



HAL
open science

Projet automatisé du test technique de non-régression comptable

Philippe Amerio

► **To cite this version:**

Philippe Amerio. Projet automatisé du test technique de non-régression comptable. Ingénierie, finance et science [cs.CE]. 2012. dumas-01220554

HAL Id: dumas-01220554

<https://dumas.ccsd.cnrs.fr/dumas-01220554>

Submitted on 26 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL ASSOCIE D'AIX-EN-PROVENCE

MEMOIRE

présenté en vue d'obtenir le

DIPLOME D'INGENIEUR C.N.A.M.

en **INFORMATIQUE** option **SYSTEMES D'INFORMATION**

par **Philippe AMERIO**

et soutenu le 11 Juillet 2012

**Projet Automatisation
du test technique de non-régression
comptable**

JURY

Président : **M. Yves LALOUM**

Membres : **M. Bastien PESCE**

M. Noël QUESSADA

M. Yannick TISON

Mme Katia CHAZEIRAT

RESUME

Ce mémoire a été réalisé au sein de la *Française Des Jeux* (FDJ), entreprise publique française ayant le monopole de la loterie, des jeux de grattage et de tirage en ligne sur tout le territoire. En plus de ces activités, la société a décidé depuis le début de l'année 2009 de lancer un site de **jeux en ligne** pour les **paris sportifs : ParionsWeb**¹. Le projet est appelé Modern Buy. La Française Des Jeux a fait appel à une société externe d'édition logicielle basée en Grande-Bretagne, nommée Laverock von Schoultz Limited (LVS), pour le développement de cette application de jeux en ligne. Cette société a depuis été rachetée par la Française Des Jeux. LVS s'occupe uniquement de la partie développement. Les autres métiers liés au projet tels que le marketing, les études, la gestion de projet, l'intégration, la qualification, la recette et la mise en production ont été alloués à des ressources internes. La partie intégration, qualification technique, mise en production et maintenance est assurée par l'équipe Homologation et Maîtrise Opérationnelle des Applications (HMOA) Multimédia Sport. A chaque livraison, le chef de projet HMOA et son équipe prennent en charge l'application pour vérifier l'adéquation du produit développé par rapport aux exigences. Le projet est basé sur le modèle du cycle en V².

Ce projet s'est terminé. Mais d'autres projets ont depuis été mis en place afin de faire évoluer l'application de jeux en ligne ABP³ : Modern Live, etc. pour plus de simplicité, nous utiliserons le terme *Modern* pour désigner le projet en cours au moment de l'automatisation.

L'équipe HMOA Multimédia Sport est composée de quatre personnes et des prestataires externes en complément pour la qualification technique de l'application. Je fais partie de ces prestataires qui sont en moyenne au nombre de trois et mon employeur est Altran Technologie, société de services en ingénierie. C'est dans ce cadre, que constitue la qualification technique de l'application ABP, que le projet automatisation du test technique de non-régression⁴ comptable a été mis en place.

En effet, nous verrons dans la suite de ce document les raisons qui ont amené à la réalisation des tests automatisés et notamment à l'automatisation du test comptable.

Le test comptable a pour but de vérifier les calculs des gains, des remboursements des joueurs ainsi que le calcul du chiffre d'affaires du site de ParionsWeb conformément aux règles comptables.

Ce projet d'automatisation qui ne devait initialement durer que deux mois, s'est finalement étendu sur plus d'un an et demi. Seule la période de septembre 2010 (première discussion) à juin 2011 (fin de la 3ème étape de réalisation) ne sera toutefois prise en compte dans le cadre de ce mémoire.

Le schéma ci-dessous situe l'équipe HMOA dans le cycle de vie du projet.

¹ Site accessible à l'adresse <https://parionsweb.fdj.fr/>

² Le modèle du cycle en V est un modèle conceptuel de gestion de projet.

³ Advanced Betting Platform (ABP) est le nom de l'application développée par LVS pour les paris sportifs en ligne.

⁴ Les tests de non-régression sont les tests d'un programme préalablement testé, après une modification, pour s'assurer que des défauts n'ont pas été introduits ou découverts dans des parties non modifiées du logiciel, comme suite à des modifications effectuées.

