

Justin VILLEROT

Mise en place de tests dans un logiciel de contrôle qualité.

Rapport Technique

Stage au sein de Qualigest
Du 3 avril au 24 juin 2023

Maître de stage : Nicolas CAZETTE
Enseignant Tuteur : Corinne PATERLINI

Sommaire

Introduction.....	4
1) Présentation générale.....	5
1.1 Outils Utilisés.....	5
1.2 Fonctionnement d'un test.....	5
1.3 Plus de documentation.....	6
1.4 Autres options.....	6
1.5 Pourquoi ce choix.....	7
2) Comment faire & Difficulté.....	9
2.1 Les différentes fonctions tests.....	9
2.2 Enregistrer un test.....	10
Conclusion.....	13
Table des figures.....	14
Sources.....	15

Introduction

Le but de ce projet était de mettre en place une automatisation des tests fonctionnelle de l'application. L'objectif était de pouvoir complètement automatiser ces derniers, sans parasiter le code du projet, ainsi que de documenter le processus d'automatisation, afin qu'ils puissent être reproduits sur d'autres projets.

L'objet de ce rapport est donc de faciliter la reprise de l'automatisation par d'autres développeurs, en expliquant comment elle a été faite et comment elle peut être reproduite.

Premièrement, sera présenté quels outils ont été utilisés, comment ils fonctionnent. Puis comment est fait un test plus en détail, et des difficultés qui peuvent être rencontrées.

1) Présentation générale

1.1 Outils Utilisés

Pour mettre en place les tests, l'outil de test automatique de Windev a été utilisé. Il a été choisi, car il permet d'avoir plus d'information sur l'état de l'application, et il permet de continuer à utiliser le même IDE que l'entreprise, et donc par extension le même langage de programmation.

Cependant, il reste certaines choses qui ne peuvent pas être testées avec cette méthode, comme le contenu des fichiers exporté ou l'impression d'un document.

Mais l'utilisation d'un logiciel externe comme un bot clavier souris aurait compliqué la vérification de la base de données. Une solution aurait été de développer nous-même des bots qui feraient chaque test, mais le processus aurait été beaucoup plus long et fastidieux. Qui plus est certaines informations uniquement obtenables via Windev serait introuvable.

1.2 Fonctionnement d'un test.

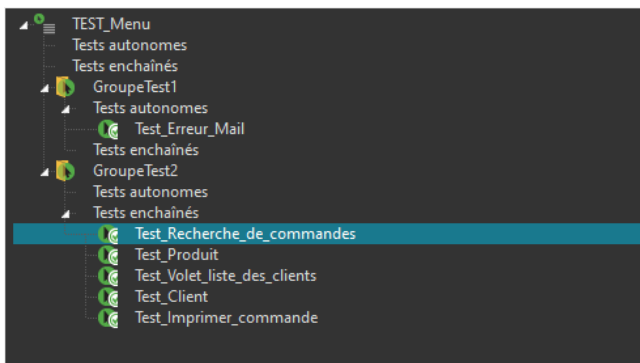


Figure 1: Arborescence Type

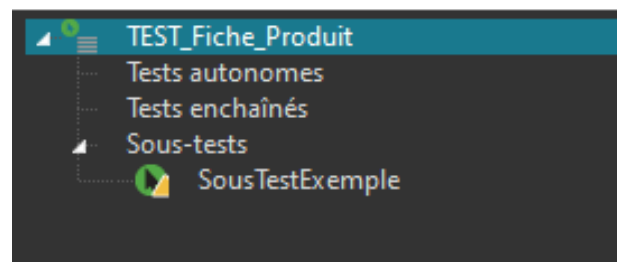


Figure 2: Arborescence Type 2

Premièrement, existe 3 types de test, comme on peut le voir ci-dessus :

Les tests autonomes, à la fin de chaque test l'application va se fermer. Cela permet de tester des choses qui pourraient être changées par des précédents tests, comme les droits d'utilisateurs, quel utilisateur est connecté...

