

PRÉNOM NOM

INGÉNIEUR QA SENIOR ET TEST AUTOMATION LEAD, PLATEFORMES À FORT TRAFIC

620 tests automatisés maintenus	22 min durée de non-régression	9 M utilisateurs couverts
0,4 % anomalies échappées en production	150 mises en production par an, contre 38	12 développeurs formés

PROFIL

Je construis des stratégies de test qui tiennent en production, pas seulement en recette. 8 ans sur des plateformes à fort trafic, dans le e-commerce, la banque en ligne et l'assurance santé. Mon terrain : réduire le temps de non-régression sans perdre de couverture, et rendre les résultats lisibles par des gens dont le métier n'est pas le test. J'interviens tôt, dès l'affinage des besoins, parce qu'une anomalie trouvée en sprint coûte environ 10 fois moins cher qu'une anomalie trouvée par un client. J'automatise ce qui doit l'être, je documente le reste, et je forme les équipes à maintenir la suite après mon départ. Je livre des chiffres, pas des impressions.

COMPÉTENCES CLÉS

Stratégie et plan de test (approche par les risques) · Automatisation E2E Playwright et TypeScript · Cypress et Selenium WebDriver · Tests API REST et GraphQL (RestAssured, Postman) · Tests de performance k6 et JMeter · Non-régression et gestion du parc de cas de test	Intégration continue GitLab CI, Jenkins, GitHub Actions · BDD, Gherkin et Cucumber · Jeux de données de test et anonymisation · Tests contractuels Pact entre services · Tests mobiles Appium (iOS et Android) · Pyramide de test et démarche shift left	Gestion des anomalies Jira et Xray · Reporting qualité et KPI (Allure, Grafana) · Monitoring qualité en production (Datadog, Sentry) · Parcours synthétiques et alerting sur production · Accessibilité RGAA et WCAG 2.1 · Coaching QA et montée en compétence des développeurs
--	--	---

EXPÉRIENCE PROFESSIONNELLE

Test Automation Lead, plateforme e-commerce janvier 2023 à aujourd'hui

[Nom du client] opère une plateforme de vente en ligne qui sert 9 millions de visiteurs uniques par mois. À mon arrivée, elle livrait 38 mises en production par an. La QA était éclatée entre 6 squads produit, sans socle commun : chaque équipe automatisait dans son coin, la non-régression tournait 4 heures et plus personne ne lisait les rapports. J'ai été missionné pour construire un socle unique et le faire adopter, en restant embarqué dans les squads plutôt qu'en cellule à part. Organisation Scrum, 42 personnes côté produit, dont 12 développeurs sur mon périmètre paiement et tunnel de commande.

- Définir la stratégie de test des 6 squads, de la pyramide de test aux critères de sortie de release, pour arrêter de rejouer 4 heures de scénarios avant chaque livraison.
- Refondre la suite de non-régression sur Playwright, en repartant des 1 400 cas existants pour ne garder que les 620 qui couvrent réellement les parcours à risque.
- Industrialiser l'exécution dans GitLab CI avec parallélisation sur 8 agents, déclenchement à chaque merge request et rapport Allure publié automatiquement.
- Mettre en place la surveillance qualité en production : 24 parcours synthétiques joués toutes les 15 minutes, alertes Slack et corrélation avec les traces Datadog.
- Former 12 développeurs à écrire et maintenir leurs propres tests, avec revue de code des tests et permanence QA de 2 heures par semaine.

Résultats : Non-régression complète ramenée de 4 heures à 22 minutes. Le rythme de livraison est passé de 38 à 150 mises en production par an, soit 3 par semaine. · Anomalies bloquantes détectées en production divisées par 3 en 12 mois, de 1,2 % à 0,4 % des tickets ouverts.

