



Rapport de Stage de Fin d’Études

En vue de l’obtention du diplôme

**D’INGENIEUR D’ETAT**

En Génie Informatique (GI)

Thème :

Conception de Stratégies de tests etRéalisation des tests techniques et fonctionnels

Réalisé à l’entreprise :

Intelcia IT SOLUTIONS

Réalisé par :

M. OUTLIOUA Salah Eddine

Encadré par :

Mme. BAZOUN Khadija, Encadrante à l’entreprise

Pr. BOUARIFI Walid, Encadrant à l’ENSA-M

Soutenu le : 17/06/2023, devant la commission du jury :

Pr.

Pr.

Pr.

Année Universitaire 2023-2024

**DEDICACE**

*Je dédie ce modeste travail :*

##### À mes très chers parents :

*Qui ont toujours été là pour moi, et qui m’ont donné un magnifique modèle de labeur et de persévérance. J’espérons qu'ils trouveront dans ce travail toute mes reconnaissances et tout mon Amour.*

##### À mes frères et mes sœurs :

*Vous vous êtes dépensés pour moi sans compter, en reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d’atteindre cette étape de ma vie.*

##### À mes enseignants :

*Qui ont contribué à ma formation pendant cette année.*

##### À mes ami(e)s :

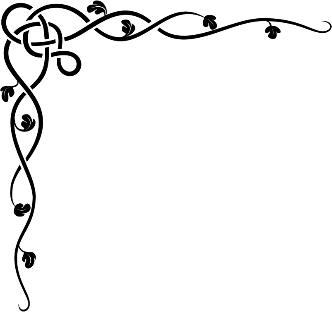
*Je vous remercie de votre patience et de votre soutien.*

##### À mes camarades :

*De l’EMSI pour les moments forts et agréables qu’on a passés ensemble.*

*À tous ceux qui sont proches de mon cœur.*

##### Merci à tous

**REMERCIEMENT**

Après avoir rendu grâce à **Dieu** le tout-puissant et le Miséricordieux, mes plus sincères remerciements vont à mes parents pour tous les sacrifices qu’ils ont consentis pour mon bien- être, pour leur aide et soutien, et leur encouragement depuis toujours dans mes études.

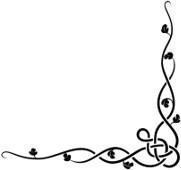
J’adresse mes remerciements les plus sincères à **Mr CHEIKH AISSAM**, domaine manager, pour son encadrement, sa disponibilité et ses conseils fructueux qu’il m’a prodiguées le long de mon projet.

Je saisis également cette occasion pour exprimer mes remerciements à toute l’équipe MERCURE pour leurs disponibilités, et leurs aides non négligeables qu’ils m’ont apportée lors de mon travail à INTELCIA.

Je suis également très reconnaissant envers **Mr OULAD HAJ THAMI RACHID** pour son suivi attentif, le temps qu'elle a consacré à mes travaux, ses précieux conseils et son rôle essentiel dans la réussite de mon encadrement.

Je tiens à remercier toute **l’équipe pédagogique** de l’**EMIS** ainsi que mes professeurs responsables de la filière IIR pour avoir assuré la partie théorique.

Enfin, mes remerciements vont à tous ceux qui ont participé de près ou de loin à la réussite de ce travail.



**RÉSUMÉ**

Ce travail s’inscrit dans le cadre de mon stage de fin d’études à Ecole Nationale des Sciences Appliquées (ENSA), en vue d’obtention du diplôme d’ingénieur d’état en génie Informatique. J’ai eu l’opportunité d’effectuer mon stage au sein de Intelcia, Intelcia groupe est un groupe marocain spécialisé dans les métiers de l’externalisation des services avec une démarche de partenaire à valeur ajoutée.

Pendant notre stage de PFE, nous avons participé au projet Mercure, une plateforme complète dédiée à moderniser les opérations de commande dans le domaine des télécommunications. L'objectif central de ce projet était de simplifier et d'optimiser les processus de commande en unifiant les canaux de vente, qu'ils soient directs ou indirects. Nous avons également eu l'opportunité de travailler sur plusieurs modules de la plateforme, en mettant l'accent sur les tests fonctionnels.

Notre responsabilité principale était d'assurer la conformité des fonctionnalités, de détecter d'éventuels dysfonctionnements et de garantir la qualité générale de la plateforme. En étroite collaboration avec l'équipe de développement, nous avons élaboré des scénarios de test, identifié et soigneusement documenté les anomalies rencontrées.

Notre contribution a également contribué à l'amélioration continue de la plateforme Mercure. Ce stage nous a permis d'acquérir une expérience précieuse dans le domaine de testing, tout en participant activement à l'évolution d'un projet majeur.

**LISTE DES ABREVIATION**

**B2B :** Business to Business

**VD :** Vente direct

**VI :** Vente indirect

**SDLC :** Software Développement Life Cycle

**TAD :** Téle travaille

**SSO :** Single sign-On

**IHM :** Interface Homme-Machin

**TDC :** Tableau de configuration

**RMCPTA :** Renouvellement mobile et changement de forfait

**CPTA :** Changement de forfait **MEP :** Mise en production **PFI :** plateforme d'intégration

**PFV :** Plateforme de Validation

**TNR :** Test Non Régressif

**TABLE DE FIGURE**

[Figure 1 : Intelcia Logo 13](#_bookmark11)

[Figure 2 : Présence à l'international 14](#_bookmark12)

[Figure 3 : Historique du groupe INTELCIA 14](#_bookmark13)

[Figure 4 : Structure de la société Intelcia 15](#_bookmark15)

[Figure 5 :Logo Intelcia It Solutions 16](#_bookmark19)

[Figure 6 : Organigramme de pôle IT Solutions 17](#_bookmark21)

[Figure 7 : Le pôle d'expertise IT d'Intelcia 18](#_bookmark23)

[Figure 8: Logo SFR Buisiness 18](#_bookmark25)

[Figure 9 : Principales références du groupe INTELCIA 19](#_bookmark28)

[Figure 10 : Méthodologie choisie (SCRUM) 24](#_bookmark41)

[Figure 11: Cycle de vie d'un sprint de la méthode SCRUM 26](#_bookmark43)

[Figure 12 : Planning des sprints 27](#_bookmark45)

[Figure 13 : Diagramme de Gantt 28](#_bookmark47)

[Figure 14 : Coût relatif de correction de bugs 31](#_bookmark53)

[Figure 15 : différence entre testeur et développeur 33](#_bookmark57)

[Figure 16 : Caricature entre le développeur et le testeur 34](#_bookmark59)

[Figure 17 : Jira Software 36](#_bookmark64)

[Figure 18 : Confluence Mercure 37](#_bookmark66)

[Figure 19 : Oracle SQL Developer Logo 37](#_bookmark69)

[Figure 20 : Postman Logo 38](#_bookmark71)

[Figure 21 : PuTTY Logo 39](#_bookmark73)

[Figure 22 : Architecture Macro du Mercure 42](#_bookmark80)

[Figure 23 : Interface Mercure 44](#_bookmark88)

[Figure 24 : Ancien IHM mobile 45](#_bookmark90)

[Figure 25 : Nouveau IHM mobile 45](#_bookmark91)

[Figure 26 : Ancien page d'ajout de ligne 46](#_bookmark93)

[Figure 27 : Nouvelle page d’ajout de ligne 46](#_bookmark94)

[Figure 28 : Ancien TDC mobile 47](#_bookmark96)

[Figure 29 : Nouveau TDC mobile 47](#_bookmark97)

Figure 30 : Tableu Plan de test 50

[Figure 31 : Diagramme BPMN des tests en PFI 52](#_bookmark103)

[Figure 32 : Diagramme BPMN des Tests en PFV 53](#_bookmark104)

[Figure 33 : Mise à jour du plan de test 54](#_bookmark106)

[Figure 34 : exemple de ticket jira 55](#_bookmark108)

[Figure 35 : Exemple d'un jira bogue 55](#_bookmark109)

[Figure 36 : exemple rapport des tests 56](#_bookmark110)

**TABLE DE MATIERE**

[DEDICACE 2](#_bookmark0)

[REMERCIEMENT 3](#_bookmark1)

[RÉSUMÉ 4](#_bookmark2)

[LISTE DES ABREVIATION 5](#_bookmark3)

[TABLE DE FIGURE 6](#_bookmark4)

[TABLE DE MATIERE 8](#_bookmark5)

[INTRODUCTION GENERALE 11](#_bookmark6)

[Chapitre 1 : Présentation de l’organisme d’accueil 13](#_bookmark7)

1. [Introduction 13](#_bookmark8)
2. [Organisme d’accueil 13](#_bookmark9)
   1. [Présentation du groupe Intelcia 13](#_bookmark10)
   2. [Structure de la société 15](#_bookmark14)
   3. [Les valeurs de la société Intelcia 15](#_bookmark16)
3. [Le pôle IT Solutions 16](#_bookmark17)
   1. [Introduction 16](#_bookmark18)
   2. [Organigramme du pole IT Solutions 17](#_bookmark20)
   3. [Domaines d’expertise 17](#_bookmark22)
   4. [Partenaire SFR Business 18](#_bookmark24)
   5. [Services de SFR Business 18](#_bookmark26)
   6. [Références du groupe Intelcia 19](#_bookmark27)
4. [Conclusion 19](#_bookmark29)

[Chapitre 2 : Contexte et Management du projet 21](#_bookmark30)

1. [Introduction 21](#_bookmark31)
2. [Présentation du projet Mercure 21](#_bookmark32)
   1. [Présentation du projet 21](#_bookmark33)
   2. [Les enjeux 21](#_bookmark34)
   3. [Les Besoins de projet 21](#_bookmark35)
   4. [Problématique 22](#_bookmark36)
   5. [Le Besoin du Client 22](#_bookmark37)
   6. [Etude de l’existant 23](#_bookmark38)
3. [Management du projet 23](#_bookmark39)
   1. [Cycle de vie de développement du projet 24](#_bookmark40)
   2. [Méthodologie du travail 25](#_bookmark42)
   3. [Planning, Sprint et Cérémonie 27](#_bookmark44)
   4. [Planning du projet 28](#_bookmark46)
4. [Conclusion 29](#_bookmark48)

[Chapitre 3 : Méthodes et Outils de Test, Historique et Evolution 30](#_bookmark49)

1. [Introduction 30](#_bookmark50)
2. [Fondamentaux des tests logiciels 30](#_bookmark51)
   1. [La nécessité des tests 30](#_bookmark52)
   2. [Les diffèrent Types des testes 31](#_bookmark54)
   3. [Différence entre le testeur et le développeur 31](#_bookmark55)
   4. [Approche mentale 32](#_bookmark56)
   5. [La psychologie des tests 33](#_bookmark58)
   6. [Améliorer la communication entre le développeur et le testeur 34](#_bookmark60)
3. [Outils et technologie utilisée 35](#_bookmark61)
   1. [Outils de gestion 35](#_bookmark62)
      1. [Jira 35](#_bookmark63)
      2. [Confluence 36](#_bookmark65)
   2. [Outils Technique 37](#_bookmark67)
      1. [Oracle SQL Developer 37](#_bookmark68)
      2. [Postman 38](#_bookmark70)
      3. [PuTTY 38](#_bookmark72)
4. [Conclusion 39](#_bookmark74)

[Chapitre 4 : Mise en œuvre du plan de test 40](#_bookmark75)

1. [Introduction 40](#_bookmark76)
2. [Le projet Mercure 40](#_bookmark77)
   1. [Principes généraux 40](#_bookmark78)
   2. [Processus métier 41](#_bookmark79)
   3. [Le périmètre fonctionnel 42](#_bookmark81)
   4. [Actes de gestion sur les offres 43](#_bookmark82)
   5. [Valeur ajouter des Tests Fonctionnels 43](#_bookmark83)
3. [La refonte et évolution de l’univers mobile 44](#_bookmark84)
   1. [La Refonte Mobile 44](#_bookmark85)
   2. [Nouvelles Pages et Interfaces 44](#_bookmark86)
      1. [Interface Mercure 44](#_bookmark87)
      2. [Interface mobile 44](#_bookmark89)
      3. [Interface d’ajout de ligne 45](#_bookmark92)
      4. [Tableau de configuration Mobile 46](#_bookmark95)
   3. [Fonctionnalités et offres ajoutées 47](#_bookmark98)
   4. [Notion des future Flipping 48](#_bookmark99)
4. [Mise en Œuvre du plan 49](#_bookmark100)
   1. [Plan de Test 49](#_bookmark101)
   2. [Exécution des Tests 51](#_bookmark102)
   3. [Évolution du Plan de Test 53](#_bookmark105)
5. [Gestion des Tests et des Tickets sur la PFV 54](#_bookmark107)
6. [Conclusion 56](#_bookmark111)

[Conclusion générale et perspectives 57](#_bookmark112)

[BIBLIOGRAPHIE 58](#_bookmark113)

**INTRODUCTION GENERALE**

De nos jours, chaque entreprise aspire à satisfaire ses clients et à améliorer ses applications afin de garantir une version finale de qualité. Dans cette perspective, SFR s'est engagé dans un projet ambitieux de refonte de Mercure, un outil de prise de commande web, multicanal, orienté haut de marché.

L'objectif de ce projet est de développer une nouvelle plateforme de commande web, multicanal et haut de gamme, spécialement conçue pour répondre aux besoins des Ingénieurs Commerciaux internes et externes. Cette initiative vise à remplacer les anciens systèmes de commande, Homere et Ecommande, qui étaient respectivement utilisés pour les canaux de vente directe et indirecte. En repensant l'outil de commande dans sa globalité, SFR cherche à améliorer l'efficacité opérationnelle, à unifier les canaux de vente et à offrir une expérience utilisateur optimale aux Ingénieurs Commerciaux.

Le projet Mercure représente une plateforme complète qui vise à moderniser les opérations de commande dans le domaine des télécommunications. Son objectif principal est de simplifier et d'optimiser les processus de commande en unifiant les canaux de vente directe et indirecte, tout en offrant des fonctionnalités avancées telles que l'éligibilité en masse, la génération de devis, la signature de contrat et l'injection de commandes dans les workflows de production.