Les tests enchaînés, ils s'effectuaient les uns après les autres sans fermer l'application. Donc les actions des précédents tests peuvent influencer le test précédent. À noter que dans tous les cas les changements sur la base de données sont répercutés sur les prochains tests.

Les sous-tests sont des tests enchaînés qui ne peuvent être exécutés qu'à la suite de l'appel de la fonction `PiloteFenêtre` qui va appeler un sous-test lorsqu'une fenêtre définie est ouverte.

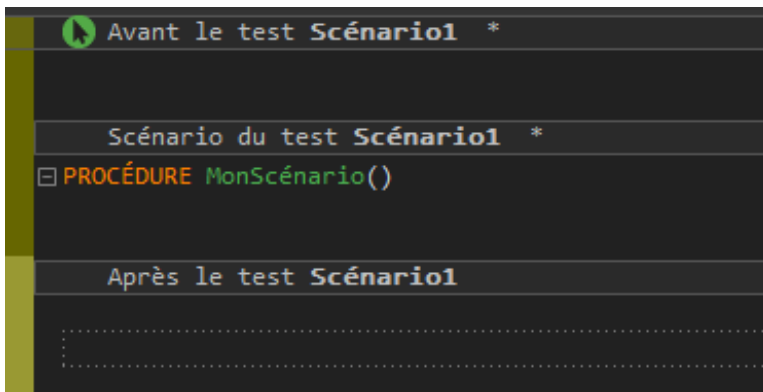


Figure 3: Découpage d'un test

Ensuite, comme on peut voir ci-contre un test, est découpé en trois parties.

Avant le test, cette partie sert généralement à créer des nouveaux enregistrements dans la base de données qui serviront au test, ou a configuré des paramètres spécifiques comme un paramètre lié à une entreprise dans le cas de QualiSPC.

Le scénario est la partie où le test va effectuer des actions sur l'application.

Après le test, dans cette partie, on peut vérifier que les choses se sont bien effectuées comparer les valeurs de la BDD avec des calculs...

À noter qu'il existe aussi une partie avant tout les tests et après tous les tests.

De plus les différentes parties d'un test s'exécute dans un ordre spécifique :

- Initialisation du projet
- Initialisation pour tous les tests
- Initialisation Groupe test
- Avant Test
- Initialisation Page
- Code scénario (action sur page)
- Après Test
- Fin Groupe
- Fin Test
- Fin Application.

L'ordre des parties de test permet de comprendre, on peut interférer sur l'initialisation des différents éléments de l'application depuis les différentes définitions de variable des tests. Cependant, il n'est pas possible de modifier des choses sur l'initialisation de l'application.

1.3 Plus de documentation

- [Doc PCSoft – Présentation test automatique](#)
- [Doc PCSoft - Gestionnaire de tests automatiques](#)

Le second lien offre une explication plus complète du fonctionnement des tests.

1.4 Autres options

Une autre solution aurait été d'utiliser un bot déjà existant ou d'en créer un nous-même. Cela auraient permis de tester l'impression et l'export de fichier plus en profondeur, car l'on aurait pu

faire des actions sur des fenêtres générées automatiquement sur Windev ou sur des fenêtres d'application hors Windev comme LibreOffice Calc ou une visionneuse de PDF.

1.4.1 Bot clavier souris existant :

Il existe plein de bots qui permette d'enregistrement des actions clavier souris. Par exemple :

- Power Automate de Microsoft [Lien](#)
- TinyTasks [Lien](#)
- Et bien d'autres [liens](#)

Le problème de ces enregistreurs est qu'ils ne permettent pas d'avoir de retour sur le bon fonctionnement de l'application. Ils sont conçus pour enregistrer des macros, ou des actions effectuer, mais pas vérifier si les conditions d'origine sont respectées. Ils ne permettent donc pas de savoir si une fenêtre a bien été ouverte ou si un quelque chose a bien été enregistré dans la BDD.

Pour obtenir ces informations, il faudrait créer des scripts qui vont lancer les macros prés enregistrer et faire des vérifications sur la base de données ou sur l'application. Cependant, comme nous n'avons accès qu'à l'aspect graphique de l'application, il peut être compliqué de faire des vérifications dessus.

