

ISO/IEC 25010:2023, Ingénierie des systèmes et du logiciel, SQuaRE, Modèle de qualité du produit

Modèle de qualité pour les produits logiciels

40 exigences. Il n'existe pas de certificat accrédité pour cette norme. Version de 07/2026.

L'ISO/IEC 25010:2023 est un modèle de qualité, pas une norme d'exigences avec des clauses auditable, et ne peut donc pas être certifiée par accréditation. Le modèle sert de grille pour formuler les exigences de qualité de manière complète ; la mesure suit l'ISO/IEC 25020 et l'évaluation suit l'ISO/IEC 25040, et le résultat est un rapport d'évaluation traçable, pas un certificat. Il n'est pas permis de faire de la publicité pour une prétendue certification ISO 25010.

Entreprise	Date	Établi par

1 Aptitude fonctionnelle3

1.1 L'ensemble des fonctions couvre toutes les tâches spécifiées et les objectifs des utilisateurs pour lesquels le produit est utilisé (Exhaustivité fonctionnelle). Les écarts entre les exigences documentées et les fonctions mises en œuvre sont connus et évalués.

Comment le prouver: La preuve provient d'une matrice de traçabilité reliant chaque exigence à un cas de test, mesurable comme taux de couverture des exigences en pourcentage. Dans les petites équipes, un outil de suivi des tickets avec une étiquette par exigence et un test de recette par objectif utilisateur suffit.

Ouvert En cours Satisfait Non applicable

Preuve / justification

1.2 Pour les entrées définies, le produit fournit des résultats avec le degré de précision requis (Exactitude fonctionnelle). Les tolérances et écarts admissibles sont spécifiés.

Comment le prouver: La grandeur mesurable est la densité de défauts pour mille lignes de code ou par fonction, étayée par des tests unitaires et d'intégration comparant les résultats attendus aux résultats obtenus, ainsi que par les statistiques d'anomalies de l'outil de suivi. Pour la logique de calcul, conservez des jeux de données de référence avec les résultats attendus.

Ouvert En cours Satisfait Non applicable

Preuve / justification

1.3 Les fonctions fournies soutiennent les tâches des utilisateurs sans étapes superflues et correspondent à l'usage prévu (Pertinence fonctionnelle).

Comment le prouver: La preuve provient du taux de réussite des tâches et du nombre d'étapes par tâche utilisateur lors d'un test d'utilisabilité restreint avec cinq participants, consigné par tâche. Les données d'usage ou les demandes d'assistance révélant des détours complètent le tableau.

Ouvert En cours Satisfait Non applicable

Preuve / justification

2 Efficacité de performance3

2.1 Les temps de réponse, de traitement et de débit restent inférieurs aux seuils définis (Comportement temporel). Les seuils sont définis pour les scénarios d'utilisation typiques.

Comment le prouver: Mesurable via le temps de chargement et la latence d'interaction relevés avec Lighthouse ou Web Vitals (LCP, INP), ainsi que les temps de réponse serveur issus des journaux au 95e centile. Consignez les seuils par scénario dans un court budget de performance.

Ouvert En cours Satisfait Non applicable

Preuve / justification

2.2 La consommation de temps processeur, de mémoire, de stockage et de bande passante réseau reste dans les limites définies (Utilisation des ressources).

Comment le prouver: La preuve provient des valeurs de supervision du CPU, de la RAM et du volume de données en charge, ainsi que de la taille du paquet livré issue du rapport de compilation. Pour les petites entreprises, un relevé de mesure issu d'un test de charge par rapport aux limites du matériel cible suffit.

Ouvert En cours Satisfait Non applicable

Preuve / justification

2.3 Le nombre maximal d'utilisateurs, de transactions ou d'enregistrements simultanés est connu et respecté (Capacité).

Comment le prouver: La preuve provient d'un test de charge (par exemple avec k6 ou JMeter) avec un point de saturation documenté et le point à partir duquel le taux d'erreurs commence à augmenter. Résumez le résultat dans un court rapport de capacité indiquant la limite réelle et la demande prévue.

