

RÉUSSIR
L'AUTOMATISATION
DU PROCESSUS DE
TEST FONCTIONNEL

L'automatisation du processus de test fonctionnel optimise la qualité
des logiciels et maximise leur valeur opérationnelle.

Sommaire

Présentation générale	3
Que sont les tests fonctionnels ?	4
Pourquoi automatiser les tests fonctionnels ?	4
Comment réussir l'automatisation du processus de test fonctionnel ?	5
Quelles sont les meilleures pratiques d'automatisation des tests fonctionnels ?	6
Que rechercher dans une solution d'automatisation des tests fonctionnels ?	7
Borland® SilkTest® – La solution d'automatisation des tests fonctionnels	8
Gestion du cycle de vie qualité	9
Synthèse	9

Présentation générale

Les organisations de développement logiciel sont aujourd'hui face à un triple challenge : elles doivent optimiser la qualité d'applications toujours plus complexes ; dans des délais et avec des budgets plus serrés que jamais ; afin de fournir des solutions informatiques « gagnantes » grâce au ROI et au développement de nouveaux avantages concurrentiels qu'elles procurent.

Comment certaines organisations de développement ont-elles répondu aux multiples challenges des livraisons de logiciels complexes ? Par l'automatisation répond-t-on généralement... Il y a en effet d'excellentes raisons à cela puisque un grand nombre d'entreprises de pointe – quels qu'en soient les domaines et les dimensions – ont démontré la puissance des plates-formes d'automatisation des tests fonctionnels pour accélérer les délais de développement et réduire les coûts de livraison d'applications de haute qualité. Cependant, près de 85 % des entreprises échouent dans cette démarche d'automatisation¹. Ce taux est symptomatique d'un état d'esprit organisationnel consistant à envisager l'automatisation comme une solution rapide et prête à l'emploi en croyant qu'il suffit d'acquérir le meilleur produit d'automatisation des tests pour résoudre « comme par magie » des problèmes de manque de temps et de ressources...

L'automatisation des tests permet indiscutablement aux entreprises de produire des logiciels de meilleure qualité tout en maximisant l'intégration des ressources existantes ; celles qui réussissent savent néanmoins que l'automatisation des tests fonctionnels n'apporte qu'une partie de la réponse... En effet, pour optimiser la qualité logicielle, accélérer les délais de mise sur le marché et maximiser le ROI, il est indispensable de penser au-delà d'un simple outil d'automatisation et d'inscrire les tests fonctionnels dans une perspective stratégique plus étendue impliquant : des investissements initiaux pour planifier un processus de développement logiciel piloté par les impératifs de qualité ; la formation de personnels dédiés et le développement d'un framework de test adapté pour pleinement bénéficier des avantages de l'automatisation.

Ce livre blanc fournit des informations pratiques et précises sur l'expérience d'entreprises ayant réussi à automatiser leur processus de test fonctionnel et s'attache à répondre aux questions suivantes :

- Quel est le rôle stratégique des tests fonctionnels dans l'entreprise moderne ?
- Quels sont les avantages à valeur ajoutée associés à l'automatisation du processus de test fonctionnel ?
- Quelle est la meilleure approche pour garantir le succès des initiatives d'automatisation ?
- Que rechercher dans une solution d'automatisation des tests fonctionnels ?

Que sont les tests fonctionnels ?

Les tests fonctionnels (aussi nommés tests en « boîte noire ») sont un processus d'assurance qualité permettant de vérifier que les fonctionnalités d'une application destinées à l'utilisateur final (capacité à se connecter, à réaliser une transaction, etc.) fonctionnent de façon précise, fiable, sécurisée et prévisible. Que le processus de test fonctionnel soit mis en place manuellement ou de façon automatisée, il exige de mener des séries de tests d'émulation des interactions entre l'utilisateur et l'application pour valider sa capacité à obtenir les résultats escomptés.

Pourquoi automatiser les tests fonctionnels ?

Les tests manuels sont appropriés dans certains cas mais restent toujours longs, fastidieux et en inadéquation avec la brièveté des cycles de développement actuels. Ces inconvénients empêchent de réaliser des tests minutieux et laissent passer des bogues – pouvant parfois s'avérer critiques à l'utilisation. Par ailleurs, lorsque des applications doivent fonctionner sur plusieurs plates-formes, la charge des tests manuels croît proportionnellement en multipliant les risques d'erreurs humaines et d'incohérences susceptibles d'altérer les résultats finaux.