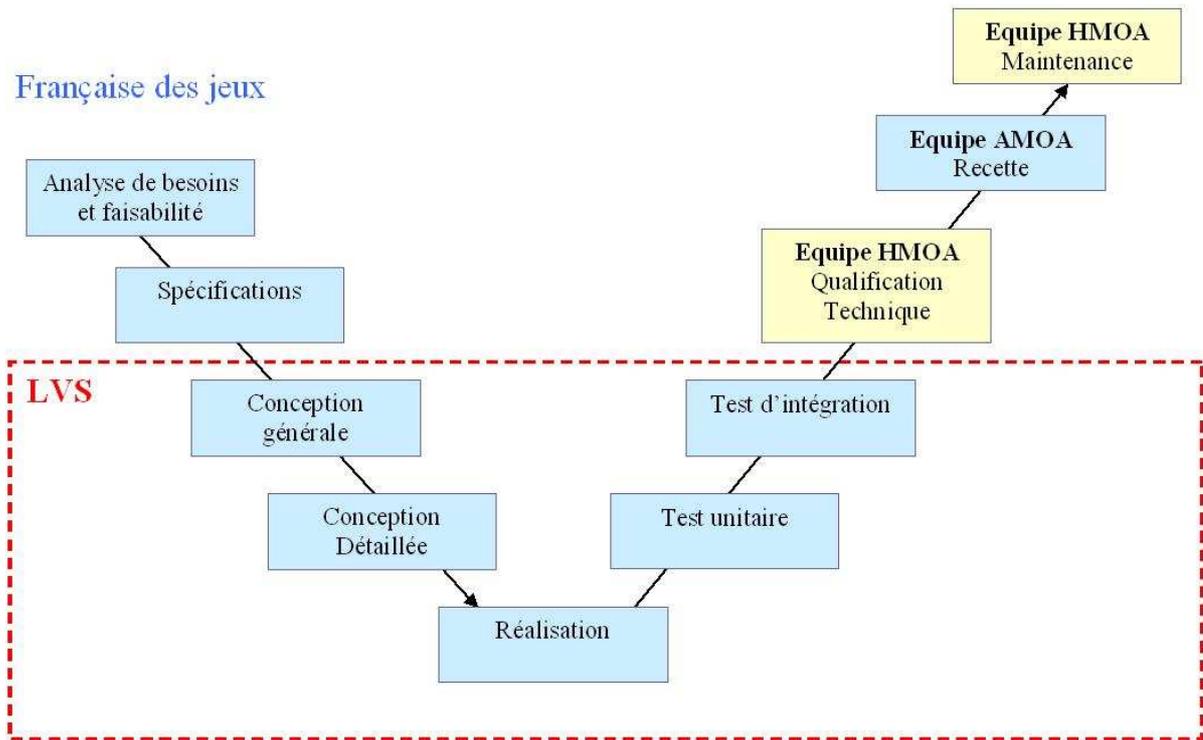


Figure 1-1 Schéma représentatif du Cycle de vie du projet *Modern*

Remerciements

Je tiens à remercier :

M. Tison Yannick, responsable HMOA Multimédia Sport à la Française Des Jeux Vitrolles, pour m'avoir fait confiance dans la gestion de ce projet, ainsi que pour ses conseils avisés et son investissement,

Mme Chazeirat Katia, manager Altran, pour m'avoir également fait confiance et recruté au sein d'Altran Technologie et de m'avoir permis d'effectuer ma mission au sein de la Française de Jeux,

M. Scalisi Laurent, responsable du service HMOA Multimédia, pour avoir validé ma participation à ce projet,

M. Blondeau Jérémie, responsable du projet *Modern*, pour avoir accepté que l'on utilise une partie du budget de qualification pour faire ce projet,

M. Huerta Stéphane, consultant Bull, pour ses précieux conseils en gestion de projet Scrum et son implication,

M. Al Oumami Amine, ainsi que Mlle Poissonnet Barbara pour leur travail et leur implication dans ce projet,

Mlle Gnanakulenthiran Sinduja, développeuse de chez LVS, pour son travail et sa disponibilité,

Tous les membres de l'équipe HMOA Multimédia Sport pour leur disponibilité et leur aide, M. Pesce Bastien pour son suivi et ses conseils dans la réalisation de ce mémoire,

M. Lombardi Jérôme et M. Borja Jean-Stéphane pour leurs aides dans la réalisation de ce rapport,

Ma famille et en particulier à ma femme pour m'avoir soutenu durant toutes ces années de formation au CNAM.