Au cours de notre stage, nous avons eu l'opportunité de travailler sur différents modules de la plateforme Mercure, en mettant l'accent sur les tests fonctionnels. Notre mission consistait à vérifier la conformité des fonctionnalités, à détecter les éventuels dysfonctionnements et à assurer la qualité globale de la plateforme. En collaborant étroitement avec l'équipe de développement, nous avons réalisé des scénarios de test, identifié et documenté les anomalies, et contribué à l'amélioration continue de la plateforme.

Ce rapport présente une synthèse du travail réalisé tout au long de ce projet. Il est composé de quatre chapitres, qui sont les suivants :

**Chapitre 1 :** " Présentation de l’organisme d’accueil ", où nous présenterons l'entreprise d'accueil, le sujet du stage, la problématique abordée et les objectifs visés. Nous détaillerons également l'approche adoptée pour assurer le bon déroulement du projet.

**Chapitre 2 :** " Contexte et Management du projet ", où nous entamerons une exploration approfondie du contexte dans lequel le projet prend place. Cela englobera une introduction détaillée au projet lui-même, en mettant en lumière ses tenants et aboutissants. Nous nous pencherons également sur la stratégie globale qui guide la gestion du projet, en examinant comment les éléments clés sont orchestrés pour aboutir au succès.

**Chapitre 3 :** "Méthodes et Outils de Test, Historique et Evolution", Exploration des Méthodes et Outils de Test, Trajectoire Historique et Évolution" où nous entamerons une analyse approfondie des différentes méthodes et outils de test employés. En parallèle, nous retracerons l'évolution historique de ces approches, mettant en lumière les changements marquants au fil du temps.

**Chapitre 4 :** "Mise en œuvre du plan de test", Dans ce chapitre, nous explorons la mise en œuvre concrète du plan de test élaboré dans le cadre du projet Mercure. Après avoir établi la méthodologie de test fonctionnel dans le chapitre précédent, notre attention se tourne désormais vers la manière dont cette méthodologie est appliquée sur le terrain.

**Chapitre 1 : Présentation de l’organisme d’accueil**

1. **Introduction :**

Ce chapitre vise à fournir un aperçu du contexte global de mon projet de fin d'études. Dans un premier temps, je présenterai la société « Intelcia IT Solutions », qui est l'organisme d'accueil. Ensuite, je décrirai le cadre général et les circonstances entourant le développement du projet Mercure.

Je détaillerai également les objectifs que nous devons atteindre, ainsi que l'approche méthodologique que nous avons adoptée pour assurer une gestion efficace de notre projet.

1. **Organisme d’accueil**
   1. **Présentation du groupe Intelcia**



Figure 1 : Intelcia Logo

INTELCIA Group est le premier groupe marocain spécialisé dans les métiers de l’externalisation des services avec une démarche de partenaire à valeur ajoutée. Il intervient dans les secteurs télécoms, médias, assurances, e-commerce et d’autres secteurs.

Créé il y a 20 ans au Maroc et porté par une ambition forte, le Groupe a rapidement étendu sa présence géographique et ses activités pour accompagner ses clients sur l'ensemble de leurs opérations d'externalisation et leur permettre de mieux se concentrer sur leurs enjeux et cœur de métier.

En 2020, INTELCIA poursuit son expansion internationale avec une implantation en Amérique dans les caraïbes et de nouvelles destinations en Europe. Avec plus de 26 580 collaborateurs, le groupe est présent aujourd’hui sur 56 sites à travers 17 pays dont 1 site client en France et 11

sites clients au Portugal et ambitionne d’atteindre 30 000 collaborateurs vers la fin de 2021 avec une présence plus importante en Europe et de nouvelles implantations aux Etats-Unis.



Figure 2 : Présence à l'international

Le groupe INTELCIA s’est développé à partir de fusions et acquisitions majeures réalisés depuis 2000 qui peuvent être représentées par la figure ci-dessous :



Figure 3 : Historique du groupe INTELCIA

* 1. **Structure de la société**

A travers ses services, Intelcia permet à ses clients de se concentrer sur leurs enjeux et leur cœur de métier. Elle propose ainsi une offre globale et multisectorielle autour de 4 domaines de solutions :



Figure 4 : Structure de la société Intelcia

* 1. **Les valeurs de la société Intelcia**

Ce qui rend Intelcia unique, ce qui les unit, ce sont les valeurs auxquelles ils croient. Ces principes fondateurs sur lesquels ils basent leurs décisions, leurs actions et leur comportement reflètent leurs croyances et leurs convictions et façonnent la culture et l'identité d'Intelcia en tant qu'entreprise. Les valeurs d'Intelcia peuvent inclure des éléments tels que l'innovation, l'excellence, la responsabilité sociale, l'intégrité et le respect de l'individu, et elles guident les choix et les comportements des employés dans leur travail quotidien.

 **Optimisme. Ambition. Audace:** Nos rêves sont notre

principal moteur. Notre vision optimiste est résolument tournée vers l’avenir. Elle nous aide à penser puis à façonner un lendemain meilleur. Nous croyons en nos capacités. Voilà pourquoi nous n’hésitons psychologiques pour réaliser nos ambitions.

**Transparence. Respect. Bienveillance:** Notre culture

d’entreprise est basée sur le partage. Générosité et cohésion sont nos leitmotivs. A INTELCIA, nous respectons la différence. C’est ainsi que nous créons des relations riches et durables qui mènent inévitablement vers un succès collectif. Nous veillons également au bien-être de nos collaborateurs de nos clients et de nos partenaires.

 **Excellence. Agilité. Inventivité:** Nous avons une

responsabilité de résultat. Nous prônons donc la prise d’initiative et la créativité dans l’exécution de nos missions. Nous sommes des bâtisseurs. Nous nous surpassons au quotidien pour trouver des solutions à tous les obstacles. C’est ce qui nous permet de répondre avec agilité aux exigences et aux besoins de nos clients

1. **Le pôle IT Solutions**
   1. **Introduction**



Figure 5 :Logo Intelcia It Solutions

Le pôle IT Solutions, créé en 2019, est le fruit de regroupement des deux structures IT : ATEXO Maroc et SBS (SFR Business Solutions) Morocco, filiales du groupe Intelcia. Le pôle accumule une expérience de 15 ans comme acteur de référence dans l’infogérance et la digitalisation.

Le pôle INTELCIA IT Solutions a pour vocation de renforcer le portefeuille d’expertise du groupe, en ligne avec sa stratégie de diversification entamée il y a quelques années avec le lancement des activités dans le Digital Advertising et le BPO (Business Process outsourcing).

Le pôle INTELCIA IT Solutions s’appuiera pour son lancement sur les expertises des entreprises acquises et regroupera différents services d’externalisation de solutions SI comme le développement et la maintenance d’applications, les services d’infrastructure, la sécurité, le testing…

* 1. **Organigramme du pole IT Solutions**

Le pôle IT Solutions est composé de plusieurs direction représentant chacune des Business Unit. La figure ci-dessous présente la structure organisationnelle du pôle :

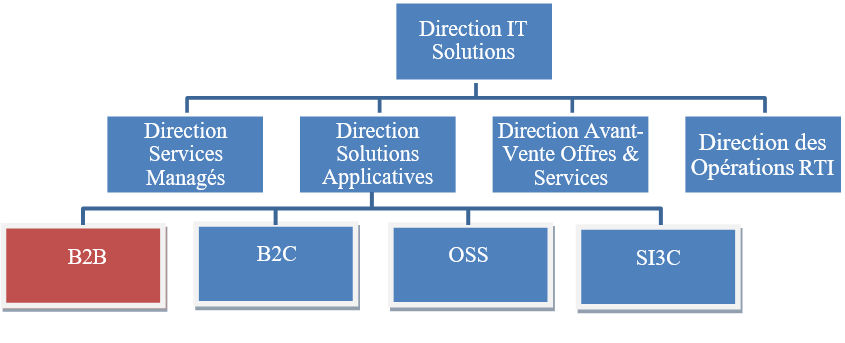


Figure 6 : Organigramme de pôle IT Solutions

Mon stage PFE a été effectué au sein de la direction des solutions applicatives, particulièrement dans l’unité opérationnelle de B2B, sous l’encadrement du Manager B2B Mme BAZOUN Khadija.

* 1. **Domaines d’expertise**

Intelcia IT Solutions est spécialisée principalement dans quatre pole d’expertise Services Managés et les solutions applicatives, Business Intelligence et Consulting.

La figure ci-après regroupe l’ensemble des domaines d’expertises d’Intelcia IT Solutions

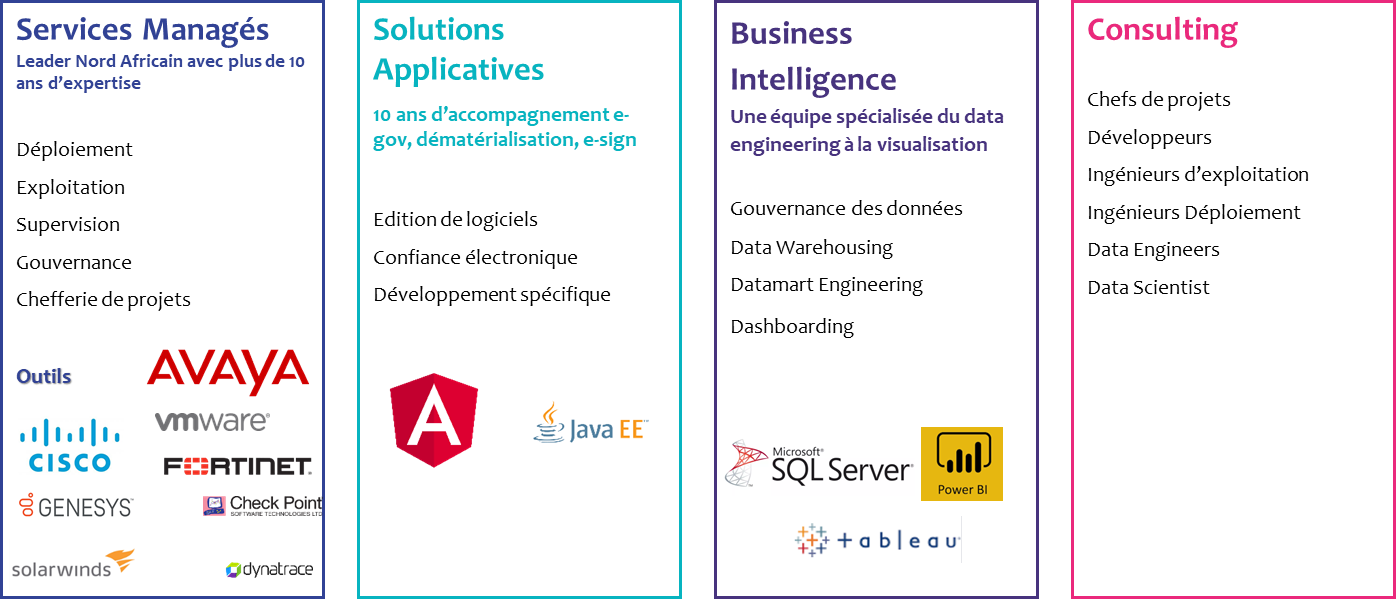


Figure 7 : Le pôle d'expertise IT d'Intelcia

* 1. **Partenaire SFR Business**



Figure 8: Logo SFR Buisiness

La Société Française du Radiotéléphone, SFR, est un opérateur français de télécommunications créé en 1987 par la Compagnie générale des eaux. SFR propose un réseau très haut débit, fixe et mobile, sur le territoire français à destination des particuliers et des entreprises. L’entreprise réalise également des investissements importants pour le développement de la 5G.

* 1. **Services de SFR Business**

Les services de SFR Business sont basés principalement sur des offres et solutions IT et infrastructures réseaux, sont présentées ci-contre :

* Les solutions de sécurité Internet
* Les solution d’Internet des Objets
* Les offres de Relation client
* Les solutions WAN et LAN

Les offres et abonnements de téléphonie fixe et mobile (4G /5G).

* 1. **Références du groupe Intelcia**

L’activité de Groupe INTELCIA ne se restreint pas au niveau national. En effet, elle est appelée à devenir l’un des bureaux de référence, notamment en Afrique et en Europe, et dispose notamment d’un grand nombre de clients.

Elle compte plus de 100 clients, comme illustré dans la figure ci-après.



1. **Conclusion**

Figure 9 : Principales références du groupe INTELCIA

En conclusion, ce chapitre a joué un rôle essentiel en nous fournissant une présentation détaillée de l’organisme d’accueil et de ses activités. Grâce à cette exploration approfondie, nous avons pu acquérir une compréhension claire de son fonctionnement, de sa mission et de sa place dans

son domaine d’activité. Cette connaissance préliminaire nous permettra d’aborder le chapitre suivant avec une base solide.

**Chapitre 2 : Contexte et Management du projet**

1. **Introduction**

Dans ce chapitre, nous plongerons au cœur même du projet. Nous examinerons en détail sa nature, sa portée et les principes de gestion qui ont été appliqués pour le mener à bien. Cette exploration approfondie nous permettra de saisir pleinement les tenants et aboutissants du projet, tout en mettant en lumière les éléments clés de sa gestion.

1. **Présentation du projet Mercure**

## Présentation du projet

Mercure a été conçu pour remplacer les anciens systèmes de prise de commande, à savoir SSA Homere et Ecommande, qui étaient respectivement dédiés aux canaux de vente directe (VD) et indirecte (VI). Le principal objectif de Mercure est d'unifier ces outils en offrant des fonctionnalités communes telles que l'éligibilité en masse, la génération de devis, la signature de contrat et l'injection de commandes dans les workflows de production.

## Les enjeux

La mise en place de Mercure a été motivée par plusieurs enjeux essentiels. Tout d'abord, il était nécessaire de disposer d'un outil capable de traiter les commandes liées au parc, sachant que 70% des ventes se réalisent sur ce dernier.

* Un outil qui permet de travailler sur le parc.
* Un outil agnostique des canaux de vente et des organisations.
* Un outil fluide et simple à utiliser.

## Les Besoins de projet

Les besoins identifiés dans le cadre de ce projet sont les suivants :

* Un outil de prise de commande web, multicanal et haut de marché.
* Une gestion efficace des commandes internes et externes.
* La possibilité de travailler sur le parc existant pour faciliter les ventes.
* Une consultation aisée du parc dès le début du processus de vente.
* La réalisation des actions liées au parc au sein de l'outil.
* L'unification des canaux de vente et des organisations.
* Une fluidité accrue pour éviter les ruptures et les ressaisies.
* La possibilité de mettre en place l'outil de manière progressive et itérative.
* Une interface fluide, simple et ergonomique pour les utilisateurs.
* Des traitements longs effectués en arrière-plan pour optimiser les performances.