L'automatisation des tests génère de nouveaux vecteurs d'efficacité qui accélèrent les cycles de validation et optimisent la qualité logicielle. L'automatisation des tests de régression et d'autres tâches répétitives permet par exemple aux équipes d'assurance qualité de se libérer de ces opérations fastidieuses pour faire porter leurs efforts sur des aspects plus stratégiques – et notamment d'optimiser la couverture de test en étendant l'automatisation à des secteurs n'ayant pas bénéficié d'une vérification aussi poussée dans les versions précédentes.

L'automatisation du processus de test optimise la qualité logicielle et l'efficacité opérationnelle ; elle présente les avantages à long terme suivants :

- **Réemploi** – L'automatisation des tests permet aux équipes d'assurance qualité de réduire les délais de livraison en réemployant des tests déjà existants sans en recréer à chaque nouveau projet. L'expérience démontre que les tests réemployables sont davantage utilisés et permettent aux équipes de déceler et corriger un plus grand nombre d'erreurs plus tôt dans le cycle de développement. Cette faculté de réemploi permet également aux équipes d'assurance qualité de constituer des bibliothèques de ressources de test – devenant à terme des actifs de propriété intellectuelle générateurs de valeur.
- **Prévisibilité et cohérence** – Grâce à l'automatisation, les équipes d'assurance qualité accèdent à de nouveaux seuils de cohérence lors de la réutilisation des tests. Les tests de régression permettent notamment de rapidement vérifier que les fonctionnalités de la version précédente sont toujours opérationnelles et de fournir un retour immédiat aux équipes de développement.
La cohérence s'applique également au processus de test lui-même grâce à une procédure reproductible de documentation des résultats permettant de recréer et vérifier les erreurs afin d'accélérer leur résolution.
- **Productivité** – L'automatisation des tests crée un environnement à haute productivité permettant aux entreprises d'étendre leur capacité de test sans ajout de ressources ; le département d'assurance qualité peut ainsi lancer des tests sans surveillance (24X7) et valider simultanément le bon fonctionnement d'une application sur plusieurs plates-formes, navigateurs et environnements. Cette organisation libère des ressources pour se focaliser sur d'autres aspects de la qualité. Ces gains de productivité présentent le double avantage de raccourcir les cycles de test et de multiplier les opportunités d'amélioration de la qualité logicielle.

Ces multiples bénéfices de l'automatisation permettent aux équipes d'évaluer très précisément les niveaux de qualité, de prendre des décisions plus pertinentes quant aux délais de mise sur le marché et de minimiser les risques de déploiement.

Comment réussir l'automatisation du processus de test fonctionnel ?

1. Déterminer quelles applications automatiser

Le domaine d'intervention privilégié de l'automatisation porte avant tout sur les applications d'entreprise appelées à connaître de multiples versions au cours de leur cycle de vie (en raison de fréquentes extensions ou modifications fonctionnelles) et sur celles devant produire des jeux de résultats cohérents à partir de données relativement stables. Ces caractéristiques applicatives permettent en effet aux efforts de qualité de pleinement bénéficier des avantages de réemploi et de prévisibilité des tests fonctionnels automatisés.

2. Choisir une approche d'automatisation des tests

Il existe différentes méthodologies pour créer des tests fonctionnels automatisés.

- **La modularité** est une approche qui subdivise l'application testée en différents composants ou modules de script. A l'aide d'un langage de script, les équipes d'assurance qualité génèrent une couche d'abstraction devant chaque composant – ayant pour effet de le « masquer » vis-à-vis du reste de l'application. Cette organisation optimise la maintenance et l'évolutivité des suites de tests automatisés en protégeant le reste de l'application des modifications survenant sur l'un de ses composants.
- **L'architecture de bibliothèque** est un autre framework basé sur des scripts consistant à diviser l'application en modules utilisés pour construire des tests. Contrairement à l'approche modulaire, l'architecture de bibliothèque décrit les modules à travers des procédures et fonctions plutôt que des scripts – pour maximiser la modularité, la simplicité de maintenance et le réemploi.
- **Le pilotage par mots-clés** (ou tests orientés tables) constituent un framework indépendant de l'application, exploitant des tables de données et des mots-clés intelligibles pour décrire les actions à mener avec l'application testée. Ces tables de données et mots-clés ne dépendent ni de la solution d'automatisation des tests qui les exécute, ni des scripts de test qui pilotent l'application et ses données. Les tests pilotés par mots-clés permettent d'associer des experts fonctionnels au processus de test ; il est en effet possible à des intervenants non-techniques de créer des tests automatisés en entrant dans une grille des termes familiers – sans développement de script ni programmation. L'initialisation exige de définir le vocabulaire, les tables de mots clés et les composants et modules de script sous-jacents. Cependant, une fois mise en œuvre, cette méthode s'avère particulièrement efficace et performante.
- **Le pilotage par les données** propose un framework pour stocker les données dans un fichier externe (comme une feuille de calcul) au lieu de les coder à l'intérieur des scripts ; de la sorte, un seul script permet de tester toutes les valeurs souhaitées. Les données étant écartées des scripts, cette méthode les protège contre toute modification potentielle des données. La mise à jour des données requiert une simple modification du tableau sans maintenance du script. Cette approche optimise la productivité des ingénieurs de test en simplifiant la maintenance et en maximisant les opportunités de réemploi des scripts d'une version à l'autre. Les tests orientés données sont souvent implémentés conjointement aux frameworks présentés précédemment.
- **L'enregistrement/exécution** élimine la nécessité de codifier des scripts pour capturer les cas de test. Elle débute par l'enregistrement des entrées manuelles réalisées avec l'application testée qui sont ensuite utilisées pour générer des scripts de test automatisés pouvant être « rejoués » ultérieurement. Bien que l'approche enregistrement/exécution soit la plus simple et rapide pour automatiser des tests, ces derniers ne peuvent pas être maintenus ni réutilisés par la suite. Toute modification de l'application exige de réenregistrer l'ensemble de la séquence – ce qui réduit à néant les gains de temps réalisés précédemment. De fait, la méthode d'automatisation par enregistrement/exécution est celle dont le ROI, le taux de réemploi et la productivité des ressources sont les plus faibles.