Pour finir, je remercie l'ensemble des membres du jury d'avoir accepté d'être présent le jour de ma soutenance.

SOMMAIRE

TABLES DES ILLUSTRATIONS.....	7
1 INTRODUCTION	9
2 CONTEXTE DE LA MISSION.....	11
2.1 ORGANISATION DE L'ENTREPRISE.....	12
2.2 PRESENTATION DE L'EQUIPE HMOA MULTIMEDIA SPORT.....	14
2.2.1 Organigramme de l'équipe HMOA Multimédia Sport.....	14
2.2.2 Activité de l'équipe.....	15
2.3 DESCRIPTION DU PROJET MODERN.....	15
2.3.1 Présentation de « ParionsWeb » et du fonctionnement des paris sportifs.....	16
3 LE PROJET AUTOMATISATION.....	20
3.1 PROBLEMATIQUE.....	21
3.2 ETUDE PRELIMINAIRE.....	21
3.2.1 Définition du besoin.....	21
3.2.2 Liste et analyse des contraintes.....	21
3.2.3 Risques.....	23
3.2.4 Analyse des outils existants.....	23
3.2.5 Proposition d'une architecture de type modulaire.....	24
3.2.6 Conclusion de l'étude.....	30
3.3 INITIALISATION DU PROJET.....	30
3.3.1 Analyse de la situation.....	30
3.3.2 Choix de la méthode.....	32
3.4 ANALYSE DETAILLEE.....	35
3.4.1 Analyse du fonctionnement du test comptable.....	35
3.4.2 Analyse technique comptable.....	42
3.4.3 Automatisation.....	43
3.4.4 L'« ordonnanceur ».....	47
3.5 PHASE DE REALISATION, D'INTEGRATION ET DE QUALIFICATION CONTINUE.....	60
3.5.1 Définition de Scrum.....	60
3.5.2 Mise en application de Scrum : utilisation des outils.....	62
3.5.3 Vue d'ensemble du projet.....	68
3.5.4 Première étape.....	71
3.5.5 Deuxième étape.....	78
3.5.6 Troisième étape.....	86
4 MISE EN APPLICATION.....	89
4.1 QUALIFICATION COMPTABLE.....	89
4.1.1 Inconvénients de l'automatisation comptable.....	92
4.1.2 Avantage dans notre projet.....	92
4.2 QUALIFICATION DE L'ARJEL.....	92
4.2.1 Description.....	92
4.2.2 Avantage de l'automatisation.....	92
4.2.3 Mise en œuvre.....	94
CONCLUSION.....	96
ANNEXES.....	98

ANNEXE 1 - PLAN DE RECETTE COMPTABLE.....	98
ANNEXE 2 - DEMANDE D'EVOLUTION POUR L'AJOUT DE FONCTIONNALITES DE PAIEMENT.....	100
ANNEXE 3 - ILLUSTRATION DE L'ORDONNANCEMENT DES MODULES <i>VIA QC</i>	103
ANNEXE 4 - EXTRAIT D'UN CODE VBSCRIPT MIS EN PLACE DANS QC	104
ANNEXE 5 - EXTRAIT DU CODE D'UN RAKEFILE	105
ANNEXE 6 - REPERTOIRE SVN UTILISE TOUT AU LONG DU PROJET	106
BIBLIOGRAPHIE	107