Ouvert En cours Satisfait Non applicable

Preuve / justification

3 Compatibilité**2****3.1 Le produit fonctionne dans un environnement partagé sans nuire aux autres produits (Coexistence).**

Comment le prouver: La preuve provient d'un test réalisé dans l'environnement cible avec d'autres logiciels s'exécutant en parallèle, en consignnant les conflits de ports, l'utilisation mémoire et les accès aux fichiers. La conteneurisation avec des limites de ressources définies est la voie pratique pour les petites équipes.

Ouvert En cours Satisfait Non applicable

Preuve / justification

3.2 L'échange de données avec d'autres systèmes fonctionne via des interfaces et des formats définis (Interopérabilité).

Comment le prouver: La preuve provient d'une description d'interface versionnée (OpenAPI ou fichier de schéma) et de tests de contrat par rapport à cette description. Vérifiez les formats d'import et d'export avec de véritables fichiers d'exemple et conservez les résultats.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4 Capacité d'interaction**8****4.1 Les utilisateurs peuvent déterminer sans aide extérieure si le produit correspond à leurs besoins (Facilité de reconnaissance de la pertinence). L'objectif et le bénéfice ressortent clairement de l'interface et des textes du niveau d'entrée.**

Comment le prouver: La preuve provient d'un court test de première impression : les participants décrivent l'objectif du produit après 30 secondes et le taux de réussite est consigné. Des captures d'écran de la page d'accueil et des écrans d'intégration viennent appuyer le constat.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.2 Les nouveaux utilisateurs atteignent les tâches principales dans un délai défini et avec un nombre limité d'erreurs (Facilité d'apprentissage).

Comment le prouver: Mesurable comme le temps jusqu'à la première tâche principale réussie et le nombre de tentatives échouées par tâche, relevé lors d'un test avec cinq nouveaux utilisateurs. Documentez l'intégration, la visite guidée ou les données de démonstration comme mesures d'accompagnement.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.3 L'exploitation est efficace et maîtrisable, et les tâches récurrentes peuvent être réalisées sans détour (Facilité d'exploitation).

Comment le prouver: La preuve provient d'une analyse du parcours de clics par tâche standard (étapes cibles par rapport aux étapes réelles) et d'une exploitation possible au clavier seul, sans souris. Le journal du test d'utilisabilité avec le taux d'abandon par tâche complète le dossier.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.4 Le produit prévient les erreurs d'exploitation ou les rend réversibles (Protection contre les erreurs utilisateur). Les actions critiques exigent une confirmation ou peuvent être annulées.

Comment le prouver: La preuve provient de cas de test avec des saisies invalides et des valeurs limites, ainsi que de la validation, de la boîte de dialogue de confirmation et de la fonction d'annulation. Évaluez les saisies erronées du journal de production et suivez leur fréquence comme indicateur.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.5 L'interaction est conçue pour être attrayante et maintient les utilisateurs motivés à poursuivre la tâche (Engagement de l'utilisateur).

Comment le prouver: Mesurable via le taux de retour, la durée de session et le taux d'achèvement des processus engagés, tirés de statistiques d'usage respectueuses de la vie privée. Une courte enquête de satisfaction avec une échelle d'évaluation suffit en complément, y compris pour les petites entreprises.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.6 Le produit est utilisable par des personnes aux capacités, langues et contextes culturels différents (Inclusivité).

Comment le prouver: La preuve provient d'une vérification par rapport aux WCAG 2.2 niveau AA : ratios de contraste, navigation au clavier, libellés pour lecteurs d'écran, testés avec axe ou l'audit d'accessibilité de Lighthouse plus un échantillon manuel. Démontrez la couverture linguistique et la localisation par l'exhaustivité des fichiers de traduction.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.7 Les utilisateurs reçoivent une aide adaptée à la tâche quand ils en ont besoin, y compris dans les situations d'erreur et les cas limites (Assistance à l'utilisateur).