Environnement technique : Playwright · TypeScript · GitLab CI · Docker · Allure · k6 · Datadog · Xray · PostgreSQL

Ingénieur QA automatisation, application bancaire mobile et API

mars 2020 à décembre 2022

[Nom du client], banque en ligne, refondait son application mobile et les 140 API qui la servent, pour 2,3 millions de clients actifs. Contrainte réglementaire forte sur l'authentification forte des paiements. J'ai rejoint une équipe QA de 4 personnes dans un train SAFE de 5 équipes, en organisation trimestrielle. À mon arrivée, la recette était manuelle à 80 % et chaque version majeure demandait 3 semaines de campagne, avec des parcours de virement qui ne pouvaient pas casser.

- Automatiser les 210 cas de non-régression des parcours d'ouverture de compte, de virement et d'authentification forte, en Cypress pour le web et Appium pour le mobile.
- Construire la couche de tests API sur 140 endpoints avec RestAssured, cas d'erreur et temps de réponse inclus, exécutée à chaque build.
- Mettre en place des tests contractuels Pact entre le front mobile et 9 services back, pour attraper les régressions d'intégration avant l'assemblage.
- Piloter les campagnes de recette métier avec 15 testeurs : préparation des jeux de données anonymisés, animation quotidienne, arbitrage des anomalies.
- Tenir le reporting qualité auprès du responsable de train et des Product Owners, jusqu'à la décision de go ou no go avant chaque mise en production.

Résultats : Campagne de recette avant version majeure réduite de 3 semaines à 5 jours, sans perte de couverture sur les parcours réglementaires. · Régressions d'intégration entre mobile et back ramenées à 2 par trimestre, contre 17 l'année précédente, grâce aux tests contractuels.

Environnement technique : Cypress · Appium · RestAssured · Java · Pact · Jenkins · Xray · Docker · Kibana

Testeur fonctionnel puis automaticien, SI assurance santé

septembre 2017 à février 2020

[Nom du client], mutuelle santé de 900 collaborateurs, migrait son outil de gestion des contrats et des remboursements vers une nouvelle plateforme. 400 000 adhérents concernés, 18 mois de projet, cycle en V avec des paliers de recette de 6 semaines. Je suis arrivé comme testeur fonctionnel sur le lot remboursements, aux côtés de 2 référents métier et de 2 éditeurs. J'ai pris l'automatisation en charge quand il est devenu évident que rejouer 900 cas à la main à chaque palier n'était pas tenable.

- Rédiger et exécuter 340 cas de test fonctionnels sur le calcul des remboursements, à partir des règles de gestion et des grilles tarifaires du métier.
- Automatiser les 180 cas les plus rejoués en Selenium WebDriver et Java, avec un modèle Page Object pour tenir la maintenance dans la durée.
- Créer un harnais de comparaison entre l'ancien et le nouveau système sur 50 000 dossiers de remboursement, afin de prouver l'équivalence des calculs avant bascule.
- Qualifier et suivre 1 100 anomalies dans Jira, en tenant la priorisation hebdomadaire avec les référents métier et les 2 éditeurs.
- Animer les ateliers de recette avec 20 gestionnaires : préparation des scénarios, formation à l'outil, collecte et reformulation des retours.

Résultats : Bascule réalisée sans anomalie bloquante en production, sur un périmètre de 400 000 adhérents et 160 millions d'euros de prestations remboursées par an. · Durée du palier de recette ramenée de 6 à 2 semaines dès le 3e palier, grâce aux 180 cas automatisés.

Environnement technique : Selenium WebDriver · Java · JUnit · Maven · Jenkins · SQL Server · Jira · Squash TM · Postman

FORMATION

Master 2 Génie logiciel, parcours qualité et validation · [Nom de l'université]

septembre 2015 à juin 2017

Licence Informatique, parcours développement d'applications · [Nom de l'université]

septembre 2012 à juin 2015

CERTIFICATIONS

- ISTQB Certified Tester Foundation Level (CTFL)
- ISTQB Advanced Level Test Automation Engineer (CT-AE)
- Professional Scrum Master I (PSM I)

LANGUES

Français, langue maternelle · Anglais, courant professionnel (TOEIC 915), rituels et documentation en anglais au quotidien