Tables des illustrations

FIGURE 1-1 SCHEMA REPRESENTATIF DU CYCLE DE VIE DU PROJET <i>MODERN</i>	3
FIGURE 2-1 ORGANIGRAMME DES SERVICES LIES A LA REALISATION DU PROJET PARIONSWEB (JANVIER 2011) .	13
FIGURE 2-2 ORGANISATION HMOA MULTIMEDIA SPORT EN DECEMBRE 2010.....	14
FIGURE 2-3 ORGANISATION HMOA MULTIMEDIA SPORT EN AVRIL 2011	14
FIGURE 2-4 SCHEMA SIMPLIFIE DU ROLE DE L'EQUIPE HMOA MULTIMEDIA SPORT.....	15
FIGURE 2-5 SCHEMA EXPLICATIF DE L'APPLICATION ABP	16
FIGURE 2-6 PAGE D'ACCUEIL DU SITE PARIONSWEB (JUN 2011)	17
FIGURE 2-7 ILLUSTRATION DE L'OUTIL « EVENT CREATOR ».....	19
FIGURE 3-1 SCHEMA MONTRANT L'AVANCEE DES DEVELOPPEMENTS A TRAVERS LE TEMPS	20
FIGURE 3-2 SCHEMA SIMPLIFIE DU TRANSFERT DES DONNEES ABP	22
FIGURE 3-3 SCHEMA DES DIFFERENTS NIVEAUX DE GRANULARITE UTILISES POUR L'ARCHITECTURE	25
FIGURE 3-4 MODELE FONCTIONNEL DE LA QUALIFICATION TECHNIQUE DE NON-REGRESSION D'ABP	26
FIGURE 3-5 SOUS-FONCTIONS UTILISEES POUR QUALIFIER LA COMPTABILITE	26
FIGURE 3-6 EXTRAIT DU PROCESSUS DE QUALIFICATION DE LA COMPTABILITE.....	27
FIGURE 3-7 SCHEMA DU PROCESSUS DE TEST COMPTABLE ET SES PRINCIPALES ACTIONS.....	29
FIGURE 3-8 SCHEMA DU PROCESSUS COMPTABLE	35
FIGURE 3-9 SCHEMA REPRESENTATIF DE L'ENCHAINEMENT DES ACTIONS DU TEST COMPTABLE SUR 4 JOURS	35
FIGURE 3-10 SCHEMA D'ANALYSE DES DONNEES COMPTABLES	37
FIGURE 3-11 SCHEMA DETAILLE DU PROCESSUS COMPTABLE ENTRE LES OPERATIONS ET LES CONTROLES	37
FIGURE 3-12 SCHEMA DU PROCESSUS COMPTABLE ET LES TYPES DE CONTROLES ASSOCIES	38
FIGURE 3-13 SCHEMA DETAILLE DU PROCESSUS AVANT CONTROLES ET LES TYPES DE CONTROLES ASSOCIES	38
FIGURE 3-14 SCHEMA TECHNIQUE DU TRANSFERT DES DONNEES COMPTABLES	39
FIGURE 3-15 SCHEMA TECHNIQUE DU TRANSFERT DES DONNEES COMPTABLES ET DES CONTROLES ASSOCIES	45
FIGURE 3-16 SCHEMA REPRESENTATIF DE L'ENCHAINEMENT DES ACTIONS DU TEST COMPTABLE SUR 4 JOURS	47
FIGURE 3-17 ILLUSTRATION DES ETAPES D'UN TEST DANS QUALITY CENTER.....	48
FIGURE 3-18 ILLUSTRATION CREATION D'UN TEST AUTOMATISE VBSCRIPT DANS QUALITY CENTER	49
FIGURE 3-19 ILLUSTRATION DU CHARGEMENT D'UN TEST DANS UN « TESTSET ».....	50
FIGURE 3-20 ECHANGE DES DONNEES ENTRE QUALITY CENTER ET LA MACHINE LOCALE	51
FIGURE 3-21 DIAGRAMME DE FLUX GESTION DES ERREURS	53
FIGURE 3-22 DIAGRAMME DE FLUX DU PROCESSUS DU PREMIER JOUR D'AUTOMATISATION COMPTABLE	55
FIGURE 3-23 DIAGRAMME DE FLUX DU PROCESSUS DU DEUXIEME JOUR D'AUTOMATISATION COMPTABLE	56
FIGURE 3-24 DIAGRAMME DE FLUX DU PROCESSUS DU TROISIEME JOUR D'AUTOMATISATION COMPTABLE.....	57
FIGURE 3-25 DIAGRAMME DE FLUX DU PROCESSUS DU QUATRIEME JOUR D'AUTOMATISATION COMPTABLE	58
FIGURE 3-26 DIAGRAMME DE FLUX DETAILLE DU PROCESSUS DE CONTROLE COMPTABLES AGP_ETL ET ETL.	59
FIGURE 3-27 VUE SYNTHETIQUE PROCESSUS SCRUM	61
FIGURE 3-28 ILLUSTRATION OUTIL DE SUIVI DE PROJET	65
FIGURE 3-29 ILLUSTRATION BURDOWN CHART	66
FIGURE 3-30 ILLUSTRATION MELEE QUOTIDIENNE	67
FIGURE 3-31 ILLUSTRATION PLANNING GLOBAL DU PROJET.....	69
FIGURE 3-32 SCHEMA DES ITERATIONS DU PROJET D'AUTOMATISATION COMPTABLE.....	70