## Problématique

La problématique centrale de mon stage dans le cadre du projet Mercure se pose quant à l'efficacité, la fiabilité et l'ergonomie de la plateforme. Comment garantir que l'outil de prise de commande web répond aux attentes des Ingénieurs Commerciaux, tant internes qu'externes, en offrant une expérience utilisateur fluide et simplifiée ? Comment assurer la qualité des fonctionnalités telles que l'éligibilité en masse, la génération de devis, la signature de contrat et l'injection de commandes dans les workflows de production ? La problématique essentielle est donc de tester et de valider chaque aspect de la plateforme Mercure afin d'assurer sa conformité, son interopérabilité et sa performance, tout en garantissant une expérience utilisateur optimale pour les utilisateurs finaux.

Afin d'explorer en profondeur la problématique, nous aborderons une série de questions clés liées à l'efficacité, la fiabilité, l'ergonomie de Mercure et la qualité de ses fonctionnalités essentielles :

* Quelles sont les attentes spécifiques des Ingénieurs Commerciaux internes et externes en termes d'expérience utilisateur sur la plateforme Mercure ?
* Quels critères peuvent être utilisés pour évaluer l'efficacité, la fiabilité et l'ergonomie de Mercure, en particulier pour les fonctionnalités ?
* Quels tests fonctionnels peuvent être réalisés pour valider la conformité et la performance de Mercure, ainsi que pour détecter d'éventuels dysfonctionnements ?

## Le Besoin du Client

Le besoin du client dans le cadre de cette problématique est de disposer d'un ingénieur chargé d'effectuer les tests techniques et fonctionnels sur la plateforme Mercure. En tant que client, il est crucial pour eux de s'assurer que l'outil répond aux attentes des Ingénieurs Commerciaux. Afin d'atteindre ces objectifs, le client reconnaît l'importance d'avoir un ingénieur dédié aux tests fonctionnels qui sera en mesure de mettre en œuvre une méthodologie rigoureuse pour évaluer chaque aspect de la plateforme Mercure. Cet ingénieur sera responsable de la

planification, de l'exécution et du suivi des tests fonctionnels, ainsi que de la documentation des résultats et des éventuelles anomalies détectées. Son rôle est essentiel pour garantir la conformité, l'interopérabilité et la performance de Mercure, tout en assurant une expérience utilisateur optimale pour les utilisateurs finaux.

## Etude de l’existant

L'étude d'existant constitue une étape cruciale dans tout processus d'évolution et de refonte d'une application. Dans cette partie, nous nous pencherons sur l'état actuel de l'application, en mettant en lumière les éléments qui nécessitent une transformation et une amélioration pour aboutir à Mercure. Cette transition implique un passage de l'ancien système "Ecommende" vers le nouveau modèle "Mercure".

L'objectif principal de cette étude est d'identifier les domaines spécifiques de l'application qui demandent une attention particulière en termes d'évolution et de refonte. Pour ce faire, nous élaborerons des plans de tests détaillés visant à évaluer la robustesse, la convivialité et la performance de chaque composant de l'application. Cette approche méthodique nous permettra de garantir que chaque partie de l'application Mercure est soigneusement testée et optimisée, en vue d'offrir une expérience utilisateur améliorée et des fonctionnalités plus avancées.

# Management du projet

La planification est une étape essentielle d'un projet, qui représente le premier défi pour le gestionnaire de projet ou le maître d'œuvre. L'objectif est de garantir que les résultats souhaités sont atteints et que les attentes des différents groupes et individus impliqués dans le projet sont satisfaites. Pour cela, il est nécessaire d'estimer la progression du projet à travers les différentes phases du cycle de développement.

## cycleCycle de vie de développement du projet

Figure 10 : Méthodologie choisie (SCRUM)

Cycle de vie du développement logicielle également connu sous le nom SDLC (Software Development Life Cycle), est un processus systématique utilisé pour gérer et contrôler le développement des logiciels. Il englobe l’ensemble des étapes, activités et phases nécessaires pour concevoir, développer, tester, déployer et maintenir un logiciel de manière efficace et de haute qualité.

Le SDLC se compose généralement des phases suivantes :

**Analyse des besoins :** Cette phase initiale consiste à comprendre les exigences et les objectifs du logiciel à développer. Elle implique des interactions avec les parties prenantes, tels que les clients, les utilisateurs finaux et les experts métier, afin de recueillir et d’analyser les besoins fonctionnels et non fonctionnels.

**Conception :** Une fois les besoins identifiés, la phase de conception se concentre sur la création d’une architecture logicielle globale. Cela inclut la définition de l’architecture du système, la conception détaillée des composants, des modules et des interfaces, ainsi que la planification des tests à venir.

**Développement :** Dans cette phase, les développeurs transforment la conception en code source en utilisant des langages de programmation et des frameworks appropriés. Les meilleures pratiques de codage, les normes de programmation et les conventions sont appliquées pour assurer la qualité du code.

**Tests :** Les tests constituent une phase cruciale du SDLC. Différentes techniques de test, telles que les tests unitaires, les tests d’intégration, les tests de système et les tests de régression, sont utilisées pour vérifier que le logiciel répond aux spécifications et fonctionne correctement. Les anomalies et les erreurs sont identifiées, signalées et corrigées lors de cette étape.

**Déploiement :** Une fois les tests réussis, le logiciel est prêt à être déployé. Il est installé sur les serveurs ou les environnements de production et est rendu accessible aux utilisateurs finaux.

**Maintenance :** Après le déploiement, le logiciel est soumis à une phase de maintenance continue. Cela comprend la correction des bogues, les mises à jour, les améliorations de performance, la prise en charge des utilisateurs et la gestion des demandes de changement.

Il convient de noter que le SDLC peut varier en fonction de la méthodologie de développement utilisée, telle que le modèle en cascade (waterfall), le modèle en V, le développement agile ou le développement itératif. Chaque approche a ses propres avantages et inconvénients, et peut être adaptée en fonction des besoins et des contraintes spécifiques du projet.

En suivant le SDLC de manière rigoureuse, les organisations peuvent s’assurer que le développement logiciel est géré de manière contrôlée, prévisible et efficace, en garantissant la qualité du logiciel livré tout au long du processus.

## Méthodologie du travail

Dans le cadre du projet, adopter une approche agile signifie être capable de s'adapter aux changements, qu'ils soient liés aux besoins, aux priorités, aux technologies ou à l'équipe de projet.

Les méthodes agiles sont des approches de gestion de projet largement utilisées dans le domaine informatique, elles se basent sur des cycles de développement itératifs et adaptatifs, en réponse aux besoins évolutifs du client, elles impliquent tous les collaborateurs et le client dans le

processus de développement du projet, ce qui permet de mieux répondre aux attentes du client dans des délais serrés tout en augmentant les compétences des collaborateurs, cela offre un avantage concurrentiel aux deux parties.

SCRUM est actuellement la méthode agile la plus populaire et a été choisie pour mener à bien le présent projet. Le choix de cette méthode est justifié par le besoin d'adaptabilité aux changements, de satisfaction du client et de livraison mensuelle de versions stables et fonctionnelles d'un sous-ensemble de fonctionnalités du système.

La méthode SCRUM repose sur la division du projet en itérations appelées « sprints ». Chaque sprint dure généralement entre deux semaines et un mois. Avant chaque sprint, les tâches sont estimées en termes de temps et de complexité en utilisant la méthode PERT à 3 Points ces estimations permettent de planifier les livraisons et d'estimer le coût des tâches pour le client, les user stories qui font l'objet d'un sprint constituent ce que l'on appelle un sprint backlog et représentent le produit potentiellement livrable à la fin du sprint.

Le cycle de vie d'un sprint Scrum est illustré dans la figure ci-dessous :



Figure 11: Cycle de vie d'un sprint de la méthode SCRUM

## Planning, Sprint et Cérémonie

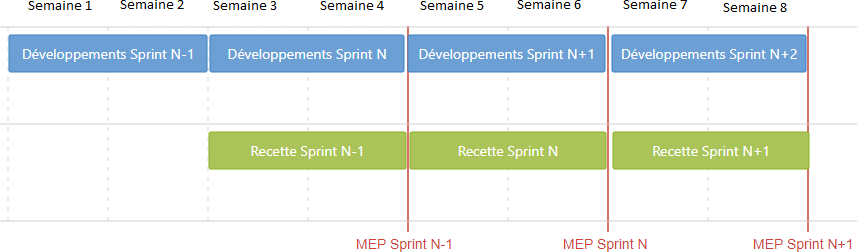


Figure 12 : Planning des sprints

L'application de la méthode Scrum dans notre projet MERCURE s'est révélée plus efficace que les méthodes traditionnelles où le développeur répond aux besoins exprimés par le chef de projet qui détient le savoir et la prise de décision, ce qui n'est pas le cas dans notre projet. En utilisant Scrum, nous avons pu répondre à la majorité de nos critères de choix, tels que la livraison dans les délais, la réactivité et la flexibilité face aux changements des exigences. Le projet MERCURE est de nature évolutive, car les besoins n'ont pas été entièrement identifiés dès le début, ce qui nous a naturellement orientés vers cette méthode agile.

Dans notre approche Scrum, les développeurs interviennent sur des sprints de deux semaines, comprenant à la fois le développement et les tests unitaires. Les testeurs prennent ensuite en charge ces fonctionnalités lors du sprint suivant. Le Time To Market sur le projet MERCURE est de deux semaines, ce qui signifie que nous déployons une nouvelle version de l'application deux fois par mois, une en début et une autre au milieu du mois.

Nous suivons également les pratiques clés de Scrum pour assurer la transparence et la coordination au sein de l'équipe. Chaque jour, nous organisons une réunion appelée "Daily Scrum" où chaque développeur informe de ce qu'il a accompli dans la journée et ce qu'il prévoit de faire le lendemain. Cette réunion dure un quart d'heure et se déroule tous les après-midis à 16h30.

Une autre pratique essentielle de Scrum que nous utilisons est le "Poker Planning". Une fois par semaine, au minimum, nous présentons les User Stories aux développeurs afin d'estimer leur complexité. Cette séance de "Poker Planning" dure environ une heure et permet d'obtenir une estimation collective des efforts requis pour chaque tâche.

À la fin de chaque sprint, nous organisons une "Sprint Review" pour faire le bilan du sprint écoulé et réaliser une démonstration des sujets majeurs développés. Cela nous permet d'évaluer les progrès réalisés et de recueillir les commentaires des parties prenantes.

Pour la gestion de projet, nous utilisons JIRA comme logiciel de suivi des demandes. C'est là que nous gérons toutes les phases du projet, du cadrage à la conception, en passant par le développement, la recette et le déploiement dans l'environnement final.

En utilisant la méthode Scrum et en nous appuyant sur ces pratiques et outils, nous sommes en mesure de gérer efficacement notre projet MERCURE, en livrant régulièrement des versions améliorées de l'application tout en restant adaptatifs aux besoins changeants

## Planning du projet

Dans le cadre de la conduite du projet, la réalisation d’un planning à suivre tout au long du stage de fin d’études s’impose. Ainsi, après avoir déterminé le processus de développement adopté, l’étape suivante est de planifier les tâches selon ce processus et selon les contraintes du projet. Afin d’avoir une efficacité maximale et de pouvoir gérer au mieux le temps qui nous a été imparti pour notre stage, nous avons élaboré un planning prévisionnel afin de prévoir des dates pour chacune des tâches et ainsi pouvoir gérer au mieux notre stage.

La figure de la page suivante est une modélisation des différentes tâches exposées ci-dessus à travers un diagramme de Gantt.

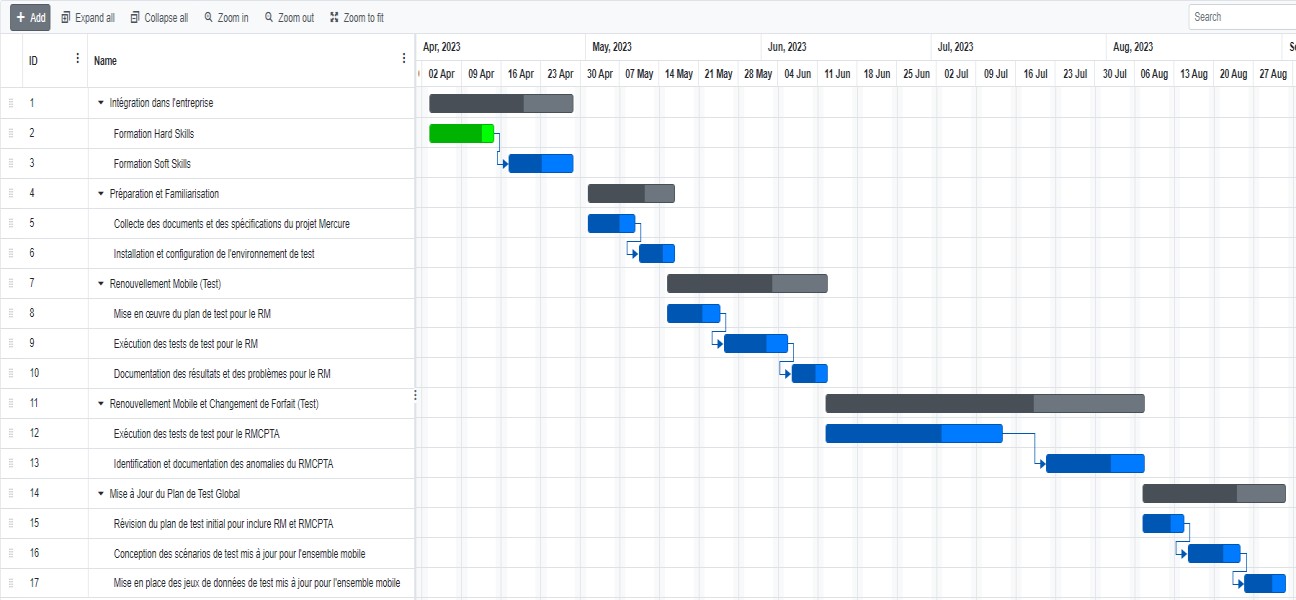


Figure 13 : Diagramme de Gantt

# Conclusion

Ce chapitre fait office de point de départ de présentation du projet. Nous avons expliqué également la problématique, la solution mise en place et finalement la méthode de travail Scrum qui garantit un cadre agile pour traiter les besoins métiers d’une façon souple et immédiate. Et pour finir la planification du projet avec le diagramme de Gantt.