Comment le prouver: La preuve provient du manuel, des textes d'aide contextuelle et de messages d'erreur compréhensibles qui suggèrent une action suivante. Indicateur : la part des demandes d'assistance que l'aide existante aurait pu résoudre, tirée du système de tickets.

Ouvert En cours Satisfait Non applicable

Preuve / justification

4.8 L'interface s'explique d'elle-même : les états, les effets des actions et les saisies requises sont reconnaissables sans connaissance préalable (Caractère auto descriptif).

Comment le prouver: La preuve provient d'une revue de l'interface par rapport à une liste de contrôle : libellés explicites, état du système visible, indicateurs de progression et de chargement, unités sur les champs numériques. Un journal d'observation issu du test utilisateur renforce le dossier.

Ouvert En cours Satisfait Non applicable

Preuve / justification

5 Fiabilité

4

5.1 En exploitation normale, le produit fonctionne sans défaillance causée par des défauts du logiciel (Absence de défauts).

Comment le prouver: Mesurable via la densité de défauts et le temps moyen entre pannes (MTBF) tirés des journaux de production, ainsi que via la couverture des tests comme indicateur préventif. La supervision des erreurs, comme Sentry, fournit les données brutes avec peu d'effort.

Ouvert En cours Satisfait Non applicable

Preuve / justification

5.2 Le produit est accessible et utilisable pendant les plages horaires convenues (Disponibilité). L'objectif de disponibilité est défini.

Comment le prouver: La preuve provient de la mesure du temps de fonctionnement par une supervision externe, exprimée en disponibilité en pourcentage par mois et comparée à l'objectif convenu. Consignez les interruptions avec leur durée et leur cause dans un journal d'incidents simple.

Ouvert En cours Satisfait Non applicable

Preuve / justification

5.3 En présence de défaillances matérielles ou logicielles, le produit reste dans un état défini et utilisable (Tolérance aux pannes).

Comment le prouver: La preuve provient de cas de test avec des pannes simulées (réseau indisponible, base de données indisponible, système tiers ne répondant pas) et du comportement documenté, par exemple mode hors ligne, nouvelle tentative ou affichage de repli. Consignez les résultats dans un journal de tests.

Ouvert En cours Satisfait Non applicable

Preuve / justification

5.4 Après une défaillance, les données et l'état de fonctionnement peuvent être rétablis dans le délai défini (Aptitude au rétablissement).

Comment le prouver: Les grandeurs mesurables sont le délai de rétablissement visé (RTO) et la perte de données maximale tolérable (RPO), démontrées par un test de restauration effectivement réalisé avec date et durée. Pour les petites entreprises, un test de restauration semestriel avec un journal succinct suffit.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6 Sécurité**6****6.1 Les données ne sont accessibles qu'aux personnes et systèmes autorisés (Confidentialité). Les droits d'accès sont accordés selon le principe du besoin d'en connaître.**[Document](#)

Comment le prouver: La preuve provient d'un concept de rôles et de permissions, du chiffrement en transit et au repos, et de tests confirmant que l'accès est refusé sans autorisation. Un contrôle des en têtes et du TLS de l'application complète le dossier.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6.2 Les modifications non autorisées des programmes et des données sont empêchées ou détectées (Intégrité).

Comment le prouver: La preuve provient de sommes de contrôle ou de signatures des artefacts, de la validation des saisies et d'une analyse des dépendances (npm audit, Dependabot) montrant que les anomalies ont été corrigées. Les contraintes de base de données et les journaux de migration renforcent ce point.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6.3 Les actions réalisées ne peuvent pas être niées ultérieurement (Non répudiation).[Document](#)

Comment le prouver: La preuve provient de journaux infalsifiables avec horodatage et auteur, et de signatures lorsque la transaction a une portée juridique. Une destination de journalisation centrale et protégée en écriture suffit pour les petites entreprises.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6.4 Les actions dans le système peuvent être rattachées sans ambiguïté à une personne ou un service responsable (Imputabilité).[Document](#)

Comment le prouver: La preuve provient de journaux d'audit portant un identifiant utilisateur pour chaque action pertinente pour la sécurité, combinés à une interdiction des comptes partagés. Contrôle ponctuel : suivez une transaction de bout en bout dans le journal et documentez le résultat.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6.5 L'identité des utilisateurs et des systèmes est vérifiée de manière fiable (Authenticité).