3. Développer un plan de test des applications

Le plan de test est un document détaillant le périmètre, l'approche, les ressources, la couverture et le calendrier de toutes les activités (manuelles ou automatisées) de test d'une application. Plus spécifiquement, le plan de test identifie : l'ensemble des fonctionnalités à tester, les tâches requises, les personnes responsables, l'environnement de test, l'approche choisie pour la conception du test, les différentes plates-formes et environnements cibles ainsi que les métriques de reporting des résultats.

4. Créer et déployer des tests automatisés

Le plan de test tient lieu de « feuille de route » pour créer des tests automatisés ; les équipes d'assurance qualité développent ensuite chaque test grâce à une solution supportant l'approche d'automatisation choisie (modularité, architecture de bibliothèque, mots-clés, données, enregistrement/exécution). Lorsque les tests sont mis en correspondance avec les besoins, définis et créés : le processus d'automatisation peut commencer.

Quelles sont les meilleures pratiques d'automatisation des tests fonctionnels ?

Gérer l'automatisation comme un projet logiciel à part entière

Tout comme le développement logiciel, l'automatisation des tests doit être consciencieusement conçue, documentée et évaluée. L'expérience prouve comme toujours que les projets mal préparés et les implémentations hâtives sont voués à l'échec.

Les entreprises qui réussissent adoptent une vision holistique de la qualité, englobant l'intégralité du cycle de vie de l'application logicielle, et appliquent les principes des meilleures pratiques lors de l'implémentation de l'automatisation des tests. Les premiers tests interviennent en particulier très tôt dans le processus de développement – voire dès la phase de définition des besoins – et s'appuient sur une structure modulaire du code pour en simplifier la maintenance. En outre, les ressources de test sont développées dans une perspective de réemploi lors de phases ultérieures du cycle qualité ou pour de futures versions de l'application.

L'implémentation d'une solution d'automatisation des tests n'est pas un projet à court terme ; il s'agit d'une activité à temps plein nécessitant un investissement initial significatif en temps et ressources. Les attentes des décideurs doivent donc être établies en connaissance de cause, tout en sachant que l'automatisation ne peut en aucun cas se substituer à l'expertise... C'est pourquoi les entreprises qui réussissent investissent dans des formations appropriées et dans le développement des compétences sur les meilleures pratiques d'automatisation, les méthodologies de test et les solutions d'automatisation elles-mêmes.

Mieux collaborer pour optimiser la qualité

La qualité n'est pas le domaine réservé des testeurs du département d'assurance qualité mais doit impliquer l'ensemble de l'entreprise. En créant un environnement collaboratif entre experts fonctionnels et développeurs lors du pré-déploiement et avec les acteurs opérationnels après déploiement, les équipes d'assurance qualité peuvent valoriser les retours d'expérience pour optimiser la qualité des logiciels et s'assurer que les applications répondent parfaitement aux besoins métiers.

Les entreprises qui réussissent à promouvoir une telle collaboration créent généralement un poste d'architecte en automatisation pour tenir le rôle de développeur de tests. Ce dernier collabore avec les experts du domaine afin de comprendre les fonctionnalités métiers requises et les principes de l'application du point de vue de ses utilisateurs. L'architecte en automatisation peut alors concevoir le framework de test en conséquence pour s'assurer que l'application répondra aux besoins métiers et aux exigences fonctionnelles, de qualité et de fiabilité des utilisateurs finaux.