On va réserver le chapitre suivant à la présentation des spécifications fonctionnelles.

**Chapitre 3 : Méthodes et Outils de Test, Historique et Evolution**

# Introduction

Le troisième chapitre de ce rapport plonge au cœur de l’étude approfondie dans le cadre de ce projet. Cette partie essentielle met en lumière la nécessité de mener des tests rigoureux pour évaluer son fonctionnement et son efficacité. Ainsi, nous aborderons le principe fondamental des tests et explorerons le processus détaillé qui les accompagne. Enfin, nous examinerons les technologies spécifiques utilisées pour soutenir cette démarche. À travers cette exploration, nous découvrirons l’impact significatif de ces éléments sur mon projet, offrant ainsi une compréhension approfondie des fondements de sa conception et de son amélioration continue.

# Fondamentaux des tests logiciels

## La nécessité des tests

Les tests garantissent que ce que vous obtenez en fin de compte est ce que vous vouliez construire. Nous sommes des êtres humains et les êtres humains font des erreurs pendant le développement Certaines erreurs n’ont pas beaucoup d’impact sur notre vie quotidienne et peuvent être ignorées, mais certaines erreurs sont si graves qu’elles peuvent briser tout le système ou le logiciel. Dans ce genre de situations, vous devez veiller à ce que ces erreurs soient détectées bien à l’avance et avant de déployer le système/logiciel dans l’environnement de production. Exemples : Les sites Web bancaires ou les distributeurs automatiques de billets devraient être soigneusement testés afin de maintenir la crédibilité de la banque. Le système de contrôle du trafic aérien est un système très critique qui doit être testé avant d’être déployé en direct, car les vies humaines en dépendent.

Comme vous pouvez le voir ci-dessous, la figure représente le coût relatif de correction de bugs. Elle offre une visualisation des coûts associés à la correction de bugs, en fonction de différents paramètres ou facteurs.

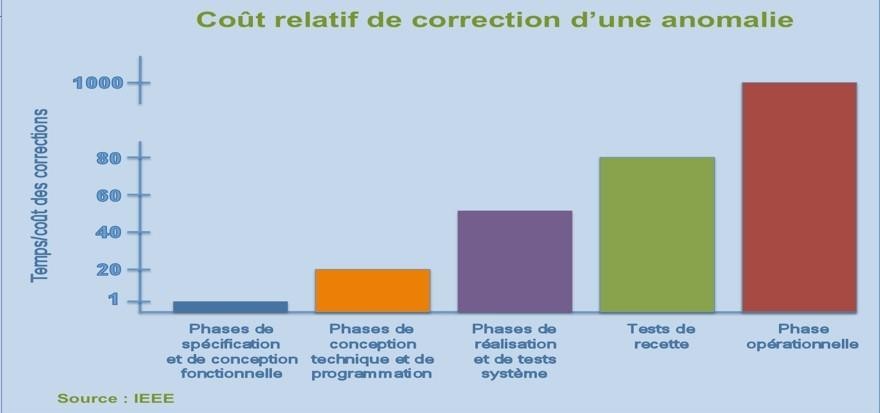


Figure 14 : Coût relatif de correction de bugs

## Les diffèrent Types des testes

Plusieurs techniques de test sont utilisées pour garantir une couverture exhaustive des fonctionnalités du système. Parmi ces techniques, on retrouve :

**-Les tests unitaires :** Ces tests visent à vérifier chaque composant individuellement pour s'assurer de son bon fonctionnement.

**-Les tests d'intégration :** Ils consistent à vérifier la communication et l'interaction entre les différents composants du système.

**-Les tests de validation :** Ces tests sont effectués pour valider les fonctionnalités du système dans son ensemble, en s'assurant qu'il répond bien aux besoins et aux attentes des utilisateurs.

**-Les tests de régression :** Ils sont réalisés pour s'assurer que les modifications ou les ajouts de fonctionnalités n'ont pas introduit de régressions dans le système, c'est-à-dire des dysfonctionnements ou des effets indésirables sur les fonctionnalités existantes.

## Différence entre le testeur et le développeur

Le testeur et le développeur sont deux rôles distincts dans le processus de développement logiciel, chacun ayant des responsabilités et des tâches spécifiques. Voici les principales différences entre ces deux rôles :

#### Responsabilités principales :

**-Testeur :** Le testeur est responsable de la validation et de la vérification du logiciel. Il conçoit, exécute et évalue les tests pour s’assurer que le logiciel répond aux exigences spécifiées. Le testeur identifie les problèmes, les bogues et les défauts du logiciel et les communique à l’équipe de développement pour correction.

**-Développeur :** Le développeur est responsable de la conception, de la création et de la maintenance du code source du logiciel. Il traduit les exigences fonctionnelles en code et développe les fonctionnalités du logiciel en utilisant différents langages de programmation. Le développeur est responsable de la qualité, de la performance et de la fiabilité du code qu’il produit.

Compétences techniques :

**-Testeur :** Les compétences techniques requises pour un testeur comprennent une compréhension approfondie des techniques de test, des méthodologies de test et des outils de test automatisés. Le testeur doit également avoir une bonne connaissance des différents types de tests et des principes de qualité logicielle.

## Approche mentale

**-Testeur :** Les testeurs adoptent une approche critique et exploratoire pour évaluer le logiciel. Ils cherchent activement des erreurs, des bogues et des problèmes potentiels. Les testeurs sont souvent axés sur la détection des défauts et l’amélioration de la qualité du logiciel.

**-Développeur :** Les développeurs adoptent une approche créative et orientée vers la solution pour développer des fonctionnalités logicielles. Ils se concentrent sur la conception efficace et l’implémentation du code en fonction des exigences spécifiées. Les développeurs sont souvent axés sur la création et la réalisation de solutions techniques.

Il est important de noter que ces deux rôles sont complémentaires et travaillent en étroite collaboration pour produire un logiciel de haute qualité. La communication et la collabo- ration entre les testeurs et les développeurs sont essentielles pour assurer un processus de développement efficace et la livraison d’un logiciel fiable.



Figure 15 : différence entre testeur et développeur

## La psychologie des tests

La psychologie des tests en informatique est un domaine d’étude qui se concentre sur les aspects psychologiques impliqués dans la conception, l’exécution et l’évaluation des tests logiciels. Les tests informatiques sont essentiels pour garantir la qualité des logiciels et minimiser les erreurs, mais ils impliquent également des interactions entre les êtres humains et les systèmes informatiques, ce qui soulève des questions psychologiques importantes.

Voici quelques éléments clés de la psychologie des tests :

1. **Perception des erreurs :** Les psychologues se sont intéressés à la façon dont les êtres humains perçoivent et réagissent aux erreurs dans les logiciels. La compréhension de la manière dont les utilisateurs interprètent et réagissent aux erreurs peut aider les testeurs à concevoir des tests plus efficaces et à identifier les types d’erreurs les plus préjudiciables pour les utilisateurs.
2. **Biais cognitifs :** Les biais cognitifs sont des erreurs systématiques de raisonnement qui peuvent influencer les décisions et les jugements des testeurs. Par exemple, l’effet de confirmation peut amener les testeurs à rechercher des preuves confirmant leurs hypothèses plutôt que de considérer objectivement toutes les possibilités. La connaissance de ces biais peut aider les testeurs à les atténuer et à améliorer la qualité des tests.
3. **Charge cognitive :** Les tests informatiques peuvent être cognitivement exigeants, ce qui signifie qu’ils peuvent surcharger les capacités mentales des testeurs. La charge cognitive peut affecter la performance des testeurs et entraîner des erreurs. Les psychologues étudient les mécanismes de la

charge cognitive et proposent des stratégies pour la réduire, comme la simplification des tâches de test ou la fourniture d’outils d’assistance.

1. **Motivation et engagement :** Les testeurs peuvent être confrontés à des tâches répétitives et monotones, ce qui peut affecter leur motivation et leur engagement. La psychologie des tests en informatique examine comment maintenir la motivation des testeurs, par exemple en introduisant des éléments de gratification ou en offrant des récompenses tangibles pour les performances de test.
2. **Rétroaction et communication :** La rétroaction est un élément essentiel des tests informatiques. Les testeurs doivent être en mesure de communiquer efficacement les résultats de leurs tests aux développeurs et aux autres parties prenantes. La psychologie des tests en informatique étudie comment fournir une rétroaction constructive et faciliter une communication claire entre les testeurs et les autres membres de l’équipe de développement.

En intégrant des connaissances issues de la psychologie dans les pratiques de test informatique, il est possible d’améliorer la qualité des tests, d’optimiser l’efficacité des testeurs et de réduire les erreurs coûteuses. Cela contribue à la création de logiciels plus fiables et conviviaux pour les utilisateurs.

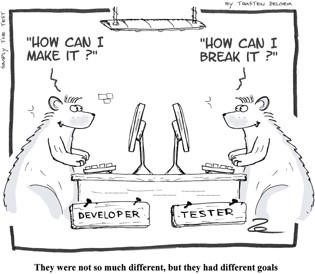


Figure 16 : Caricature entre le développeur et le testeur

## Améliorer la communication entre le développeur et le testeur

La communication entre le développeur et le testeur est d’une importance capitale dans le processus de développement logiciel. Voici quelques raisons qui mettent en évidence son importance :

* + Commencer par une collaboration plutôt que par des conflits.
  + Communiquer les découvertes sur le produit de façon neutre sans critiquer la personne.
  + Essayer de comprendre ce que ressent la personne et se mettre à sa place.
  + Confirmer que l’autre personne a bien compris ce que l’on a dit et vice versa.

En résumé, la communication entre le développeur et le testeur est essentielle pour garantir la qualité du logiciel. Une collaboration étroite, un partage de connaissances, une détection précoce des problèmes et une amélioration continue sont autant de bénéfices découlant d’une communication efficace. Il est donc crucial de favoriser une communication ouverte et régulière entre ces deux parties tout au long du processus de développement logiciel.

# Outils et technologie utilisée

## Outils de gestion

### Jira

Jira est un logiciel de gestion de projet et de suivi des problèmes développé par Atlassian. Il est utilisé par de nombreuses équipes et organisations pour planifier, suivre et gérer les tâches, les projets, les bogues et les demandes de fonctionnalités.

Voici quelques points clés à connaître sur Jira :

1. **Gestion des projets :** Jira permet de créer et d’organiser des projets, qu’il s’agisse de projets logiciels, de projets de développement, de projets de service, ou d’autres types de projets. Chaque projet peut être configuré avec des paramètres spécifiques, des flux de travail personnalisés et des permissions d’accès appropriées.
2. **Création et suivi des tickets :** Jira utilise des tickets pour représenter les tâches, les bogues, les demandes de fonctionnalités et d’autres types de travaux à réaliser. Les utilisateurs peuvent créer des tickets, leur attribuer des priorités, les affecter à des membres de l’équipe, suivre leur progression et les mettre à jour au fur et à mesure de leur avancement.
3. **Flux de travail personnalisables :** Jira offre une flexibilité dans la définition des flux de travail pour chaque projet. Les flux de travail permettent de définir les différentes étapes que les tickets peuvent traverser, de la création à la résolution. Les transitions entre les étapes peuvent être personnalisées en fonction des besoins de l’équipe et du projet.
4. **Suivi des problèmes et des bogues :** Jira facilite le suivi des problèmes et des bogues. Les équipes peuvent créer des tickets pour signaler les problèmes, les classer par gravité, leur attribuer des responsables, les hiérarchiser et les résoudre en suivant un processus structuré.
5. **Tableaux Kanban et scrum boards :** Jira propose des tableaux Kanban et des tableaux Scrum pour visualiser les tâches et les tickets en cours. Les tableaux Kanban per- mettent de suivre le flux de travail des tickets, de les déplacer entre différentes colonnes représentant les étapes du processus. Les tableaux Scrum sont utilisés pour gérer les sprints et suivre l’avancement des tickets dans un contexte de méthodologie agile.
6. **Collaboration et communication :** Jira facilite la collaboration entre les membres de l’équipe. Les utilisateurs peuvent commenter les tickets, ajouter des pièces jointes, mentionner d’autres membres de l’équipe et discuter des problèmes directement dans Jira. Cela favorise la communication transparente et la résolution efficace des problèmes.
7. **Intégrations et écosystème d’extensions :** Jira peut être intégré à d’autres outils et services utilisés par les équipes de développement. Il dispose d’un écosystème d’extensions et d’intégrations, telles que des connecteurs avec des outils de développement, des outils de test, des outils de gestion de versions, des outils de collaboration, etc. Ces intégrations permettent de synchroniser les informations, d’automatiser les tâches et de connecter les flux de travail entre différents outils.

Jira est utilisé dans une variété d’industries et de secteurs, allant du développement logiciel à la gestion de projets, en passant par les services informatiques et bien d’autres. Il offre une suite complète de fonctionnalités pour la gestion de projets, la gestion des tâches et le suivi des problèmes, permettant aux équipes de travailler de manière organisée et efficace.



### Confluence

Figure 17 : Jira Software

Confluence est une plateforme de collaboration et de gestion des connaissances développée par Atlassian. Elle permet aux équipes de travailler ensemble, de créer, d’organiser et de partager des informations au sein d’une organisation. Confluence est conçu pour faciliter la documentation, la communication et la collaboration entre les membres d’une équipe ou d’une organisation.

