this point is typically not applicable; that classification must be justified and documented, not silently omitted.

Open In progress Met Not applicable

Evidence / justification

9.2 Hazards that can arise from the product or its use are identified and assessed systematically (Risk identification).[Document](#)

What proves it: Evidence comes from a hazard and risk assessment stating likelihood, severity and mitigating measure for each hazard, practical as a table in the quality folder. Not applicable only with a written justification that there is no potential for harm to people or property.

Open In progress Met Not applicable

Evidence / justification

9.3 In the event of a malfunction the product transitions into a safe state (Fail safe).

What proves it: Evidence comes from test cases with a forced fault and the documented transition into the safe state, for example shutting down, locking or falling back to a verified default value. For software without potential for harm this point is typically not applicable, but the justification still belongs in the evaluation.

Open In progress Met Not applicable

Evidence / justification

9.4 The product warns of impending hazardous conditions early enough for a response to be possible (Hazard warning).

What proves it: Evidence comes from an alarm and warning concept with thresholds, lead time and recipients, plus a test log of triggered warnings. Without potential for harm this point is typically not applicable; the justification has to be documented.

Open In progress Met Not applicable

Evidence / justification

9.5 Interaction with other systems does not create new hazards (Safe integration).

What proves it: Evidence comes from an interface risk review and integration tests with the connected systems, including the behaviour when a partner system fails or sends faulty data. For software without potential for harm this point is typically not applicable, and that has to be recorded with a justification.

Open In progress Met Not applicable

Evidence / justification