La collaboration peut également être encouragée en recherchant une plate-forme intégrée permettant d'étendre le processus qualité au-delà des tests fonctionnels à l'intégralité du cycle de vie de l'application et de diffuser les

préoccupations de qualité à chaque étape : de la conception initiale au déploiement et à la mise en production. Cette organisation stimule les communications trans-fonctionnelles en permettant à tous les intervenants de consulter et partager toutes les informations qualité stockées dans le référentiel de la plate-forme.

Gérer des applications en constante évolution

En matière de logiciels, les changements sont inévitables – qu'ils découlent de nouveaux objectifs et besoins des utilisateurs ou de l'apparition de nouvelles plates-formes techniques. En pratique, la maintenance logicielle requise pour faire face au changement représente plus de 70 % des coûts de développement.² Les entreprises performantes parviennent à limiter ces coûts très élevés en intégrant la maintenance au sein même de l'automatisation des tests. Par exemple, en créant des composants de test modulaires qui isolent et encapsulent les fonctionnalités de l'application afin de ne corriger que le/les composant(s) affecté(s) par la modification sans avoir à re-coder l'ensemble du script pour l'adapter simplement et rapidement aux nouveaux besoins applicatifs.

La simplicité de maintenance et de réemploi sont les deux principaux vecteurs de ROI de l'automatisation des tests. Pour pleinement en bénéficier, il est indispensable de choisir une solution d'automatisation dotée d'une architecture orientée objet permettant d'adopter toutes les approches d'automatisation précédemment mentionnées (modularité, architecture de bibliothèque, mots-clés, données, enregistrement/exécution) et surtout de considérablement réduire les délais et les efforts requis pour modifier les tests afin de prendre en compte des fonctionnalités nouvelles ou étendues. Cette option permet aux entreprises de suivre le rythme des modifications applicatives et de maintenir un niveau élevé de réemploi des ressources de test d'un build à l'autre.

Que rechercher dans une solution d'automatisation des tests fonctionnels ?

Un environnement d'automatisation conçu pour la productivité – Pour rester en phase avec l'accélération des cycles de mise sur le marché des nouveaux produits, les départements d'assurance qualité doivent s'appuyer sur une solution automatisée pour construire des ressources de test modulaires et réemployables. Au-delà de l'accélération des processus, la réutilisation des cas de test optimise la qualité logicielle en améliorant la prévisibilité des sessions de test.

Une adaptabilité suffisante pour répondre à la constante évolution des besoins métiers – De nouveaux besoins apparaissent littéralement du jour au lendemain et créent un besoin immédiat de réorganisation des applications. Serez-vous en mesure de rénover au même rythme les cas de test associés ? Il est pour cela indispensable de rechercher une solution automatisée exigeant une maintenance minimale. C'est pourquoi, en raison de ses caractéristiques inhérentes de emploi et de simplicité de maintenance, il est préférable de privilégier une plate-forme orientée objet permettant de modifier rapidement les scripts de test et de bénéficier de caractéristiques exclusives de portabilité et d'extensibilité pour englober de multiples technologies.

Un support technologique étendu – Les applications actuelles doivent satisfaire les utilisateurs d'une grande variété de plates-formes, environnement et technologies. Pour éviter d'acquérir de multiples produits de test, il est donc préférable de retenir une solution d'automatisation supportant tous les environnements de test et toutes les configurations système.

Une optimisation de la qualité tout au long du cycle de vie de l'application – Si les tests fonctionnels sont une priorité initiale, ils ne représentent pas l'ensemble des impératifs de qualité qui englobent de nombreux autres domaines : tests unitaires, de performance et de montée en charge ; administration des tests ; gestion des performances applicatives après la mise en production, etc.

Une base installée conséquente – Dans un contexte d'omniprésence des technologies, les applications peuvent faire ou défaire le succès d'une entreprise... C'est pourquoi il est primordial de limiter les risques en déployant des solutions performantes ayant fait leurs preuves auprès d'une importante communauté d'utilisateurs.

Des services et un support répondant aux besoins des utilisateurs – La mise en place d'une solution d'automatisation des tests exige des investissements initiaux en temps et ressources – qui sont toujours des « denrées rares » dans les environnements de développement logiciel. Il n'en est que plus impératif de choisir un éditeur capable de fournir des prestations de formation et de consulting à forte valeur ajoutée, de compléter les ressources internes en cas de besoin et de réussir l'implémentation d'un environnement de test performant et opérationnel.