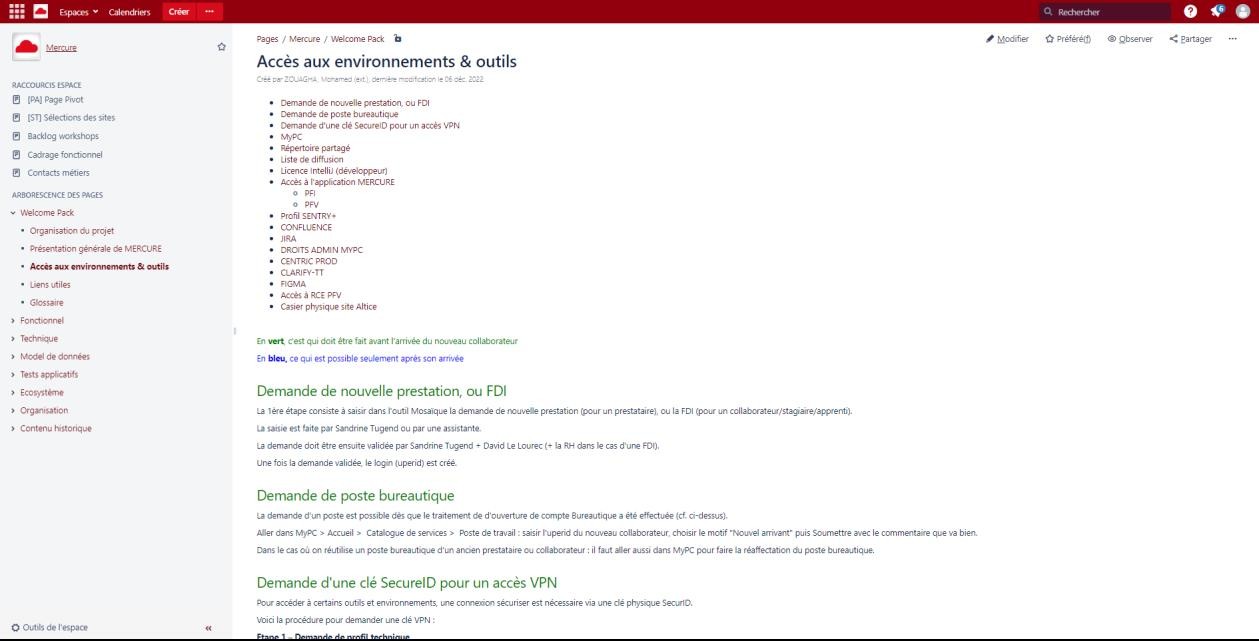


Figure 18 : Confluence Mercure

* 1. **Outils Technique**
     1. **Oracle SQL Developer**

Oracle SQL Developer est un environnement de développement graphique et gratuit, proposé par Oracle, spécialement conçu pour travailler avec des bases de données Oracle. Cet outil permet aux développeurs et aux administrateurs de bases de données de gérer, de développer et de déployer des bases de données Oracle de manière efficace. Oracle SQL Developer offre une interface conviviale pour exécuter des requêtes SQL, concevoir des schémas de base de données, créer des rapports, effectuer des analyses et gérer divers aspects de l'administration de la base de données. C'est un outil polyvalent qui facilite les tâches liées à la gestion et au développement des bases de données Oracle.



Figure 19 : Oracle SQL Developer Logo

* + 1. **Postman**

Postman est un outil de collaboration pour le développement d’API (Interface de programmation d’applications). Il s’agit d’une plateforme qui permet aux développeurs de tester, de déboguer et de documenter leurs API de manière efficace.

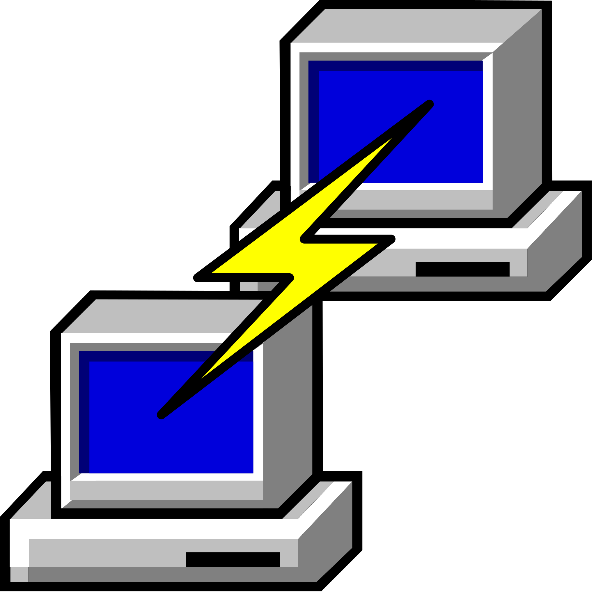
Avec Postman, les développeurs peuvent envoyer des requêtes HTTP (GET, POST, PUT, DELETE, etc.) à des endpoints spécifiques d’une API pour interagir avec celle-ci. L’outil fournit une interface conviviale qui permet de personnaliser les paramètres des requêtes, d’ajouter des en-têtes, d’inclure des données de requête, et de voir les réponses renvoyées par l’API.



* + 1. **PuTTY**

Figure 20 : Postman Logo

PuTTY est un logiciel open-source et gratuit de terminal, d'émulation de terminal et de transfert de fichiers réseau. Il est largement utilisé pour établir des connexions sécurisées via SSH (Secure Shell) vers des serveurs distants, ce qui permet aux utilisateurs d'accéder et de gérer des systèmes à distance de manière sécurisée. PuTTY prend en charge plusieurs protocoles de communication, notamment SSH, Telnet, Rlogin, et offre des fonctionnalités telles que la gestion des clés SSH, la gestion de sessions, et la possibilité de transférer des fichiers via SCP (Secure Copy Protocol) et SFTP (SSH File Transfer Protocol).



# 4. Conclusion

Figure 21 : PuTTY Logo

En conclusion, le troisième chapitre de ce rapport met en évidence l’importance capitale des tests rigoureux dans le cadre de notre projet. En comprenant le principe fondamental des tests et en explorant le processus détaillé qui les accompagne, nous sommes en mesure d’évaluer de manière approfondie le fonctionnement et l’efficacité de notre projet. Les technologies spécifiques utilisées pour soutenir cette démarche jouent un rôle crucial dans notre quête d’une conception solide et d’une amélioration continue. Grâce à cette exploration approfondie, nous avons pu saisir l’impact significatif de ces éléments sur notre projet. En somme, cette étude nous offre une compréhension approfondie des fondements de sa conception et nous fournit des bases solides pour poursuivre son développement avec confiance et efficacité. Cela va nous permettre de mieux comprendre le prochain chapitre qui est la partie réalisation.

**Chapitre 4 : Mise en œuvre du plan de test**

# Introduction

Ce chapitre présentera de manière détaillée les différentes étapes de réalisation du projet, en mettant l’accent sur les résultats obtenus, les enseignements tirés et les éventuelles pistes d’amélioration. Il s’agit d’une occasion de faire le bilan de mon expérience de stage, de partager mes réalisations et de mettre en avant les compétences que j’ai développées au cours de cette période enrichissante.

# Le projet Mercure

Le projet Mercure présente une réponse stratégique à plusieurs enjeux essentiels. Il se présente comme un outil exhaustif dédié à la gestion optimisée du parc, offrant une solution versatile compatible avec divers canaux de vente et structures organisationnelles. L'outil se distingue par son approche intuitive et fluide, conçue pour une utilisation simplifiée.

## Principes généraux

Les principes généraux de MERCURE se dévoilent sous différentes facettes :

**-Calcul d'éligibilité instantané :** L'outil réalise des calculs d'éligibilité en temps réel, en exploitant l'ID R@ comme base pour garantir des réponses rapides et précises.

**-Visibilité étendue des liens en parc :** Allant au-delà de la notion de MID, MERCURE offre une vue complète des liens en parc, apportant une profondeur accrue à l'analyse.

**-Optimisation back office avec mise en cache :** Pour améliorer l'expérience utilisateur, les traitements en back office sont judicieusement mis en cache, garantissant une navigation fluide et rapide.

**-Actes simples et guidés :** L'outil se démarque par des actes simples et guidés, simplifiant la prise en main, même pour les utilisateurs novices.

**-Gestion exhaustive de l'Univers Data Fixe :** MERCURE englobe tous les actes de gestion de l'Univers Data Fixe, créant ainsi une solution complète et centralisée.

**-Interaction optimale avec le catalogue :** Les interactions avec le catalogue sont optimisées, assurant une cohérence et une pertinence constantes.

**-Réutilisation des objets eCommande :** Les objets eCommande sont réutilisés pour la persistance des cotations, assurant une continuité et une efficacité des processus.

**-Restitution des données sans interprétation :** MERCURE présente les informations du parc sans interprétation, se limitant strictement aux données disponibles dans les BDS RCE pour une précision incontestable.

**-Traitement global des offres et actes :** L'outil traite de manière globale toutes les offres et les actes de gestion, créant un environnement complet et unifié.

**-Utilisation de services communs Pléiade :** MERCURE tire parti de services pléiade communs avec l'espace client, garantissant une cohérence à travers les systèmes.

## Processus métier

Au cœur du processus métier synthétisé, MERCURE forge un chemin fluide et efficace, apportant des fonctionnalités essentielles pour une gestion simplifiée. Le processus débute par l'accès de l'utilisateur via SESAME en Single Sign-On (SSO), garantissant une entrée sécurisée. MERCURE se distingue ensuite en offrant un accès inégalé aux données du parc client, tirant parti des services Pléiades dans la Base Front, avec une perspective de les intégrer ultérieurement dans la base du parc installé. La granularité est au rendez-vous grâce à l'inclusion de référentiels d'adresse.

Un élément clé se trouve dans les calculs d'éligibilité, orchestrés en temps réel sur les sites de cotation, avec la sauvegarde de résultats dans un cache pour une performance continue. L'intégration à eCommande prend le relais dans la phase de configuration des accès, assurant ainsi une interaction transparente avec le Catalogue HOMERE. Les fonctionnalités de MERCURE s'étendent également au processus de devis, aux workflows de remise, et à la signature électronique, simplifiant et accélérant le traitement des commandes.

L'interaction ne s'arrête pas là MERCURE se connecte harmonieusement avec ARTEMIS, contribuant à l'alimentation fluide des composantes en aval du système d'information (RCE et ADONIX dans une architecture en "Y"). Cette connexion stratégique élargit la portée de l'outil tout en contribuant à une architecture globale et intégrée.

Comme vous pouvez le voir ci-dessous, la figure représente L’architecture macro du projet :

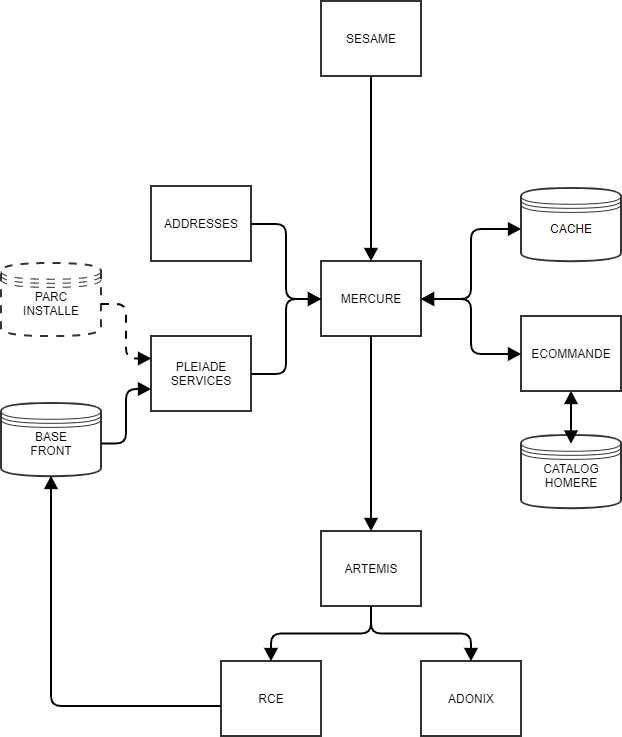


Figure 22 : Architecture Macro du Mercure

## Le périmètre fonctionnel

Le périmètre fonctionnel du projet englobe un éventail complet d'offres en matière de Data et Voix Fixe. Les offres majeures telles que :

**IPNet :** Offrant une connectivité IP hautement performante pour les entreprises, IPNet assure des communications fluides et sécurisées.

**Internet Connect :** Cette offre fournit des solutions de connectivité fiables, répondant aux besoins variés des entreprises pour une communication efficace.

**Lan2Lan :** Cette solution crée des liaisons réseau dédiées entre sites, favorisant une connectivité directe et sécurisée.

**Telephonie Fixe Voip :** C'est une offre de voix fixe qui vise à remplacer l'offre OFFICE. Aussi appelée téléphonie fixe VOIP.

**Ligne Business IP :** Cette offre offre des lignes IP dédiées, optimisant les capacités de communication et de transfert de données.

**TRUNKSIP :** Offrant des solutions avancées de téléphonie IP, TRUNKSIP simplifie les communications vocales tout en garantissant une qualité élevée.

Mercure permettra également de passer des commandes multi-offres et multi-ADG en combinant les autres offres et ADG actuellement disponibles dans eCommande :

**Univers mobile :** Souscription, RM, CPTA, RMCPTA, RM Eco Avantage.

**Univers convergent :** souscription lignes et accès PBE

**Univers services :** souscription SIS Evol, Collaboration office 365, Cyberprestations, AVS

## Actes de gestion sur les offres

Au sein de ce panorama des actes de gestion appliqués à ces offres spécifiques, un éventail de procédures se dévoile pour garantir une manipulation précise et cohérente des services. Cela englobe des actions cruciales telles que la souscription initiale, l'ajout stratégique de liens de secours pour renforcer la résilience, ainsi que des modifications variées, allant de la mise à niveau avec ou sans reconstruction de lien, jusqu'aux ajustements d'options. De plus, la prise en compte du déménagement se positionne également comme un volet crucial de ces actes de gestion, assurant une flexibilité et une continuité optimales pour répondre aux besoins évolutifs des clients.

## Valeur ajouter des Tests Fonctionnels

La valeur ajoutée intrinsèque des tests techniques et fonctionnels dans le contexte du projet Mercure s'avère particulièrement remarquable. Ces tests jouent un rôle clé en garantissant la robustesse, la cohérence et la convivialité de l'outil. Les tests techniques assurent la fiabilité des calculs d'éligibilité en temps réel ainsi que la gestion efficace du cache, contribuant ainsi à une expérience utilisateur fluide et réactive. Les tests fonctionnels, quant à eux, examinent minutieusement les différentes fonctionnalités de MERCURE, de la phase de souscription jusqu'à la modification des offres et les déménagements.

L’intégration de tests techniques et fonctionnels ajoute une couche de confiance essentielle, garantissant que MERCURE répond non seulement aux besoins des utilisateurs, mais excelle également dans la performance, la cohérence et la pertinence opérationnelle.