Comment le prouver: La preuve provient de la méthode d'authentification, de la politique de mots de passe et de l'authentification à deux facteurs pour les accès administratifs, ainsi que de la validation des jetons et des certificats. Des cas de test avec des jetons expirés et falsifiés démontrent son efficacité.

Ouvert En cours Satisfait Non applicable

Preuve / justification

6.6 Le produit résiste aux attaques et reste opérationnel pendant une attaque (Résistance).

Comment le prouver: La preuve provient d'une analyse de vulnérabilités, d'une analyse statique de code (SAST) et, lorsque cela est proportionné, d'un test d'intrusion avec une liste de constats et leur statut de correction. Pour les petites entreprises, une analyse automatisée dans le pipeline plus un court test externe annuel est la voie pratique.

Ouvert En cours Satisfait Non applicable

Preuve / justification

7 Maintenabilité**5****7.1 Le produit se compose de composants séparés dont la modification affecte le moins possible d'autres composants (Modularité).**

Comment le prouver: Mesurable via le couplage et la cohésion issus d'une analyse statique et via des graphes de dépendances (madge, dependency cruiser) montrant l'absence de cycles. Consignez les limites des modules dans un court schéma d'architecture.

Ouvert En cours Satisfait Non applicable

Preuve / justification

7.2 Les composants sont conçus pour pouvoir être réutilisés dans plusieurs contextes (Réutilisabilité).

Comment le prouver: La preuve provient de bibliothèques ou composants partagés avec leur propre gestion de versions et une interface documentée. Indicateur : la part de composants réutilisés et le niveau de duplication de code issu de l'analyse statique.

Ouvert En cours Satisfait Non applicable

Preuve / justification

7.3 Les causes des défaillances et l'impact des modifications prévues peuvent être déterminés efficacement (Analysabilité).

Comment le prouver: La preuve provient de la journalisation structurée, de la supervision des erreurs et de métriques de code lisibles telles que la complexité cyclomatique issue de SonarQube ou d'ESLint. Indicateur : temps moyen jusqu'à l'identification de la cause racine, tiré du système de tickets.

Ouvert En cours Satisfait Non applicable

Preuve / justification

7.4 Les modifications peuvent être effectuées sans introduire de nouveaux défauts et sans dégrader la qualité (Facilité de modification).

Comment le prouver: La preuve provient des tests de non régression dans le pipeline et du taux d'échec des modifications, c'est à dire la part de défauts introduits par les changements. La revue de code obligatoire et le contrôle de version avec des commits traçables renforcent ce point.

Ouvert En cours Satisfait Non applicable

Preuve / justification

7.5 Des critères de test et des tests efficaces peuvent être établis et exécutés pour le produit et ses composants (Testabilité).

Comment le prouver: Mesurable via le taux de couverture des tests en pourcentage (lignes et branches) issu du rapport de couverture et via la durée d'exécution de la suite de tests. Une exécution automatisée des tests à chaque commit en constitue la preuve.

Ouvert En cours Satisfait Non applicable

Preuve / justification

8 Flexibilité

4

8.1 Le produit peut être adapté à des environnements différents ou changeants sans devoir être reconstruit (Adaptabilité).

Comment le prouver: La preuve provient d'une configuration maintenue en dehors du code (variables d'environnement, fichiers de configuration) et de tests menés sur plusieurs environnements cibles, navigateurs ou systèmes d'exploitation dans une matrice de test. Consignez le résultat sous forme de tableau de compatibilité.

Ouvert En cours Satisfait Non applicable

Preuve / justification

8.2 Le produit gère des volumes croissants ou décroissants d'utilisateurs et de données sans baisse inacceptable de performance (Évolutivité).