De plus, les bots ne pourront pas détecter si la fenêtre n'est pas exactement au même endroit que l'enregistrement, ou si un élément ont été déplacés.

1.4.2 Bot clavier souri personnalisé :

Les problèmes amenés par l'utilisation d'un bot préexistant nous amène à l'idée que créer un bot de A à Z sera plus profitable. Si on veut avoir des informations sur le bon fonctionnement de l'application, et de sa BDD. Bien que l'accès à la BDD puisse assez facilement être fait si la base de données et présente sous forme de fichier local, cela peut se compliquer si la BDD est dans un serveur distant.

Mais le plus compliquer reste de savoir si nos actions sur l'application ont bien les effets escomptés. Pour faire cela, on pourrait faire une capture de l'écran et essayer de détecter si on peut reconnaître l'interface de l'application ainsi que le texte qui est écrit dessus. Des bibliothèques telles que [pytesseract](#) ou [C# OCR Image to Text](#), pourraient être utilisés.

Même en faisant tout cela ne nous nourrissait toujours pas accès à des informations qui sont uniquement accessibles via Windev, comme les variables globales de l'application ou l'état d'un champ invisible...

1.5 Pourquoi ce choix

Utiliser les outils fournis par Windev permet dans un premier temps de conserver les outils et langages déjà utilisés par l'entreprise.

Mais plus important est le fait que les outils de Windev permettent d'avoir beaucoup plus d'information sur l'état des différents éléments de l'application. De plus utiliser d'autres outils comme un bot pour simuler le clavier et souris aurait compliqué l'accès et la vérification de la BDD.

Il aurait été possible de faire de zéro un bot qui peut accéder à une BDD, mais cela aurait long est aurait demandé un plus long apprentissage, car il aurait fallu apprendre un nouveau langage et complètement codée chaque test.

Les quelques fonctionnalités gagnées par le fait de ne pas utiliser les outils Windev sont bien moins intéressantes que les informations et la praticité apportée par les outils Windev.

2) Réalisation d'un test

2.1 Les différentes fonctions tests

Dans Windev, il y a plein de fonctions qui sont spécifiques au test automatique.

[Doc – PCSoft Fonctions spécifiques aux tests](#)

La plupart des fonctions dans la documentation ci-dessus sont des fonctions qui permettent de vérifier la valeur de variable ou de faire des logs de test.

Les logs de test sont similaires aux erreurs de Windev, mais elles sont en plus de la fenêtre des erreurs répertoriées dans une autre fenêtre, comme ci-dessous.

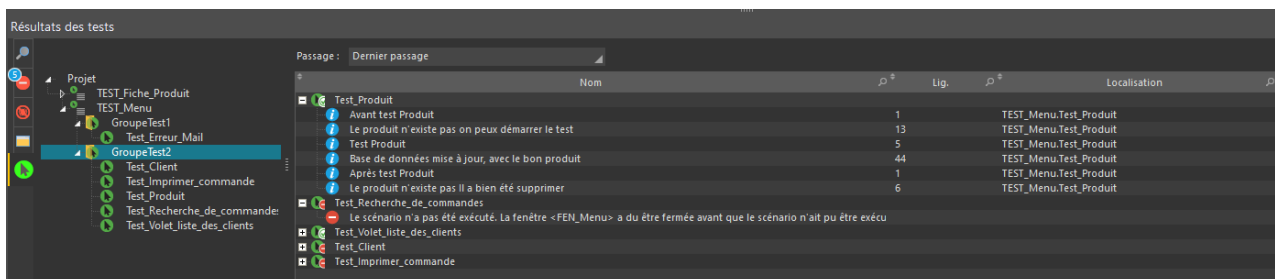


Figure 4: [Volet Résultat des Tests](#)

2.1.1 Les pilotes

Les fonctions les plus intéressantes ajoutées pour le test automatique sont sûrement les fonctions pilotes. Elle permet de contrôler le clavier et la souris ainsi que les différents éléments de l'interface de Windev comme les volets, le menu ou les combos.