# La refonte et évolution des comptes de facturations

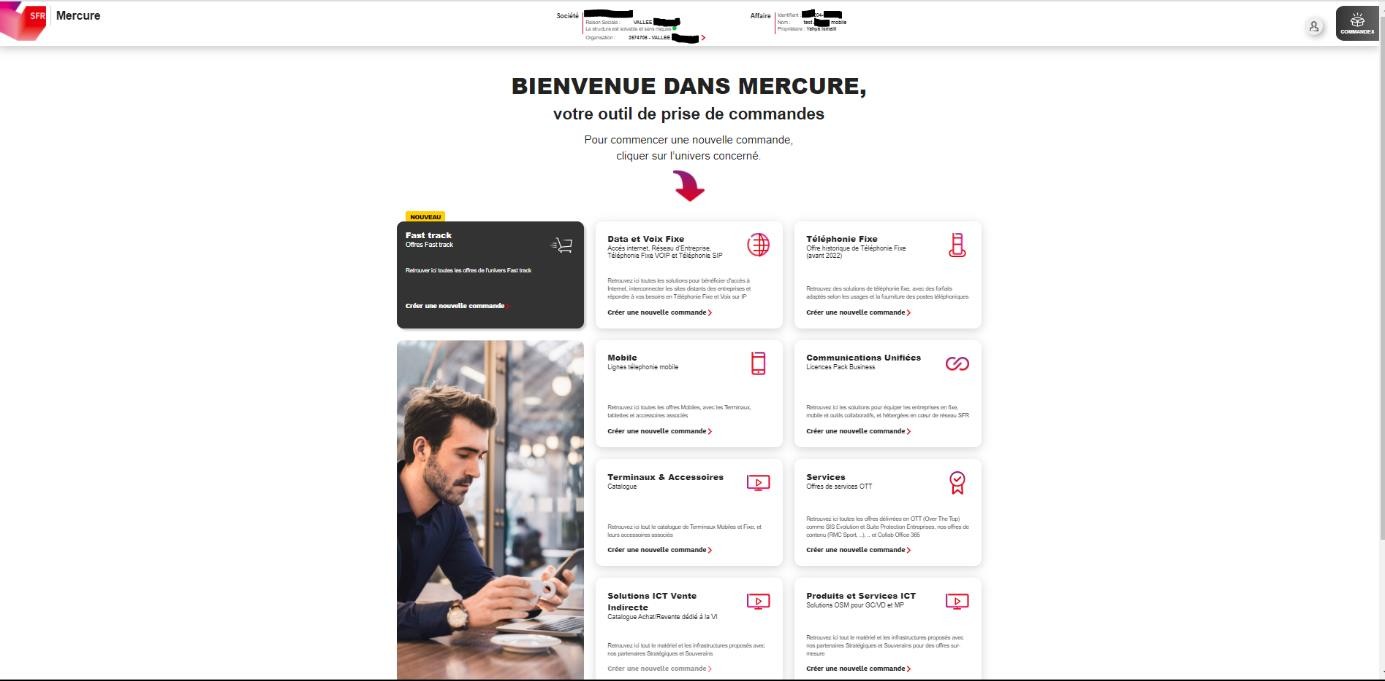
## La Refonte Compte de Facturation

Cette Refonte vise à remplacer les anciens systèmes de commande, Homere et Ecommande, qui étaient respectivement utilisés pour les canaux de vente directe et indirecte. En repensant l'outil de commande dans sa globalité, SFR cherche à améliorer l'efficacité opérationnelle, à unifier les canaux de vente et à offrir une expérience utilisateur optimale aux Ingénieurs Commerciaux.

## Nouvelles Pages et Interfaces

Nous explorerons les nouvelles pages et interfaces qui ont été intégrées dans le projet, introduisant ainsi des améliorations significatives pour les utilisateurs :

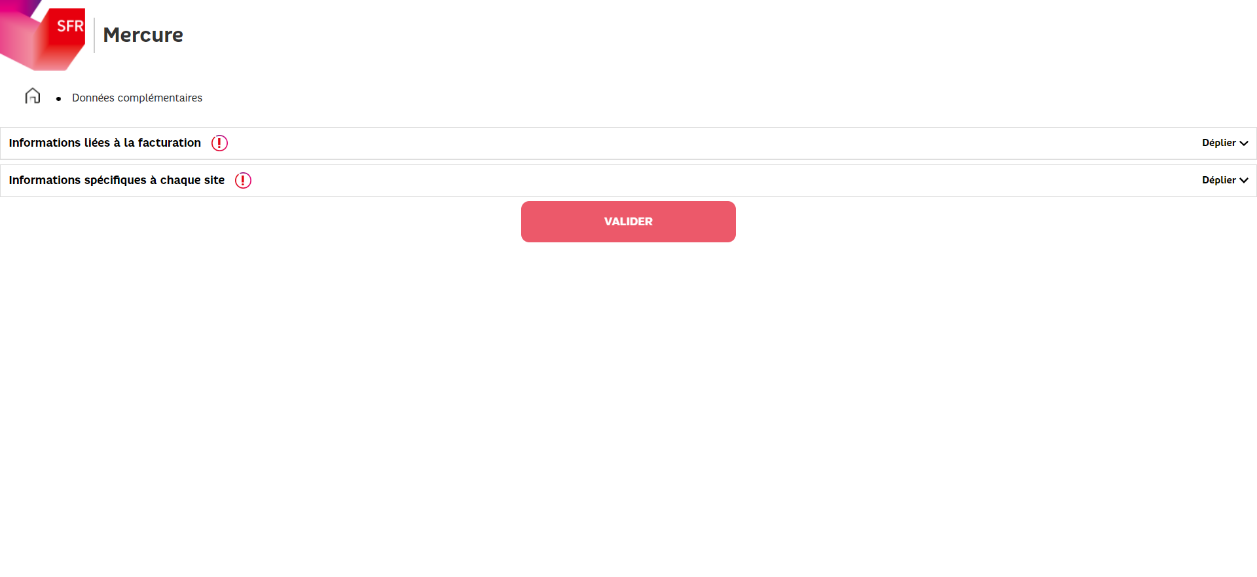
### Interface Mercure



### Interface compte de facturation

Figure 23 : Interface Mercure

-L’ancien Interface IHM du compte de facturation :

Figure 24 : Ancien IHM Compte de facturation 

* Le nouveau IHM de compte de facturation :



Figure 25 : Nouveau IHM Compte de facturation

* + 1. **Interface d’ajout d’un contact**

-Ancien Page d’ajout d’un contact :

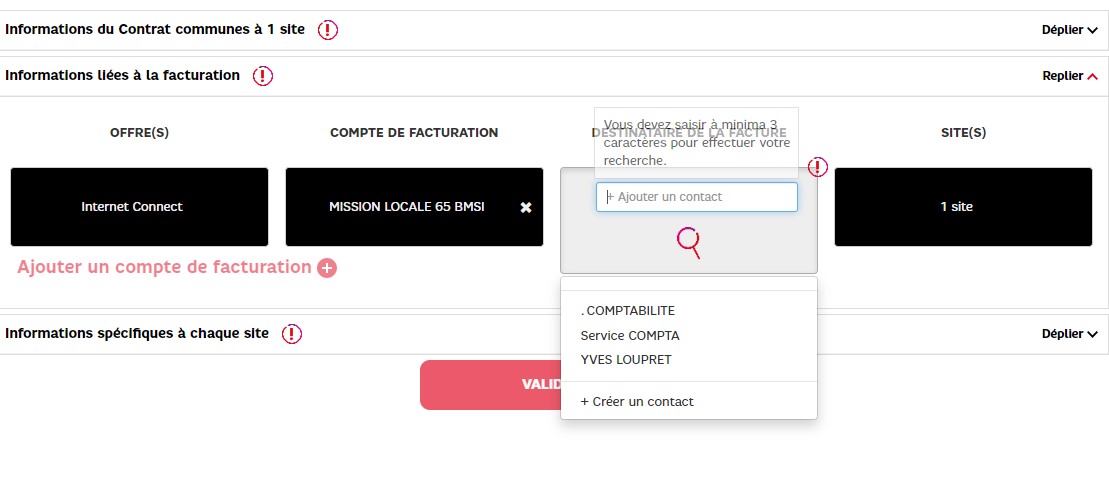


Figure 26 : Ancien page d'ajout d’un contact

- Nouvelle page d’ajout d’un contact :

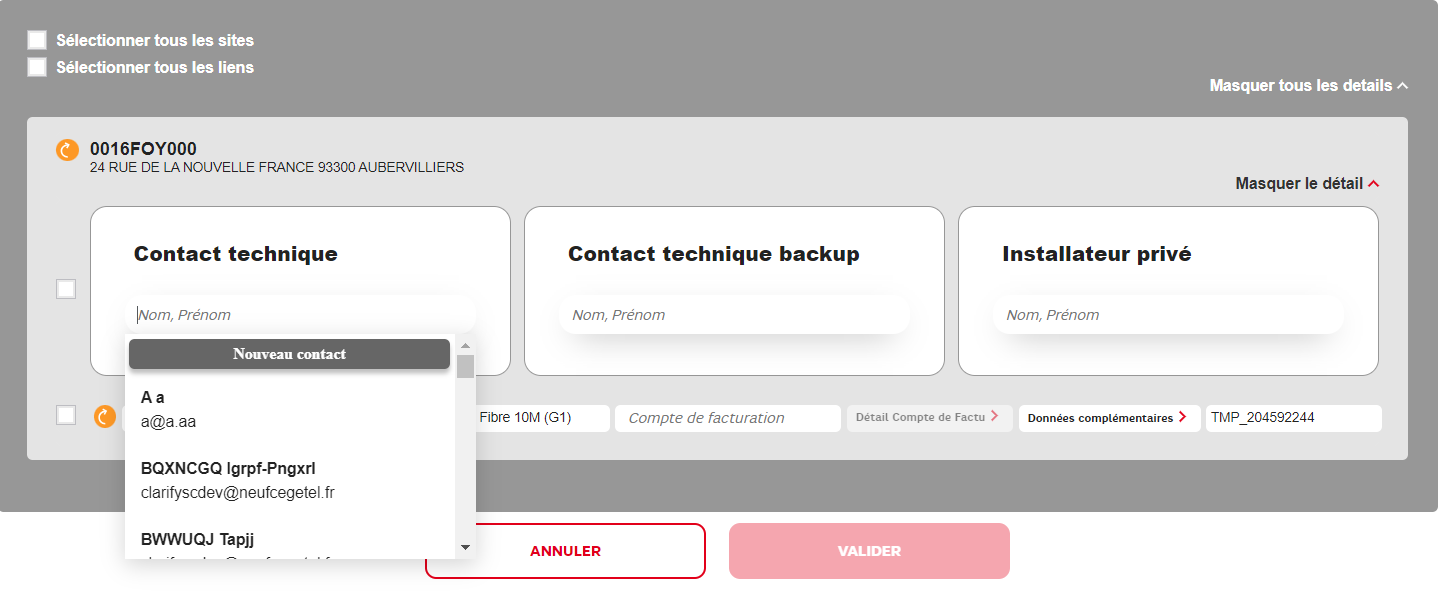
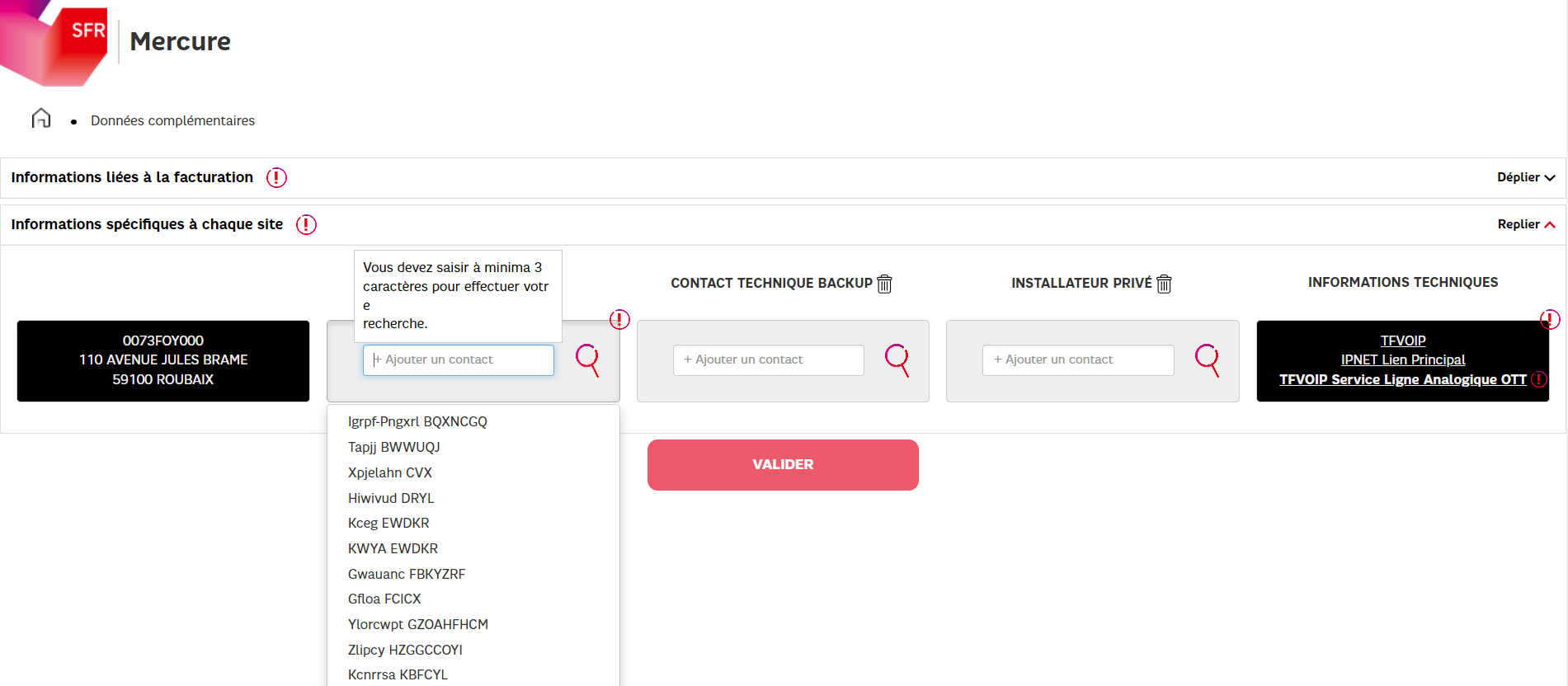


