

Les différents types de test que j'ai fait  
Les problématiques que j'ai (ce qu'on attend demain)  
regroupe tout ce qu'on m'a demandé jusqu'à maintenant

Sur toutes les nouvelles fonctionnalités de nouveaux tests à automatiser.

2 parties : Tests automatisés  
Pipeline Azure DevOps

Améliorer la pipeline pour

A quoi ça sert  
Où tu en es  
Ce qui reste à faire

Expliquer ranorex et son intérêt pour l'équipe

Permet d'automatiser des parcours utilisateurs, simulation de personne qui travaille sur le produit

Test du fonctionnement interne de Digicompta

Test case contient des records qui sont des actions

A partir de quoi

Test d'une fonctionnalité

Test de sorties

Quel type de test (Ranorex\_Liaison) : Test de création avec différentes versions de la compta et de la GC.

Particularité de DigiCompta par rapport aux autres? Dev par Hubbix / diff: principe communiquer avec la ComptaOL et DigiCompta : sys d'authentification (problème donc j'ai dû simuler les API Digicompta

Tester que les synchro se lance bien

Outil de gestion de projet

**Au cours de ma formation de deux ans au sein de l'entreprise EBP, A cegid company occupant le poste de Développeur de Tests automatisés j'ai pu aider mon équipe notamment;**

- **Sur la gestion du temps de la QA (grâce aux nombreux tests mis en place**

**par mes soins ce qui leur a permis d'avancer sur d'autres sujets comme les périodes fiscales, etc.)**

**Au cours de la formation les différentes compétences que j'ai pu acquérir sont les suivantes:**

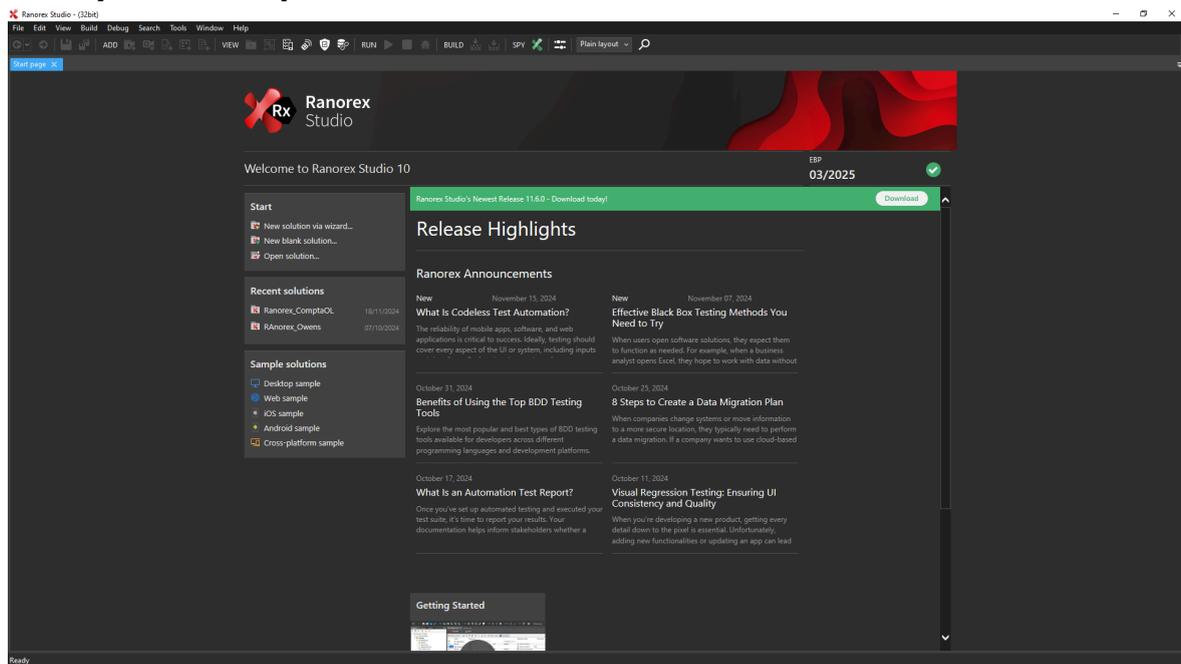
- **Un esprit d'analyse et de synthèse accru**
- **L'esprit d'équipe : certaines missions me poussaient à demander de l'aide au sein de l'équipe dans laquelle j'étais, auprès des développeurs comme auprès de la QA ( Équipe qualité) ou même de mon manager.**
- **Réactivité**
- **Respect des procédures**
- **Connaissance**
- **Adaptabilité**
- **Gestion des priorités : Sur Azure Devops nous avons une partie backlog, où nous remplissons nos projets respectifs (Epic), à l'intérieur différentes grandes étapes (Features), et plus petits objectifs mais qui compose les features (US - User Stories) avec des indices de priorités allant de 1 à 4.**
- **Reporting d'activités : Avec les daily que l'on faisait ceux-ci consistait à faire un résumé de sa matinée et ce sur quoi on se lançait dans l'après-midi.**