Pour plus d'information se référer à la documentation ci-dessous

[Doc – PCSoft Fonctions de pilotage](#)

2.1.2 La transition entre les fenêtres.

Pour passer d'une fenêtre à une autre, il y a deux méthodes :

Utiliser le mot réservé SUR, ce qui permet de garder tout le code dans un seul fichier, mais cela peut causer des problèmes dans certains cas comme lorsqu'une fenêtre est ouverte lors de l'initialisation de la fenêtre de lancement de l'application (comme avec Fen_principale dans QualiSPC). Pour plus d'information lire la documentation [PCSoft](#).

Ensuite, il est aussi possible d'utiliser la fonction PiloteFenêtre qui va lancer un sous-test défini lorsqu'une fenêtre est ouverte. Comme les autres pilotes qui réagissent à des événements, ils doivent être positionnés avant l'action qui ouvre la fenêtre, cela peut permettre de faire des choses que l'on ne pourrait pas faire avec le mot réservé SUR, car cela permet d'interagir entre les différentes parties d'un test.

2.2 Enregistrer un test.

Pour faire un test, on peut enregistrer une série d'actions sur l'application. Ensuite, l'enregistrement génère un code qui permet de reproduire les actions précédemment effectuées. Ensuite, il faut retravailler ce code si on veut ajouter des vérifications sur des valeurs ou imprimer des choses dans la console. Il faut aussi modifier certaines choses, car le code généré par l'enregistrement n'est pas parfait. Par exemple, il faut remplacer les SUR Erreur Retour par la fonction `PiloteErreur()` qui doit être appelée avant l'action qui va provoquer une erreur...

Aussi, il peut être intéressant de faire les parties avant et après test et de lancer une fois le test à vide, afin de pouvoir enregistrer directement sur les données de test.

2.2.1 Test Enregistré par l'utilisateur

En plaçant un fichier WX nommé `nomexecutable.WX` dans le répertoire de l'exécutable d'une application Windev, on peut enregistrer différentes choses qui se passent durant l'application. Il est aussi possible de mettre des fonctions dans le code qui lanceront l'enregistrement, comme pour les statistiques collectées durant l'exécution qui est actuellement dans le code.

Voilà un exemple de fichier :

```
[LOG]
Actif=1
Fichier=[%NomExe%]LOG_D[%Date%]_T[%Heure%]_U[%NomUtilisateur %]_M[%NomMachine%].wlog
Option="*"

[Scenario]
Actif = 1
Fichier=[%NomExe%]TEST_D[%Date%]_T[%Heure%]_U[%NomUtilisateur %]_M[%NomMachine%].wsct
CheminDLL = ".\"
```

Ce fichier va générer un fichier LOG qui enregistrera quelle fonction fenêtre et variable sont appelées quand elles le sont et plein d'autres informations. La partie scénario va générer un test et des sous-tests. (test qui ne peut être lancé que par un autre test généralement pour faire des actions dans d'autres fenêtres que la fenêtre principale du test.)

Les avantages de ces tests et qu'ils sont mieux générés que ceux faits en utilisant la fonction enregistrer de Windev, mais cela reste toujours des tests automatiques.

De plus, simplement, en copiant un fichier dans le répertoire de l'exécutable, il est possible de générer beaucoup de choses (voir ci-dessous) ce qui permet de le faire même sur une configuration cliente.

Autrement, ce fichier permet de gérer :

- [la désactivation du JIT](#). (Just In Time Compilation)
- [la gestion des tests automatiques utilisateur](#).
- [l'audit dynamique](#).
- [le log d'exécution](#).
- [les connexions utilisées par l'application](#).

[Plus d'info](#)

Il est à noter que cette méthode d'enregistrement génère un code qui va préférer utiliser la méthode Pilote fenêtre au mot réserver SUR. Il va sont créés plein de fichier qui contiendront des sous-tests.