Figure 27 : Nouvelle page d’ajout d’un contact

* + 1. **Interface d’ajout d’un compte de facturation**

-Ancien interface d’ajout d’un compte de facturation :



-Nouveau interface d’ajout d’un compte de facturation :

Figure 28 : Ancien interface d’ajout d’un compte de facturation

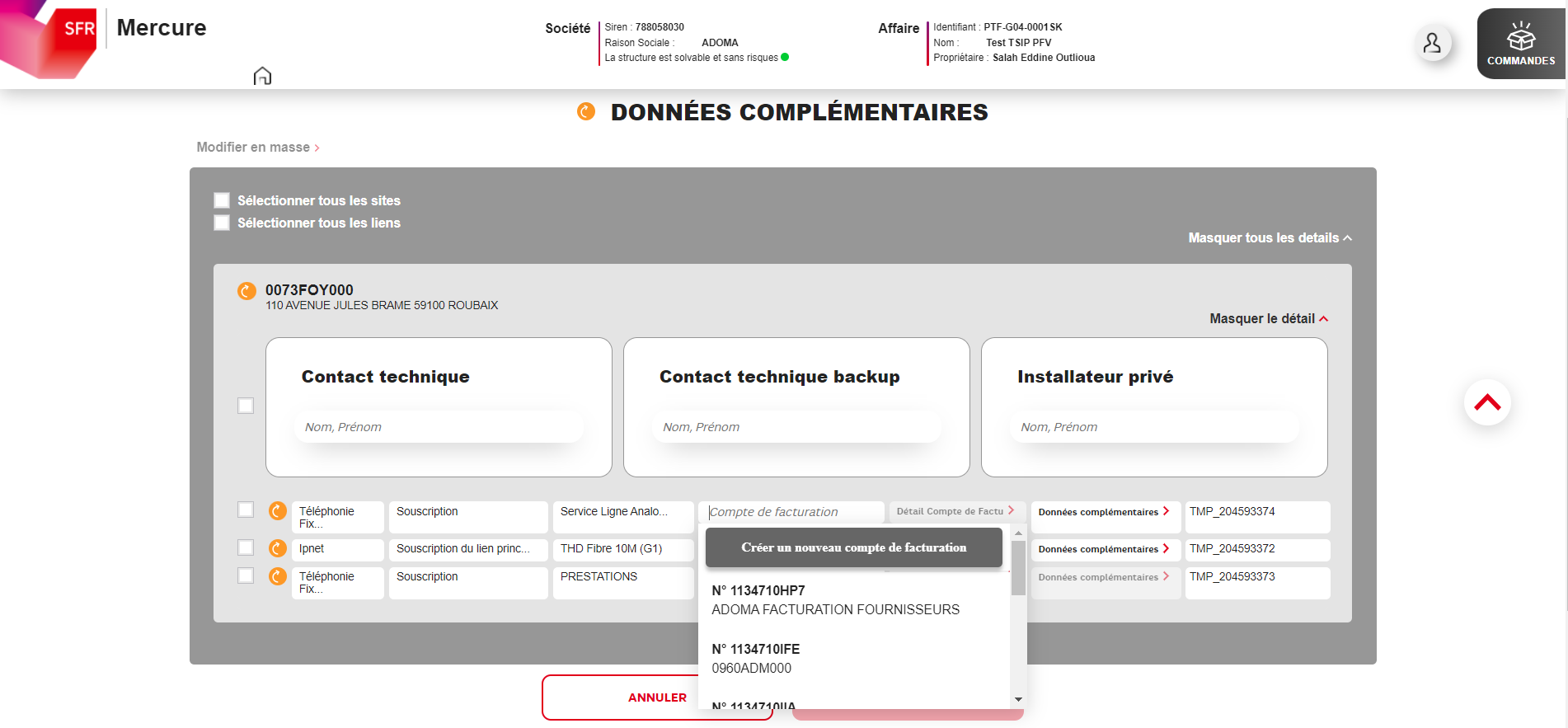


Figure 29 : Nouveau interface d’ajout d’un compte de facturation

* 1. **Fonctionnalités et offres ajoutées**

Nous mettrons en lumière les multiples ajouts et améliorations qui ont enrichi le projet, témoignant de son évolution continue pour répondre aux besoins changeants de ses utilisateurs

Parmi ces fonctionnalités ajoutées durant ma période de stage et les sujets que j'ai testés, certaines méritent une attention particulière :

#### - Modifier en masse :

La modification en masse dans l’univers data et voix fixe ainsi que dans l’univers mobile désigne la capacité d'associer un contact technique, un contact technique de backup ou un installateur privé à tous les sites ou à tous les liens d'une organisation en sélectionnant cette organisation pour lui affecter le même compte de facturation. Les utilisateurs ont la flexibilité de personnaliser leur expérience en fonction de leurs besoins spécifiques.

* 1. **Notion des future Flipping**

La notion de "future flipping" est une approche innovante qui repense fondamentalement notre relation avec le futur. Elle nous exhorte à abandonner l'approche passive de l'attente du futur pour devenir des acteurs proactifs dans sa création.

#### La Transition d'une Attitude Passive à Proactive :

Contrairement à une vision traditionnelle du futur, le "future flipping" incite à la prévision stratégique et à l'anticipation des tendances émergentes. Il nous pousse à reconnaître les opportunités cachées et à concevoir des solutions avant même que les problèmes ne surgissent. Cette approche transforme ainsi le futur en une toile blanche sur laquelle nous pouvons peindre nos aspirations.

#### La Vision et la Planification :

Au cœur du "future flipping" se trouve une vision claire du futur que nous désirons atteindre. Cette vision devient le catalyseur de notre planification proactive. En identifiant les étapes nécessaires pour concrétiser cette vision, nous pouvons guider nos actions présentes de manière à créer l'avenir que nous souhaitons.

#### L'Adaptabilité Continue :

Une composante cruciale du "future flipping" réside dans notre capacité à ajuster constamment nos plans en fonction des évolutions du monde qui nous entoure. L'adaptabilité continue est essentielle pour assurer que notre vision du futur reste alignée sur la réalité changeante.

1. **Mise en Œuvre du plan**
   1. **Plan de Test**

Nous décrivons en détail notre approche pour garantir la fiabilité et la performance de l'application Mercure. Le plan de test vise à identifier les points faibles, à valider les fonctionnalités clés et à assurer une expérience utilisateur optimale.

Parmi les principaux récits pour lesquels j'ai créé des scénarios figurent :

#### Alimentation de la gestion électronique des documents (GED) hors trames :

Cette récit vise à modifier la transmission des pièces jointes (PJ) vers la GED pour les injections ARTEMIS, PLEIADE et PLEIADE\_WS. Auparavant, les PJ étaient intégrées directement dans les trames, ce qui posait des problèmes de taille de trame, de manipulation des données et de duplication des documents dans la GED.

Les modes de transmission actuels sont :

* **IEM** : PJ envoyées à la GED et déposées sur le NAS RCE.
* **ARTEMIS** : PJ encodées en base64 dans la trame, puis envoyées à la GED.
* **PLEIADE\_WS** : PJ encodées en base64 dans la trame et déposées sur le NAS RCE.
* **PLEIADE\_WS\_SERVICES** : PJ encodées en base64 dans la trame et déposées sur le NAS RCE, avec référence à la GED.
* **PLEIADE** : PJ encodées en base64 dans la trame et déposées sur le NAS RCE, avec référence à la GED.

L'objectif est que Mercure envoie les PJ de manière unitaire vers la GED et fournisse un documentId dans les trames au lieu du contenu en base64. Cette évolution impactera principalement les injections ARTEMIS et IEM.

#### Les déploiements complexes :

L'objectif est d'aider les équipes déploiement à fournir aux gros clients une vision synthétique de l'avancement du déploiement de leurs commandes, dans le cadre de commandes complexes.

Le tableau suivant présente en détail les différents scénar

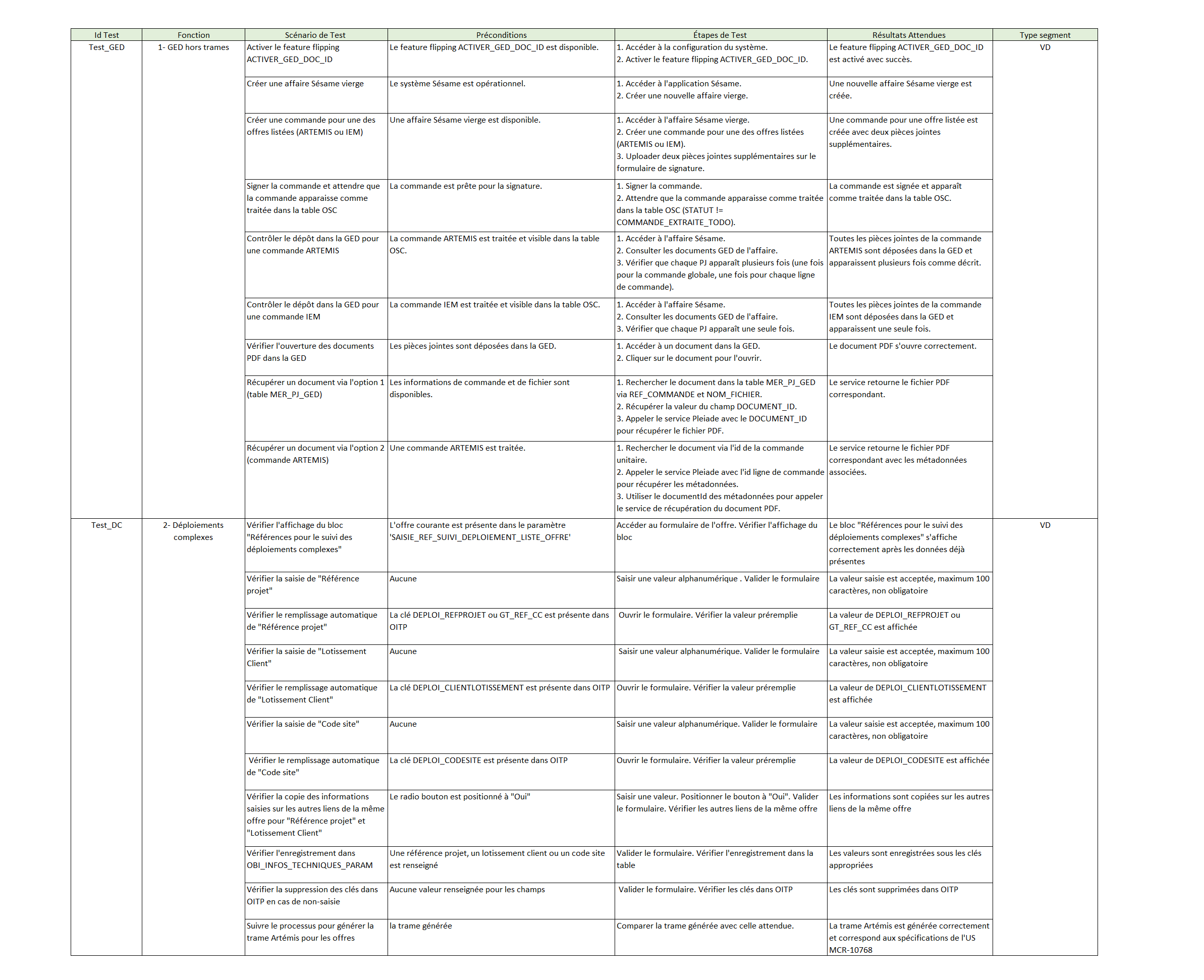


Figure 30 : Tableau Plan de test

* 1. **Exécution des Tests**

Pour exécuter nos tests, tout d'abord, nous vous expliquerons les différents environnements dans notre projet Mercure.

**-Environnements de Développement et de Validation dans Mercure :** Dans le cadre du projet Mercure, la gestion efficace des mises en production (MEP) repose sur une approche rigoureuse de développement et de validation. Pour atteindre cet objectif, notre équipe utilise deux environnements distincts et complémentaires : la Plateforme d’intégration (PFi) et la Plateforme de Validation (PFV).

**-La Plateforme d’intégration PFI :** La PFi constitue la première étape cruciale dans notre processus de déploiement et de test. Elle sert principalement de terrain de jeu pour les testeurs et les développeurs, où les nouvelles fonctionnalités et les modifications sont d'abord introduites.

L'exécution des tests a débuté par l'accès à l'application via SESAME en utilisant Single Sign-On (SSO) pour créer notre affaire. Une fois dans l'environnement Mercure, nous avons eu accès à l'ensemble des offres disponibles. L'interface de Mercure s'est déployée devant nous, nous permettant de parcourir les différentes offres proposées. Voici quelques points clés de la PFi :

* **Environnement de Test Initiatique :** La PFi est l'endroit où j'ai initié mes essais sur la dernière version du sprint en cours (sprint N). Cela m'a permis de m'assurer que les nouvelles fonctionnalités sont fonctionnelles et ne perturbent pas les fonctionnalités existantes.
* **TNR (Test Non Régressif) :** Les TNR sont effectués sur la PFi pour vérifier que les modifications apportées n'ont pas d'effet indésirable sur les composants déjà existants de l'application. Cela garantit la stabilité du produit.
* **Validation par les Testeurs :** Une fois la version installée sur la PFi, on effectue des tests approfondis pour valider la qualité et la conformité des fonctionnalités par rapport aux spécifications.