**Ranorex Studio est une plateforme de test automatisé complète, conçue pour les tests fonctionnels et la validation des applications logicielles sur différentes plateformes (applications de bureau, mobiles, web). Ce logiciel est très apprécié dans les environnements de développement Agile et DevOps en raison de sa capacité à automatiser les tests de manière rapide et efficace.**

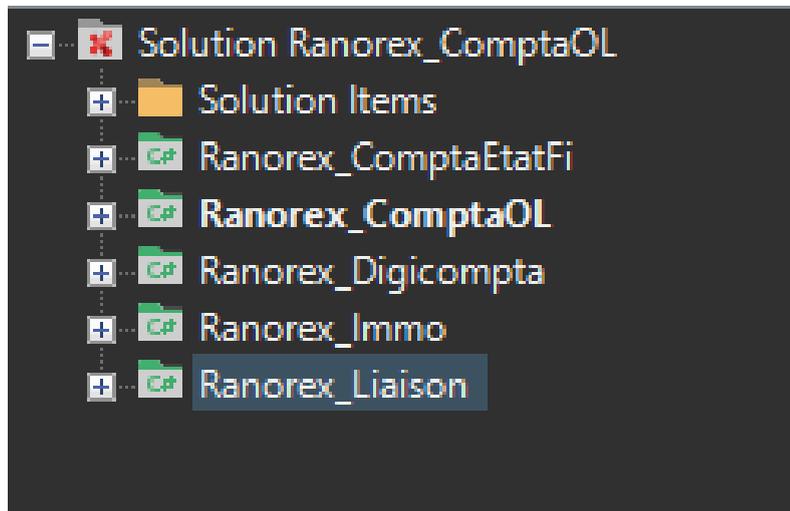
**Dans mon cas, Ranorex a été utile pour la réalisation de mes tests automatisés J'ai pu mettre en place une solution de tests automatisés regroupant d'autres tests que j'avais mené dans mes débuts sur la plateforme.**

**Il ne me reste plus qu'à corriger certaines étapes de certains tests car il peu y avoir des changements au niveau de l'interface utilisateur et un autre projet de**

## test, Synchro banque.



**Grâce à cette plateforme j'ai pu mettre en place une solution qui regroupe 4 projets de tests, le nom de cette solution est Ranorex\_ComptaOL .**



**Cette solution comporte les projets de tests suivant:**

- **Ranorex\_ComptaEtatFi, comme dit dans le nom, deux cas de tests dans un projet de test, Ranorex\_Compta et Ranorex\_EtatFi. Tous les cas de tests viennent d'un test plan en particulier, en l'occurrence pour**

**Ranorex\_Compta, Test plan TNR Compta et pour Ranorex\_EtatFi, Test plan TNR Etat Fi (Test de sorties), Avant cela, ces tests de sorties était lancés manuellement. En les automatisant c'est à peu près 1h30 de gagner.**

Item	Data binding / iterations	Description
Ranorex_Compta - Test suite		
SmartFolder	SmartFolder_DataSource Rows: 3 (Filtered: [3])	
[SETUP]		
Test_Dossier_Demo		
Test_Creation_Dossier		
Test_Aide_En_Ligne		
Test_Reformater_Compte		
Test_Facture_Vente_Echeance		
Test_Facture_Echeance		
Test_Avoir		
Test_Lettre		
Test_Rapprochement_Bancaire		
Test_Ajouter_Banque		
Test_DeclarationTVA		
Test_ValidationEcritureLoiAntiFraude		
Test_ClotureAnnuelle		
Test_Supprimer_Dossier		
Test_Restauration		
Test_Supprimer_Dossier		
[TEARDOWN]		