2.2.2 Test Enregistré avec Windev

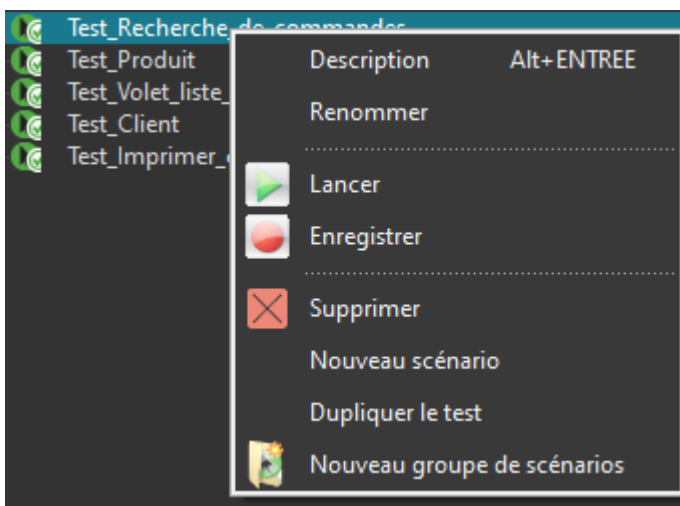


Figure 5: Enregistrement test

En cliquant sur le bouton enregistrer dans le menu contextuel d'un scénario de test, il est possible d'enregistrer un test.

Cet enregistrement, à l'instar de l'enregistrement par l'utilisateur va enregistrer sur Go et non sur l'exécutable.

À la fin de l'enregistrement, le scénario sera rempli.

Noter que comme dit précédemment cette méthode utilise uniquement le mot réservé SUR. Ce qui peut causer des problèmes, notamment avec les pop-ups

d'info et d'erreur qui peuvent nécessiter l'utilisation du pilote erreur ou info.

2.2.3 Lancement des tests

2.2.3.1 Lancement sur Go

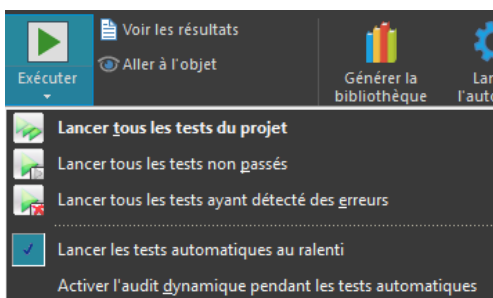


Figure 6: Lancement des tests

Il est possible de lancer les tests individuellement ainsi que par groupe de test, ainsi que tout d'un coup. Comme on peut le voir sur l'image ci-jointe, il est possible d'exécuter les tests au ralenti ou non. Cela permet de mieux voir où sont les problèmes.

Les tests lancés de cette manière seront effectués sur l'équivalent d'un Go et non sur l'exécutable de l'application.

Il est à noter que les points d'arrêt du débogueur ont tendance à faire planter les tests. Le problème est qu'il déplace le focus sur le code donc les actions qui auraient dû être effectuées dans un champ sont effectuées dans le code. En conséquence ATTENTION cela peut rajouter des bugs de compilation dans le code.