Le schéma ci-dessous illustre la séquence des étapes de test en PFI :

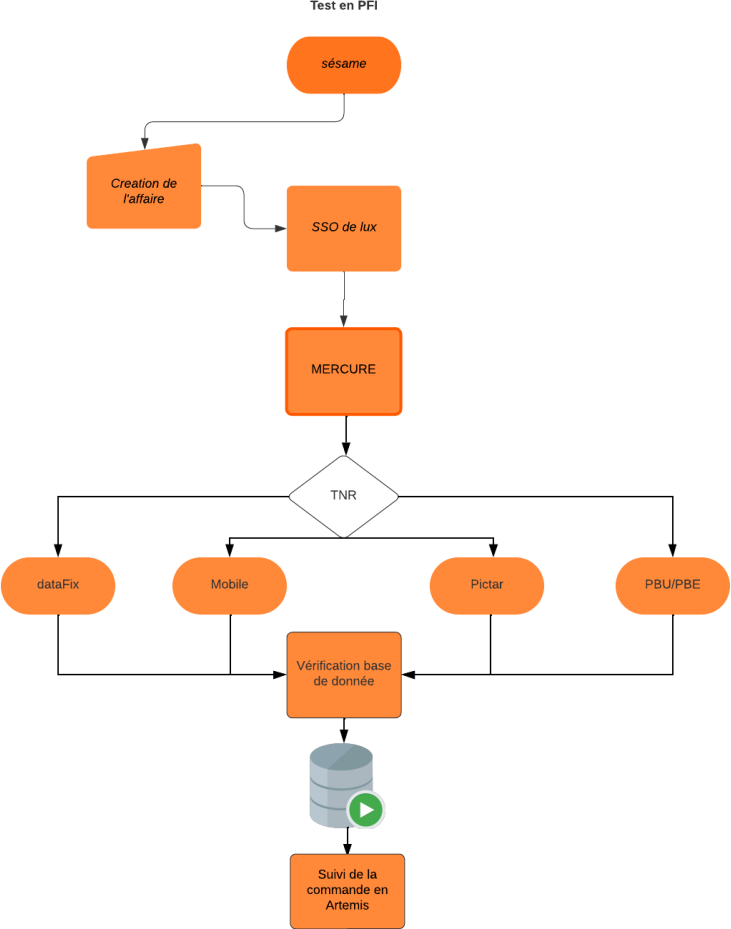


Figure 31 : Diagramme BPMN des tests en PFI

**-La Plateforme de Validation (PFV) :** Après avoir obtenu la validation des testeurs sur la PFi, la version est ensuite déployée sur la Plateforme de Validation (PFV). Cette plateforme offre un environnement similaire à la production et constitue la dernière étape avant la mise en production réelle.

Pour assurer la cohérence et la précision des résultats, nous avons vérifié la base de données, spécifiquement la table "Obi\_one\_desc", afin de suivre l'évolution du service Artemis. Notre objectif était de confirmer que la commande avait été correctement traitée et avait atteint l'étape du paiement, garantissant ainsi le bon déroulement du processus. Voici quelques points clés de la PFV :

* **Tests Approfondis :** Sur la PFV, une série de tests rigoureux est effectuée pour garantir que la version est prête pour la mise en production. Les tests incluent souvent des scénarios de charge, de sécurité et de performance.
* **Assurance de la Qualité :** La PFV vise à assurer la qualité, la stabilité et la performance du logiciel avant qu'il ne soit déployé en production. Tout problème ou bogue identifié est corrigé avant la livraison finale.

Le schéma ci-dessous illustre la séquence des étapes de test en PFV :

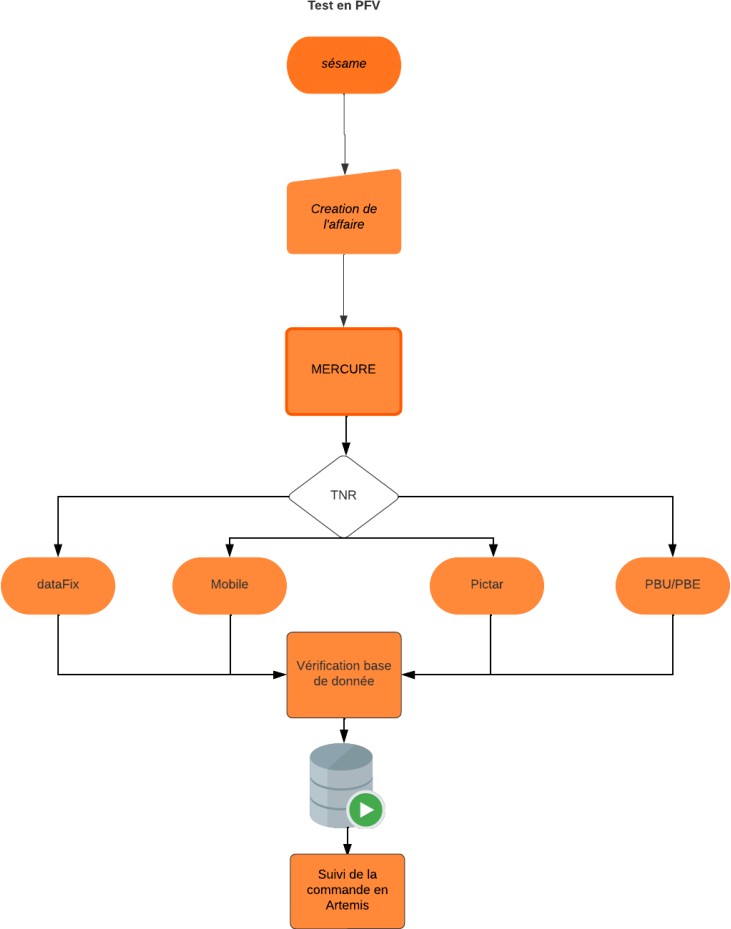


Figure 32 : Diagramme BPMN des Tests en PFV

* 1. **Évolution des tests de non-régression**

L'évolution de notre plan des tests de non régression s'est avérée nécessaire à la suite des développements récents du projet, en particulier l'intégration des fonctionnalités de la GED hors trame et les déploiement complexes. Ces nouvelles fonctionnalités ont apporté des changements significatifs à l'application, ce qui a conduit à une adaptation de notre stratégie de test.

Dans cette phase d'évolution, nous avons réexaminé notre plan des tests de non régression existant pour inclure des scénarios spécifiques à ces nouvelles fonctionnalités. Nous avons pris en compte les exigences fonctionnelles et techniques associées à la GED hors trame et les déploiement complexes., en veillant à ce que nos tests couvrent de manière exhaustive ces fonctionnalités ajoutées.

Le tableau suivant présente le scénario ajouté de la GED hors trame dans notre plan de test de non régression :



Figure 33 : Mise à jour du plan de test

1. **Gestion des Tests et des Tickets sur la PFV**

Au sein du projet Mercure, l'étape cruciale des tests fonctionnels et, parfois, techniques sur la Plateforme de Validation (PFV) est confiée à chaque testeur.

Cette section mettra en lumière le rôle essentiel que je joue en tant que testeur dans ce processus, où je reçois des tickets personnalisés via l'outil Jira, correspondant à une variété de sujets, tels que de nouvelles fonctionnalités, des bogues à corriger et des user stories Dans les figures suivantes, je présenterai quelques exemples de tickets, illustrant ainsi différentes situations et cas d'utilisation :

Le ticket représente une user story relative à la gestion des options, que j'ai testée et validée avant de procéder au déploiement de ces changements en production.

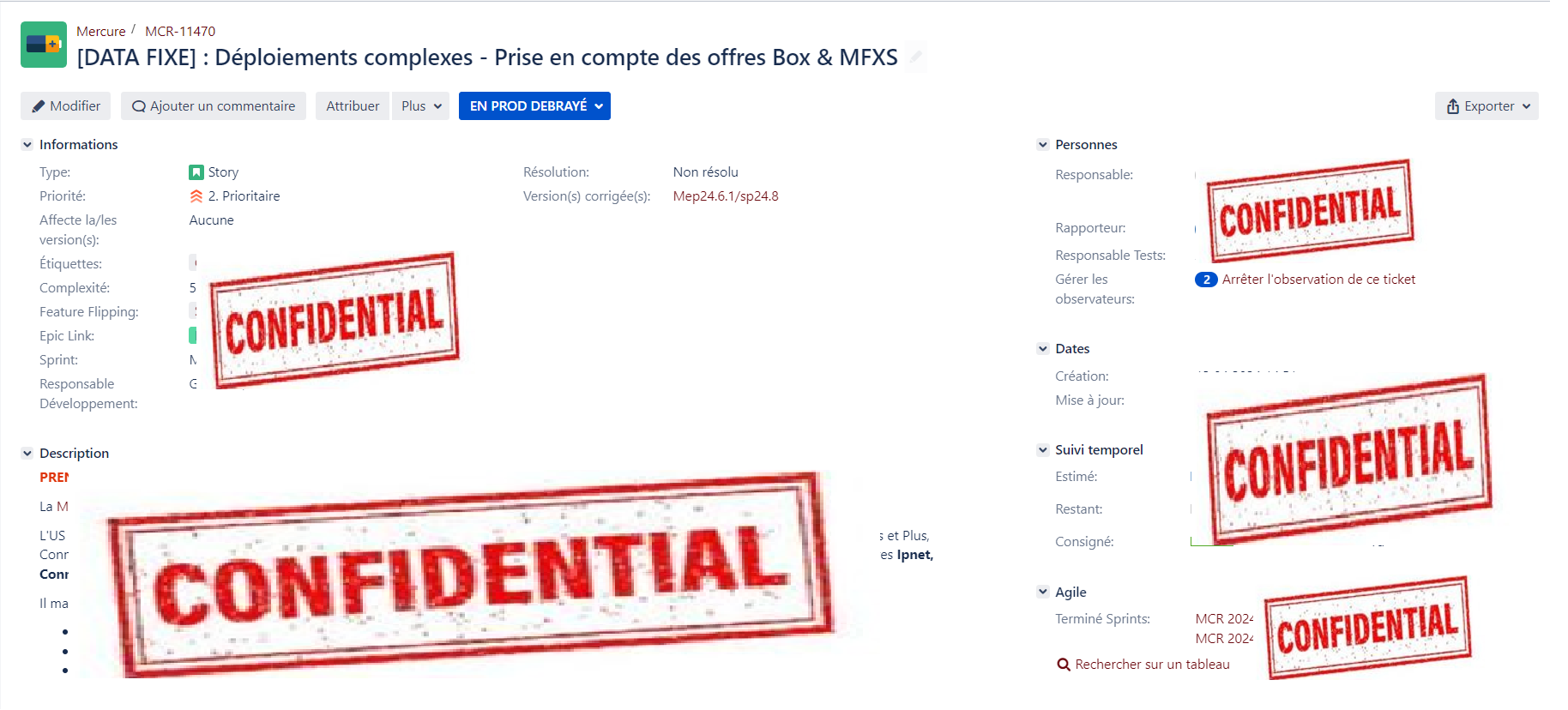


Figure 34 : Exemple de ticket jira

Dans le même contexte, j'ai identifié un bogue que j'ai documenté en créant un ticket Jira lors du sprint N. Par la suite, j'ai effectué des tests après sa correction au cours du sprint N+1 :



Figure 35 : Exemple d'un jira bogue

Mon rôle consiste à garantir que chaque composant du logiciel est rigoureusement testé, contribuant ainsi à la qualité globale du produit.

Après avoir examiné et testé ces tickets, nous explorerons également le processus de création de rapports quotidiens pour suivre l'état global des recettes, en utilisant différents statuts pour indiquer l'avancement et la résolution des problèmes.

La figure suivante présente le dernier rapport de résultats des tests que j'ai envoyé après avoir terminé les tests :



Figure 36 : exemple rapport des tests

Cette section détaille ma contribution au processus de tests fonctionnels et techniques, ainsi que la manière dont nous gérons et documentons notre progression vers la mise en production réussie de chaque version.

1. **Conclusion**

En concluant ce chapitre sur la gestion des tests et des tickets dans le projet Mercure. Il est clair que cette étape joue un rôle crucial dans l'assurance de la qualité et de la fiabilité de notre logiciel. En tant que testeur, ma responsabilité était de veiller à ce que chaque ticket attribué, qu'il s'agisse de nouvelles fonctionnalités, de bogues corrigés ou de user stories, soit minutieusement testé sur la Plateforme de Validation (PFV). La création de rapports quotidiens, avec des statuts précis, a facilité la communication au sein de l'équipe et a permis un suivi transparent de l'état de chaque recette.

**Conclusion générale et perspectives**

En conclusion, mon expérience au sein de INTELCIA dans le cadre de mon projet de fin d'études a été profondément enrichissante, tant sur le plan personnel que technique. Durant cette période, j'ai eu le privilège de collaborer avec une équipe compétente et passionnée, ce qui a grandement contribué à mon acquisition de compétences et de connaissances précieuses.

Sur le plan humain, j'ai été particulièrement impressionné par la culture de travail positive et collaborative qui prévaut chez INTELCIA. Les membres de l'équipe se sont toujours montrés prêts à partager leur savoir-faire et à m'apporter leur soutien quand nécessaire. Cette expérience m'a permis de comprendre l'importance cruciale de la communication efficace, de la collaboration au sein d'une équipe et de la capacité à résoudre des problèmes de manière collective. Ces compétences humaines acquises chez INTELCIA seront inestimables pour le reste de ma carrière professionnelle.

Sur le plan technique, INTELCIA m'a donné l'opportunité de mettre en pratique mes connaissances théoriques en me familiarisant avec des technologies de pointe. J'ai été confronté à des défis techniques complexes qui m'ont poussé à repousser mes limites et à trouver des solutions innovantes. Mon travail a principalement consisté à réaliser des tests fonctionnels et techniques, contribuant à la détection précoce d'anomalies et à l'optimisation du processus de test, ce qui s'est traduit par des gains de temps significatifs.

À l'issue de cette expérience enrichissante chez INTELCIA, je suis actuellement en pleine transition vers une nouvelle étape de ma carrière professionnelle. Je m'apprête à intégrer une autre équipe au sein d'un projet prometteur, où je débuterai ma carrière professionnelle en tant que testeur de bout en bout pour l'ensemble de la chaîne de commande, couvrant plusieurs services et projets. Ce nouveau défi s'annonce particulièrement exigeant, nécessitant une compréhension approfondie des technologies qui sous-tendent ces tests, ainsi qu'une expertise pour garantir la sécurité et la fiabilité du traitement des commandes, de leur initiation jusqu'au dernier service de la chaîne. Cela requiert non seulement de l'expérience, mais également une vision globale et une connaissance approfondie des différents maillons de la chaîne.

Ainsi, je me concentre actuellement sur l'acquisition de ces compétences techniques et la préparation nécessaire pour aborder ce prochain chapitre de ma carrière avec confiance et réussite.

Mon expérience chez INTELCIA m'a fourni des bases solides et des compétences essentielles qui me seront inestimables dans cette nouvelle aventure professionnelle.

**BIBLIOGRAPHIE**

-https://[www.intelcia.com/fr](http://www.intelcia.com/fr) : Présentation Intelcia

-https://confluence-jmsp.private.sfr.com/ : Présentation Mercure https://[www.lucidchart.com/](http://www.lucidchart.com/) : Les Diagrammes

-https://chat.openai.com/ : ChatGPT

- https://[www.google.com/](http://www.google.com/) : Google