Item	Data binding / iterations	Description
Ranorex_EtatFi - Test suite		
Test_EtatFi	Test_EtatFi_DataSource Rows: 3 (Filtered: [3])	
[SETUP]		
StartAUT	Bound variables: 2	
Test_Dossier_Demo		
Ouvrir_Demo		
Test_Creation_Dossier		
Ouvrir_MenuBurger		
Nouveau_Dossier		
Creation_Dossier_EF	Bound variable: 1	
Test_Aide_En_Ligne		
Page_Id_Societes		
Test_Dossier_SIREN		
Ajouter_Dossier_SIREN		
Creer_Dossier_SIREN		
[TEARDOWN]		
CloseAUT	Bound variable: 1	

**Aujourd'hui les tests de sorties sont opérationnels et sont lancés régulièrement.**

Seulement, quelques modifications peuvent être apportées aux tests à chaque nouvelle version des logiciels.

- **Ranorex DigiCompta a une chose que les autres tests n'avaient pas. L'API a été une particularité propre à DigiCompta. J'ai dû en créer une car il était nécessaire pour la validité des tests. Avec l'aide de YVO et les devs j'ai pu la mettre en place. Aujourd'hui, Ranorex DigiCompta est en place et opérationnelle. Comme pour Ranorex ComptaEtatFi, je peux être amené à faire des changements sur ces tests à chaque nouvelle version.**

Item	Data binding / iterations
✓ Ranorex_Digicompta - Test suite	
- ✓ DigiCompta_TestPlan	
+ [SETUP]	
+ ✓ Synchronisation_ExerciceCompta	
+ ✓ Synchronisation_ComptesCompta	
+ ✓ Synchronisation_ComptesAux	
+ ✓ Synchronisation_ComptesAux_2	
+ ✓ Synchronisation_LongueurComptesA...	
+ [TEARDOWN]	

Item	Data binding / iterations
✓ Ranorex_DigiCompta2 - Test suite	
- ✓ Test_Plan2	
+ [SETUP]	
+ StartDigiCompta2	Unbound variable: 1
+ Reformatage_ComptesGen	
+ Synchro_ComptesAux	
+ TEST_Acces_Rapide	
+ Plan_Postes_Analytiques_KO	
+ Creation_Rapport	
+ Synchro_ComptesGen_Modif	
+ Creation_Rapport_Synchro	
+ Synchro_CmptCompta	
+ [TEARDOWN]	
+ CloseDigiCompta2	Unbound variable: 1

Visual Studio 2022

Ouvrir les éléments récents

Plus ancien

- Projet\_API.sln C:\...\Ranorex\RanorexStudio\_Projects\Ranorex\_ComptaOL\Ranorex\_Digicompta\API 22/10/2024 10:20
- Ranorex\_ComptaOL.sln C:\...\dylan.lernout\Documents\Ranorex\RanorexStudio\_Projects\Ranorex\_ComptaOL 07/10/2024 11:39
- API C:\...\Ranorex\RanorexStudio\_Projects\Ranorex\_ComptaOL\Ranorex\_Digicompta 07/10/2024 10:20

## - Ranorex\_Immo

Item	Data binding / iterations	Description
☑ Ranorex_Immo - Test suite		
└─ [SETUP]		
└─ StartAUT	Bound variable: 1	
└─ Creation_Dossier		
└─ Ajout_Immo		
└─ Ajout_Immo	Unbound variable: 1	
└─ Plan_Immobilisation		
└─ Ajout_Contrat		
└─ Ajout_Contrat		
└─ Sauvegarde_Restauration		
└─ Sauvegarde		
└─ Suppression_Dossier		
└─ Restauration		
└─ [TEARDOWN]		
└─ Suppression_Dossier		
└─ CloseAUT	Bound variable: 1	

