Retour au Dossier Organisationnel	
Sujets connexes	Cartographie fonctionnelle de LoGeAs Lancer les tests E2E (Playwright)

Procédure de test (procédure #11)

Informations qualité

Suivi des modifications majeures	oct. 2023 Passage des tests dans la carto fonctionnelle aout 2017 Migration sur DoKuWiKi et refonte complète pour v9 déc. 2015 - Nicolas Marchand - Création du document
Suivi des approbations	Ce document correspond à l'élément ProjeQtor Document # 11 -PROC-NFlog-10-Procédure de gestion des tests - ProjeQtOr
Objet	L'objectif de ce document est de définir les différents niveaux de tests du produit LoGeAs
Destinataires	**- Validation des modifications :** Gérant **- Approbation du document** : Equipe dev & Equipe Assistance

Liens rapides aux documents liés

A partir de septembre 2023 les tests d'interface sont gérés (définition, passage ...) dans la cartographie fonctionnelle
Accés à la liste des tests unitaires
Accés à la liste des tests internes

Les différents types de test

Tests d'interface

Les tests d'interface définissent des suites d'actions à réaliser sur le logiciel qui doivent produire un résultat donné. Ils se basent sur la cartographie fonctionnelle et sur les exigences de la norme NF logiciel comptable.
On distinguera trois types de tests d'interface :

- Les tests liés au respect des règles de la marque NF
- Les tests de non-régression
- Les tests spécifiques aux besoins d'un client

Ils sont définis dans directement dans la cartographie fonctionnelle et leurs réalisations y sont suivie directement.

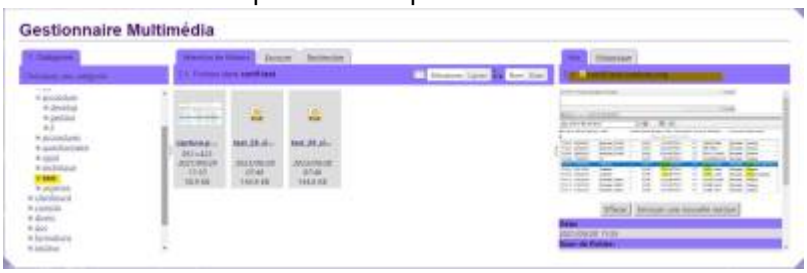
Qui passe ces tests

Les tests sont passés par l'équipe d'assistance (éventuellement renforcé par des “devs”, autre que celui qui aurait participer à un correctif lié récent)

Comment passer ces tests

1. Ce connecter à la cartographie fonctionnelle
2. Aller dans l'ergo “Univers test”
3. Choisir le test à passer

Cas d'un test passé pour la première fois

1. Vérifier que le déroulé du test est compréhensible, correctement paginé et que les éventuelles pièces nécessaires sont jointes.
2. Si des pièces sont manquantes les ajouter dans le gestionnaire de fichier du wiki dans le dossier test et récupérer l'adresse en cliquant sur la pièce en haut à droite faire ensuite un lien dans le corps du test. 

3. Vérifier que le lien est fait avec **toutes** les fonctionnalités touchées dans le test (Cette vérification est à faire dans la carto “Vues\Fonctions/Tests”, le cas échéant l'ajouter)

Déroulement du test

1. Suivre **a la lettre** la procédure indiqué
2. Si on arrive au résultat escompté ajouter le résultat en cochant la case OK,
3. dans la cas contraire :
    1. créer un ticket projeqtor indiquant le test et le problème rencontré
    2. ajouter le résultat en décochant la case OK et en indiquant dans le commentaire le numéro du ticket projeqtor