Comment le prouver: La preuve provient de tests de charge avec un nombre croissant d'utilisateurs, en suivant le temps de réponse et le taux d'erreur, complétés par la stratégie de mise à l'échelle (horizontale ou verticale). Indicateur : temps de réponse à charge simple, double et quintuple.

Ouvert En cours Satisfait Non applicable

Preuve / justification

8.3 Le produit peut être installé, mis à jour et désinstallé de manière fiable dans l'environnement cible (Facilité d'installation).

Comment le prouver: La preuve provient d'un guide d'installation reproductible et d'un test d'installation automatisé sur un système propre, mesuré en taux de réussite et en temps nécessaire. Effectuez également une fois la désinstallation et le retour arrière, et consignez le résultat.

Ouvert En cours Satisfait Non applicable

Preuve / justification

8.4 Le produit peut remplacer un autre produit utilisé dans le même but tout en préservant les données et les flux de travail (Remplaçabilité).

Comment le prouver: La preuve provient de fonctions d'import et d'export dans des formats ouverts et d'un test de migration avec des données réelles issues de l'ancien système, vérifié en termes d'exhaustivité et de fidélité des données. Conservez le modèle de données et les scripts de migration sous contrôle de version.

Ouvert En cours Satisfait Non applicable

Preuve / justification

9 Sécurité fonctionnelle

5

9.1 Le domaine de fonctionnement sûr est défini, et le produit détecte et gère tout écart par rapport à ce domaine (Contrainte opérationnelle).

Comment le prouver: La preuve provient de limites de fonctionnement documentées (valeurs seuils, états admissibles) et de tests vérifiant le comportement en dehors de ces limites. Si le logiciel ne présente aucun potentiel de dommage pour les personnes ou les biens, ce point n'est généralement pas applicable : cette classification doit être justifiée et documentée, et non simplement omise.

Ouvert En cours Satisfait Non applicable

Preuve / justification

9.2 Les dangers pouvant résulter du produit ou de son utilisation sont identifiés et évalués de façon systématique (Identification des risques).[Document](#)

Comment le prouver: La preuve provient d'une analyse des dangers et des risques indiquant la probabilité, la gravité et la mesure d'atténuation pour chaque danger, réalisable sous forme de tableau dans le dossier qualité. Non applicable uniquement avec une justification écrite de l'absence de potentiel de dommage pour les personnes ou les biens.

Ouvert En cours Satisfait Non applicable

Preuve / justification

9.3 En cas de dysfonctionnement, le produit passe dans un état sûr (Sécurité en cas de défaillance).

Comment le prouver: La preuve provient de cas de test avec une défaillance provoquée et de la transition documentée vers l'état sûr, par exemple arrêt, verrouillage ou retour à une valeur par défaut vérifiée. Pour un logiciel sans potentiel de dommage, ce point n'est généralement pas applicable, mais la justification doit tout de même figurer dans l'évaluation.

Ouvert	En cours	Satisfait	Non applicable
--------	----------	-----------	----------------

Preuve / justification

9.4 Le produit avertit suffisamment tôt de conditions dangereuses imminentes pour qu'une réaction soit possible (Alerte de danger).

Comment le prouver: La preuve provient d'un concept d'alarmes et d'avertissements avec seuils, délai d'anticipation et destinataires, ainsi que d'un journal de test des avertissements déclenchés. Sans potentiel de dommage, ce point n'est généralement pas applicable : la justification doit être documentée.

Ouvert	En cours	Satisfait	Non applicable
--------	----------	-----------	----------------

Preuve / justification

9.5 L'interaction avec d'autres systèmes ne crée pas de nouveaux dangers (Intégration sûre).

Comment le prouver: La preuve provient d'une revue des risques liés à l'interface et de tests d'intégration avec les systèmes connectés, y compris le comportement en cas de défaillance d'un système partenaire ou d'envoi de données erronées. Pour un logiciel sans potentiel de dommage, ce point n'est généralement pas applicable, ce qui doit être consigné avec une justification.

Ouvert	En cours	Satisfait	Non applicable
--------	----------	-----------	----------------

Preuve / justification