## - Ranorex\_Liaison

Item	Data binding / iterations	Description
☑ Ranorex_Liaison - Test suite		
└─ ☑ TestCase		
└─ [SETUP]		
└─ StartLiaison	Bound variable: 1	
└─ Test_Création		
└─ Creation_DossGC	Bound variables: 2	
└─ Ouverture_Compta		
└─ Compta_Lies	Bound variables: 3	
└─ Test_Création2		
└─ Creation_DossCompta	Bound variable: 1	
└─ Creation_GC	Bound variable: 1	
└─ Liaison		
└─ Test_Création3		
└─ Doss_Lies_Compta	Bound variable: 1	
└─ Test_Création4		
└─ Doss_Lies_GC	Bound variables: 2	
└─ Test_Migration5/6		
└─ Restauration_Migration		
└─ Migration6		
└─ Test_Migration7		
└─ Migration_GC		
└─ Test_Migration8		
└─ Migration_Compta		
└─ Test_Migration9		
└─ Migration_ComptaGC		
└─ Test_Migration10		
└─ Migration_GCCompta		
└─ [TEARDOWN]		
└─ CloseLiaison	Bound variable: 1	

**Le but de ce projet était d'automatiser le plus de tests récurrents possibles pour**

que la QA ait plus de temps à consacrer aux choses plus importantes.

Pour aller dans le détail, le projet API pour Ranorex Digicompta a été codé en C# avec l'aide de l'infrastructure de développement web ASP.NET, sa version open source.

Pour pouvoir uniquement tester le côté ComptaOL et pour pouvoir éviter l'authentification j'ai dû simuler les API compta.

Voici le swagger de l'API

Projet\_API 1.0 OAS3  
http://localhost:5117/swagger/v1/swagger.json

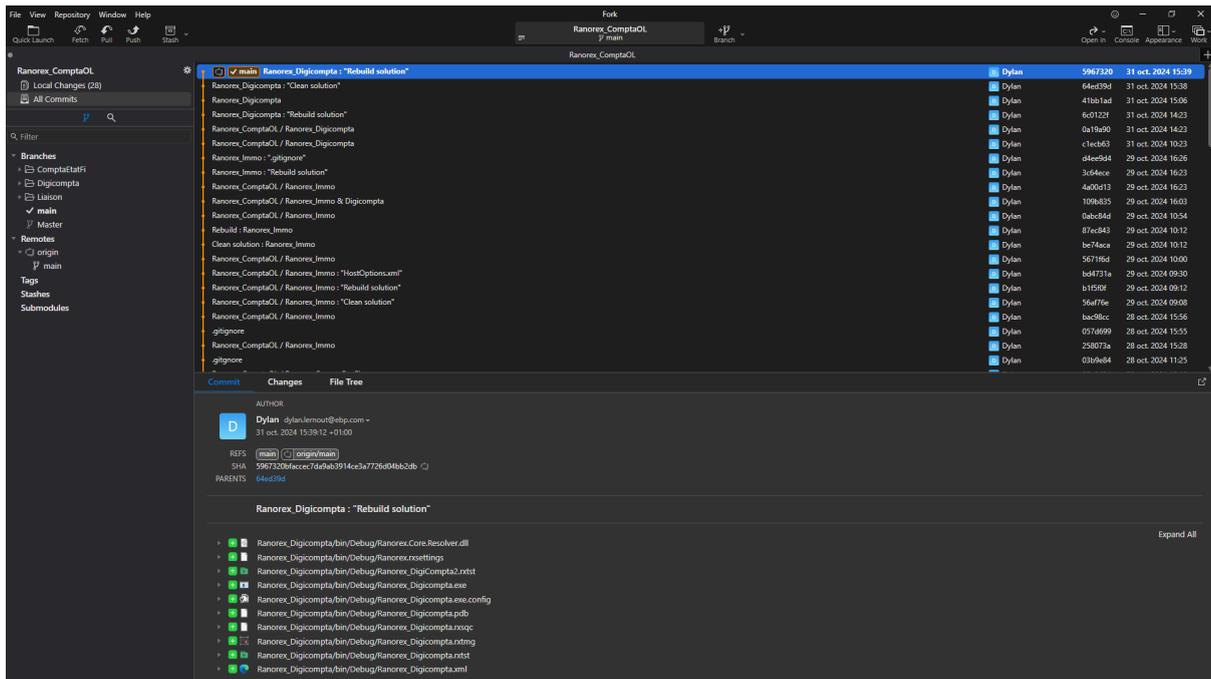
- Account**
  - POST /api/comptes-generaux
  - PUT /api/comptes-generaux
  - GET /api/comptes-generaux
- AccountingYears**
  - POST /api/exercices-comptables
- Activation**
  - POST /api/Activation
- AnalyticGrid**
  - POST /api/grilles-analytiques
  - GET /api/grilles-analytiques
  - PUT /api/grilles-analytiques
  - DELETE /api/grilles-analytiques
- AnalyticPlan**
  - POST /api/plans-analytiques
  - GET /api/plans-analytiques
  - PUT /api/plans-analytiques
  - DELETE /api/plans-analytiques
- EntryLine**
  - GET /api/facture/ecriture
- Journals**
  - POST /api/journaux
- MonthlyJournals**
  - POST /api/journaux-mois
- MoyensPayment**
  - POST /api/moyens-paiement
- parametrageGeneral**
  - POST /api/parametrage-general
- ParametrageSociete**
  - POST /api/parametrage-societe