FIGURE 3-33 SCHEMA ORGANIGRAMME DE L'EQUIPE LORS DE LA PREMIERE PHASE DU PROJET	71
FIGURE 3-34 SCHEMA DE L'ARCHITECTURE REST MISE EN PLACE PAR LVS	78
FIGURE 3-35 SCHEMA DE L'ORGANISATION DE L'EQUIPE LORS DE LA DEUXIEME PHASE DU PROJET	81
FIGURE 3-36 SCHEMA SIMPLIFIE DE L'ARCHITECTURE LOGICIELLE	82
FIGURE 3-37 SCHEMA DE L'ORGANISATION DU PROJET, DE TYPE SCRUM, MISE EN PLACE	88
FIGURE 4-1 SCHEMA DU PROCESSUS D'AUTOMATISATION COMPTABLE ET SES ACTIONS AUTOMATISEES	90
FIGURE 4-2 SCHEMA PRESENTANT LES ITERATIONS NECESSAIRE A L'AUTOMATISATION COMPTABLE	91
FIGURE 5-1 JEU DE DONNEES COMPTABLES EN ENTREE PLAN DE RECETTE V3	98
FIGURE 5-2 RESULTATS ATTENDUS POUR TOUS LES JOURS	99
FIGURE 5-3 ILLUSTRATION DE LA DEMANDE D'EVOLUTION (PART 1)	100
FIGURE 5-4 ILLUSTRATION DE LA DEMANDE D'EVOLUTION (PART 2)	101
FIGURE 5-5 ILLUSTRATION DE LA DEMANDE D'EVOLUTION (PART 3)	102
FIGURE 5-6 ILLUSTRATION DU PROCESSUS COMPTABLE AUTOMATISE DEPUIS UN « TESTSET » QC	103
FIGURE 5-7 EXTRAIT DU CODE SOURCE DU VBSCRIPT DANS QC APPELANT UN EXECUTABLE	104
FIGURE 5-8 EXTRAIT DU CODE SOURCE DU RAKEFILE UTILISE POUR LA CREATION DES EXECUTABLES.	105
FIGURE 5-9 ILLUSTRATION DU REPERTOIRE SVN UTILISE POUR LE SUIVI DE VERSION DES FICHIERS	106

1 INTRODUCTION

La création du site ParionsWeb a démarré en 2009 avec la mise en place du projet Modern Buy. Depuis, plusieurs autres projets *Modern* ont été initiés et chacun a contribué à faire évoluer l'application. L'équipe HMOA Multimédia Sport reçoit donc plusieurs fois durant l'année des nouvelles versions d'ABP qu'elle doit qualifier avant la mise en production.

Les membres de l'équipe HMOA Multimédia Sport en charge des projets *Modern* se doivent d'être très vigilants sur la qualité des livraisons pour chaque évolution de l'application ABP. Sans quoi des erreurs pourraient être découvertes en production. Pour cela, chaque membre de l'équipe travaille sur une partie précise de l'application afin de tester son fonctionnement et remonter au plus tôt les éventuels dysfonctionnements. Cependant, à chaque nouvelle livraison de l'application, des fonctionnalités que les membres de l'équipe HMOA Multimédia Sport doivent contrôler s'ajoutent et élargissent le périmètre des tests techniques.