Tests fonctionnels

Les tests fonctionnels on pour but de tester de manière automatique des grandes fonctionnalités du logiciel : tests de non-régression des fonctionnalités de haut-niveau (ex. : génération) en se basant sur des jeux de données validés et en comparant le résultat de l'exécution de la fonction avec les données de référence. S'il y a des différences, elles doivent être expliquées (changement de comportement, correction d'un bug…) Les procédures de test sont stockées dans le sous-dossier certif:test:testfonction **A ce stade nous n'avons pas trouvé de logiciel (abordable pour notre petite structure) nous permettant d'automatiser les tests, nous avons donc préféré travailler uniquement sur les deux autres types de test.**

Tests Unitaires (non remis en version WEB)

Tests automatiques des grandes fonctions internes. Ces tests sont encodés dans un programme parallèle en utilisant le framework DUNIT.
Nous utilisons ce framework de test à code source libre, pour le développement et l'exécution de cas de test automatisés pour nos applications.
Le framework DUnit est disponible pour Delphi et C++. Ce framework simplifie le processus de développement de tests pour les classes et méthodes de nos applications.
L'utilisation de tests unitaires permet d'améliorer la stabilité de nos applications. Tester un ensemble standard de tests à chaque fois qu'une petite modification est effectuée à travers le code permettra de voir les problèmes tôt dans le cycle de développement et donc de les corriger.
Pour en savoir plus

Accés à la liste des tests unitaires

Tests Internes

Le logiciel réalise des tests internes à divers moments clefs.
Ces tests ont pour but d'aider l'utilisateur :

- à la bonne utilisation du logiciel,
- à la conformité de sa comptabilité aux règles générales, ou spécifiques à son regroupement

mais aussi à détecter des problèmes internes lors des phases initiales de saisie. Certains problèmes sont à corriger par l'utilisateur, d'autres impliquent un contact avec l'assistance.

Accés à la liste des tests internes

Qui écrit et qui passe les tests ?

La personne qui code le correctif n'est pas celle qui teste (on parle bien ici des tests finaux et répétitifs).
C'est la personne qui décrit le bug (s'il est important) qui met en place le test (et le passe une première fois).

On trouvera ci-dessous les éléments de procédures à suivre sous forme d'exemples

Bug découvert ou remonté

- Le bug est décrit, par Mlle. K., si celui-ci est critique, elle met en place un test et un jeu de données afin de pouvoir tester le résultat dans ProjeQtOr sur le ticket numéro 52, il est alors affecté au responsable du pôle developpement M. G., qui le gère et le fait traiter par M. V. .
- A la fin du traitement M. V. pensant qu'il n'y a plus d'erreur 500, affecte le ticket en mode “A tester” à Mlle. K.
- Celle-ci explicite dans l'intranet le test à réaliser avec éventuellement le jeu de données et passe une première fois le test pour vérifier. Si le test passe alors elle passe le ticket en “validé”. Si elle obtient une erreur 500 elle ré-affecte le ticket à la personne de son choix

- Dans le cas où le test est OK elle avertit le responsable des tests M. N. afin que ce test soit intégré dans le SIGdT et qu'il puisse être réalisé quand une des unités ascendantes est modifiée

Réalisation des tests avant livraison d'une version

- Une beta Vx est générée.
- Le SIGdT est exécuté par le responsable des développements M. G. SIGdT vérifie la liste des unités modifiées depuis la dernière release, il en déduit la liste des unités interfaces qui doivent être re-testées et donc la liste des exigences (au sens ProjeQtOr) qui risquent de ne plus être satisfaites.
- Il fait parvenir la liste à M. N. en charge des tests qui les repartit entre Mlles. K., M. et lui-même.
- Si un test n'est pas OK un ticket de bug est émis et affecté à M. G.
- Si tout les tests sont OK, M. G. est avertit par M. N. qu'il peut basculer la beta Vx en release Vx, sinon il corrige et sort une Vx.y et reboucle au début de ce paragraphe.

From:
https://wiki-logeas.fr/certif/ - **dokuwiki-certif**

Permanent link:
https://wiki-logeas.fr/certif/doku.php?id=certif:procedure:test&rev=1790082110

Last update: **2026/09/22 15:01**