**J'ai ensuite implémenter l'API conçue dans mon projet Ranorex DigiCompta puis j'ai rentrer ça dans ma solution Ranorex ComptaOL**

**Aujourd'hui il ne reste plus rien à faire sur l'API, elle est opérationnelle.**

**Les tests auto et la solution ont été conçus en local et mis dans le projet Azure Devops : Compta - Etats Fi pour que les collaborateurs puissent voir l'avancée de mon travail. Cela a été possible grâce à Fork. Une application que j'utilise, me permettant de pousser les changements fait en local vers un repos existant.**

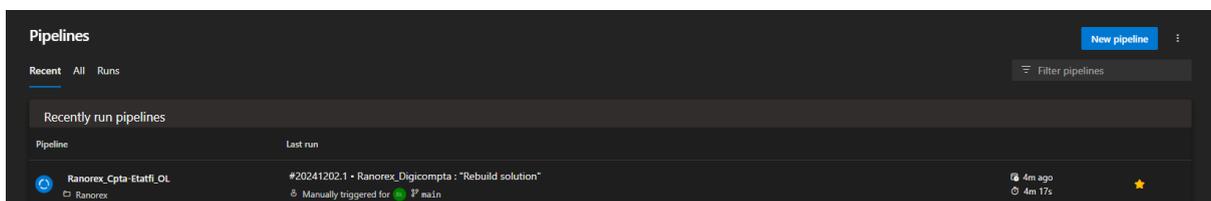


**Outil de gestion de configuration : Fork est un client git facile à utiliser pour interagir efficacement avec des dépôts Git via une interface graphique.**