2.2.3.2 Lancement sur un exécutable

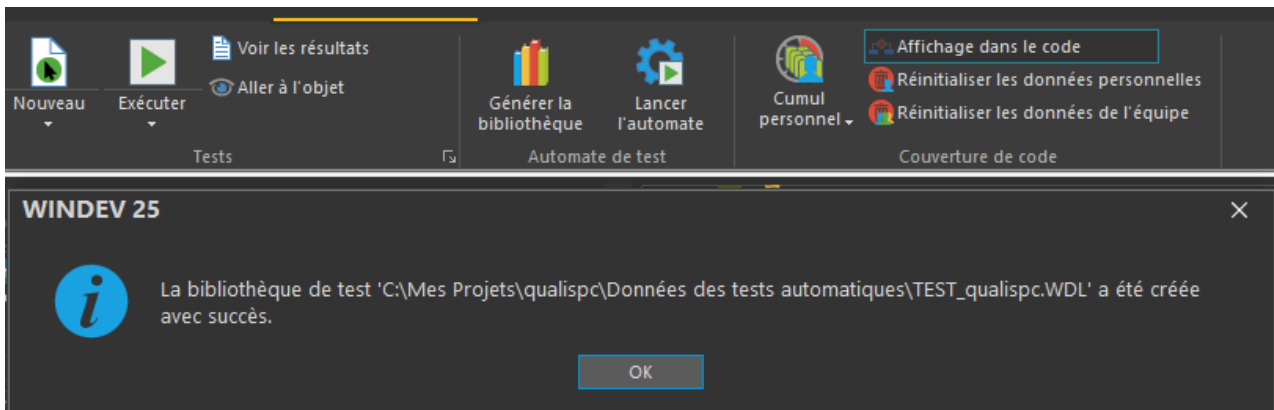


Figure 7: Génération de la bibliothèque de test et lancement de l'automate

Il est aussi possible de lancer les tests directement sur un exécutable. Pour ce faire, il faut d'abord générer une bibliothèque de test comme montré sur l'image ci-dessus.

Ensuite, on peut lancer les tests sur l'exécutable en cliquant sur lancer l'automate, cela va ouvrir une fenêtre où on peut paramétrer l'emplacement de l'exécutable et de la bibliothèque qui seront testés.

Il est à noter que seul le résultat du test est affiché pas le détail. L'automate n'est donc pas très pratique pour régler les problèmes qui peuvent apparaître.

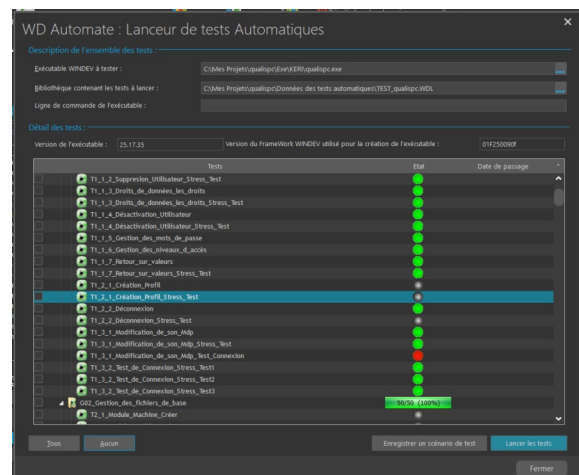


Figure 8: Fenêtre de configuration de l'automate de test.

2.2.4 Résultat des tests

Lorsque les tests ont été effectués les résultats s'afficheront dans le [Volet Résultat des Tests](#).

Noté que s'il y a un grand nombre de tests l'affichage des résultats peut prendre du temps et l'application n'est pas utilisable pendant que les résultats chargent.

Conclusion

Bien que les outils de test automatique fourni par Windev soit limité aux applications Windev et ne puissent pas être utilisé sur des logiciels extérieurs tels que Excel ou une visionneuse de PDF. Les avantages qu'ils fournissent sont bien plus nombreux que les inconvénients.

C'est pour cela que je pense que l'utilisation ou la conception d'un bot extérieur n'est pas intéressante, car on perd l'accès à beaucoup d'information qui peuvent rendre le test plus complexe. Comme les variables globales de l'application ou l'état de l'interface qui est invisible comme les champs cachés et autre.

Table des figures

Table des Figures

Figure 1: Arborescence Type.....	5
Figure 2: Arborescence Type 2.....	5
Figure 3: Découpage d'un test.....	6
Figure 4: Volet Résultat des Tests.....	9
Figure 5: Enregistrement test.....	11
Figure 6: Lancement des tests.....	11
Figure 7: Génération de la bibliothèque de test et lancement de l'automate.....	12
Figure 8: Fenêtre de configuration de l'automate de test.....	12

Sources

Doc PCSoft

- [Tests automatiques : Présentation](#)
- [Gestionnaire de tests automatiques](#)
- [Test automatique créé par l'utilisateur](#)
- [Fichier WX](#)
- [Tests automatiques : Fonctions de pilotage](#)
- [SUR \(Mot réservé\)](#)
- [Tests automatiques : Fonctions spécifiques aux tests](#)

Autre

- [Forum PCSoft](#)
- [Power Automate](#)
- [Enregistreur de souris / clavier](#)
- Bibliothèque Image vers texte [Python](#) / [C#](#)