D'autre part, la qualité des livraisons est très peu satisfaisante, ce qui complique grandement le travail de qualification. La plupart du temps, des correctifs sur des anomalies détectées sont livrés par LVS et nécessitent d'effectuer à nouveau des contrôles sur les fonctionnalités de base : authentification, prise de paris, paiement... L'application se complexifie et la grande fréquence de livraisons des correctifs ne laisse pas beaucoup de temps aux membres de l'équipe pour effectuer tous les contrôles nécessaires.

C'est pour remédier à ce manque de temps, que le responsable de l'équipe HMOA Multimédia Sport, a souhaité investir dans les nouvelles technologies de l'information. Cette volonté s'est traduite par le lancement du projet d'automatisation des tests techniques.

Dans un premier temps, nous verrons la démarche d'analyse préliminaire au projet, qui, à partir d'une analyse des besoins et de l'existant, nous aura permis d'apporter les éléments nécessaires à la prise de décision de démarrage du projet. Grâce à cette analyse, l'objectif *d'automatisation du test technique de non-régression comptable* a pu être clairement défini et les bases de l'architecture établies. Nous pourrions constater que les conditions dans lesquelles s'est déroulé le projet ont fortement influencé le choix de la méthodologie de gestion de projet.

Par la suite, nous regarderons de façon approfondie la mise en application de l'architecture logicielle ainsi que le fonctionnement du processus comptable qui, à eux deux, ont fourni tous les éléments nécessaires pour le démarrage des développements.

Une partie sera ensuite consacrée à la description de la méthodologie de gestion de projet mise en place, ainsi qu'aux outils qui ont été utilisés et qui ont intégré progressivement notre (l'équipe) quotidien.

Puis nous développerons les trois étapes de réalisation qui ont marqué le projet avec un travail d'équipe de plus en plus consolidé et l'utilisation de technologies variées.

Enfin, nous verrons comment nous avons mis en application l'automatisation comptable.

Pour conclure, nous reviendrons sur l'ensemble des travaux d'automatisation réalisés au sein du service HMOA Multimédia Sport. Nous ferons alors un bilan du projet en établissant les forces et les faiblesses et proposerons des solutions pour la réussite d'un projet d'automatisation.

2 CONTEXTE DE LA MISSION

Altran Technologie est une société de service qui intervient, en outre, chez ses clients pour des prestations d'accompagnement de projets en ingénierie système. Je suis employé dans cette société et j'effectue depuis mars 2010 une mission au sein de l'équipe HMOA Multimédia Sport de la Française Des Jeux en tant que ingénieur validation technique.

Ma première mission concernait la qualification technique d'une partie de l'application ABP : le contrôle de la non-régression des processus de paiement, de la réconciliation⁵ des données, de la limitation des sessions invitées et authentifiées, de l'extraction des données vers l'ETL⁶ et vers la Base Analyse Joueur (BAJ).

La BAJ est la base d'enregistrement des historiques des prises de paris des joueurs.

Dans le cadre de ce mémoire et de la mission associée, il a été convenu avec mon responsable, M. Yannick Tison, qu'une partie de mon temps serait consacrée à la gestion du projet d'automatisation des tests techniques de l'équipe HMOA Multimédia Sport.

Ce mémoire présente notre (l'équipe) travail effectué sur l'automatisation des tests, au sein du service d'intégration, de qualification technique et de mise en production du projet *Modern* de la Française de Jeux sur le site de production de Vitrolles.

⁵ Comme pour tous les sites Internet de la Française Des Jeux, *ParionsWeb* communique avec la plate-forme centrale, contenant tous les comptes et sessions joueurs, pour que ces derniers puissent s'authentifier et faire des prises de jeux. La réconciliation est le processus asynchrone qui intervient pour mettre à niveau les informations entre les deux plateformes lorsqu'il y a eu un problème de communication.

⁶ Extract-Transform-Load: Il s'agit d'une technologie informatique intergicielle (comprendre middleware) permettant d'effectuer des synchronisations massives d'information d'une base de données vers une autre.

2.1 Organisation de l'entreprise

La *Française Des Jeux* a une organisation de type matricielle. C'est-à-dire que les Directions métiers et le directeur de projets sont co-responsables de la performance du projet.