**Celui permet d'effectuer les actions suivantes:**

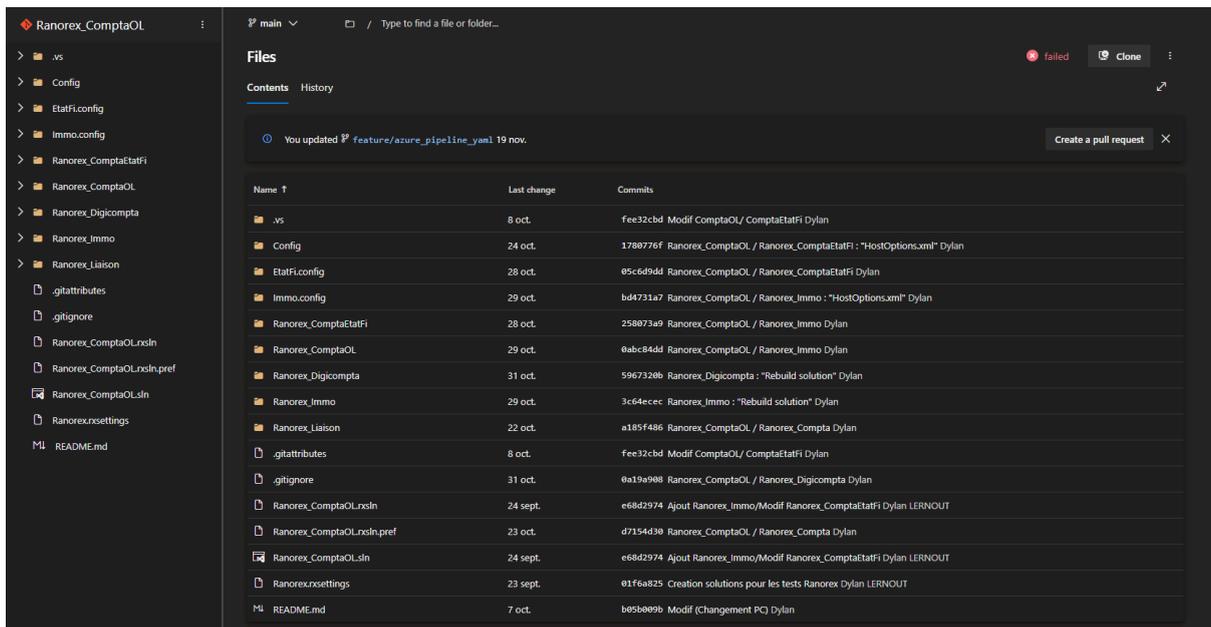
- **Cloner un dépôt**
- **Effectuer des commits**
- **Pousser des modifications**
- **Gérer des branches**

**Dans mon cas, j'ai utilisé Fork pour pousser les différentes modifications faites sur mes tests qui se trouvent sur mon pc, en local vers un dépôt distant, un repos Azure DevOps**



**J'ai eu à m'occuper de nombreux autres repos:**

- **Ranorex**
- **Ranorex\_cpta**
- **Ranorex\_Immo**
- **Ranorex\_old**
- **Ranorex\_ComptaOL**
- **Ranorex.DigiCompta**



Actuellement je suis sur une nouvelle modification du reposit Ranorex\_ComptaOL cette fois-ci menée sur le dépôt distant, sur Azure DevOps. Créer un pipeline YAML.

Une étape primordial pour se rapprocher de l'objectif qui est de rendre Ranorex ComptaOL utilisable à tous, c'est à dire

← Ranorex\_Cpta-Etatfi\_OL

Runs Branches Analytics

Description	Stages	Time
#20241202.1 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✖	53m ago 34m 23s
#20241130.1 - Ranorex_Digicompta : "Rebuild solution" Scheduled for main 5967328b	✖	1 samedi 2m 26s
#20241123.1 - Ranorex_Digicompta : "Rebuild solution" Scheduled for main 5967328b	✖	23 nov. 2m 24s
#20241118.3 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✖	18 nov. 33m 32s
#20241118.2 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✖	18 nov. 34m 8s
#20241118.1 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✖	18 nov. 51m 38s
#20241116.1 - Ranorex_Digicompta : "Rebuild solution" Scheduled for main 5967328b	✔	16 nov. 39m 15s
#20241109.1 - Ranorex_Digicompta : "Rebuild solution" Scheduled for main 5967328b	✔	9 nov. 46m 7s
#20241107.2 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✔	7 nov. 37m 57s
#20241107.1 - Ranorex_Digicompta : "Rebuild solution" Manually triggered for main 5967328b	✖	7 nov. 51m 33s
#20241105.1 - Updated run-tests.yml Manually triggered for feature/azure_pipeline_yaml 65bad3ff	✔	5 nov. 39m 7s

Qu'est-ce qu'un pipeline?

Sur Azure DevOps, une pipeline est un outil utilisé pour automatiser plusieurs tâches, grâce à l'approche CI/CD (Intégration continue/Déploiement continu) qui vise à améliorer l'efficacité du développement du logiciel ainsi que la distribution d'une nouvelle version du logiciel. Elle permet de garantir que les applications sont régulièrement testées et déployées de manière fiable et efficace.

L'intégration continue (CI) consiste à intégrer automatiquement et régulièrement les modifications de code dans un référentiel de code source partagé.

Le déploiement continu (CD) consiste à la publication automatique des mises à jour dans ces environnements.