Borland® SilkTest® – La solution d'automatisation des tests fonctionnels

Borland® SilkTest® est une plate-forme d'automatisation des tests fonctionnels reconnue pour son ouverture sur tous les environnements applicatifs d'entreprise : .NET, Web, Java™, client/serveur, etc. SilkTest permet en effet aux entreprises de valider que leurs applications répondent parfaitement aux objectifs métiers dans le cadre des impératifs actuels de brièveté des cycles de test grâce à une puissante technologie d'automatisation caractérisée par sa précision, sa cohérence et sa productivité. SilkTest supporte tous les principaux frameworks de test (modularité, architecture de bibliothèque, mots-clés, données, enregistrement/exécution) et assure également la fiabilité des applications multilingues (y compris pour les polices double-octet telles que le Kanji ou le chinois simplifié) en intégrant la solution de test fonctionnel Unicode® la plus performante de l'industrie.

Conçu dans un esprit de simplicité et de productivité, SilkTest maximise la vitesse et l'efficacité du processus de test. Les utilisateurs peuvent en effet concevoir des tests pour de multiples plates-formes, navigateurs et technologies avec le même script, en capitalisant sur la flexibilité et la robustesse du langage de script orienté objet de SilkTest (Borland® 4Test®). En outre, grâce à son système intégré et personnalisé de reprise, SilkTest permet d'exécuter des tests sans surveillance (24X7). Ainsi, en consacrant moins de temps aux tâches répétitives, les équipes d'assurance qualité peuvent déceler les erreurs plus tôt dans le processus de développement pour optimiser la qualité finale de leurs produits.

Au-delà de la validation intégrale des fonctionnalités applicatives avant la première diffusion, les utilisateurs de SilkTest peuvent également mesurer l'impact des nouveautés en réutilisant les tests existants. Ces ressources réemployables facilitent la maintenance en permettant aux utilisateurs d'accélérer la modification des tests pour rester en phase avec le rythme de changement des applications.

L'architecture produit de SilkTest permet aux entreprises de gérer économiquement les complexités de test de leurs applications. Sa technologie exclusive d'Agent fournit le contrôle et la granularité requis pour tester les scénarios complexes et imbriqués de véritables cas d'utilisation. En outre SilkTest Runtime (une version limitée de SilkTest) permet de distribuer les tests sur différents environnements et machines pour maximiser la productivité et réduire les coûts des activités de test.

Gestion de la qualité tout au long du cycle de vie

La solution globale de gestion du cycle de vie qualité de Borland permet aux entreprises de déployer des logiciels de haute qualité, de réduire les risques opérationnels et de maximiser le retour sur investissement. Dans le monde entier, de nombreuses entreprises – en particulier du classement Fortune 500 – font confiance à la gamme innovante de produits Silk pour protéger leurs niveaux de service, leur avance concurrentielle et la réputation de leur marque. La qualité ne tenant pas seulement au déploiement d'un bon outil, Borland propose également une gamme complète de services de consulting, de support et de formation pour garantir la réussite du déploiement d'un environnement d'automatisation des tests dans tous les contextes.

En complément de SilkTest, Borland propose également les solutions suivantes :

- **Borland® SilkCentral® Test Manager** pour gérer l'intégralité du processus de test.
- **Borland® SilkPerformer®** pour administrer les tests de montée en charge et de performance de niveau entreprise.
- **Borland® SilkPerformer® Component Test Edition** pour anticiper les test d'accès simultané aux composants logiciels distants.
- **Borland® SilkCentral® Performance Manager** pour contrôler les applications du point de vue des utilisateurs finaux.

Synthèse

Avec une planification et une stratégie adéquates de mise en œuvre, l'automatisation des tests fonctionnels peut considérablement optimiser la qualité logicielle en validant, dès la phase de pré-production, la précision et la fiabilité des fonctionnalités qui seront offertes aux utilisateurs finaux. L'automatisation des tests participe également à la création de nouveaux pôles d'efficacité en garantissant aux applications, même les plus complexes, un déploiement conforme aux budgets et délais impartis. Dans le monde entier, de multiples entreprises capitalisent sur ces avantages grâce à la solution primée par la profession Borland SilkTest qui leur offre une méthode reproductible et prévisible pour tester en continu les fonctionnalités de leurs applications de version en version. SilkTest inscrit la vitesse et la précision au cœur du processus de test et participe à l'optimisation de la qualité des logiciels en libérant les équipes pour d'autres tâches de contrôle et de validation afin de détecter proactivement les erreurs, le plus tôt possible au cours du processus de développement.