Le directeur de projet désigne, planifie, organise, contrôle son équipe. Il a une autorité hiérarchique sur les intervenants. Il peut ainsi donner des priorités dans la réalisation des différentes tâches. Il est responsable de l'utilisation des ressources mais les intervenants dépendent hiérarchiquement de leurs Directions métier. Dans le cas de ParionsWeb, les membres de l'équipe HMOA Multimédia Sport sont la plupart du temps entièrement dédiés à *Modern*. Cependant, il peut arriver que ces derniers interviennent sur d'autres projets. Le schéma ci-dessous détaille de quels services sont issus les intervenants du projet.

2.2 Présentation de l'équipe HMOA Multimédia Sport

L'équipe d'intégration, au mois d'Avril 2011, est composée de huit personnes : M. Tison Yannick, M. Billard Fabien, M. Bellucci Lionel, M. Montels Joël, M. Al Oumami Amine, M. Huerta Stéphane, Mlle Poissonnet Barbara et moi-même, L'équipe est composée à 50% de prestataires. La durée des prestations est limitée à dix-huit mois et les prestations sont renouvelées tous les six mois. L'utilisation des ressources externes est par conséquent à prendre en compte dans l'organisation des projets.

Par ailleurs, comme dans toute entreprise, certaines personnes essentielles peuvent être amenées à partir et la transmission de leurs connaissances est un sujet crucial. Nous pouvons ainsi nous apercevoir qu'entre décembre 2010 et Avril 2011, deux membres de l'équipe ont été remplacés. M Martorano a été remplacé par Mlle Poissonnet et M. Daudon a été remplacé par M. Al Oumami.

2.2.1 Organigramme de l'équipe HMOA Multimédia Sport



Figure 2-2 Organisation HMOA Multimédia Sport en décembre 2010

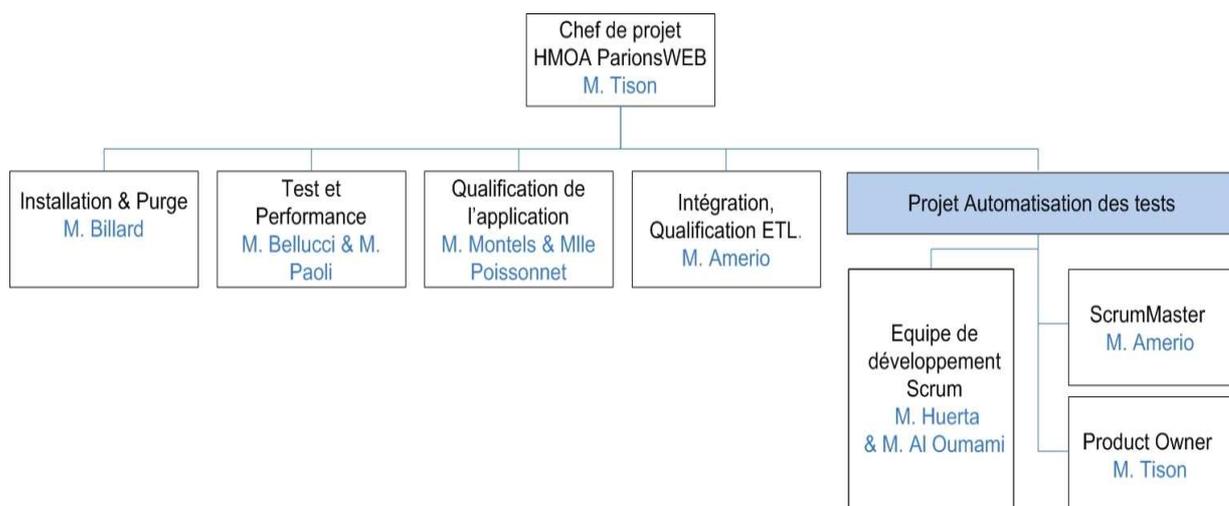


Figure 2-3 Organisation HMOA Multimédia Sport en Avril 2011

2.2.2 Activité de l'équipe

L'activité principale de l'équipe HMOA est l'intégration dans l'environnement de la Française Des Jeux, la qualification, la mise en production et la maintenance de l'application ABP dans le cadre du Projet *Modern*.

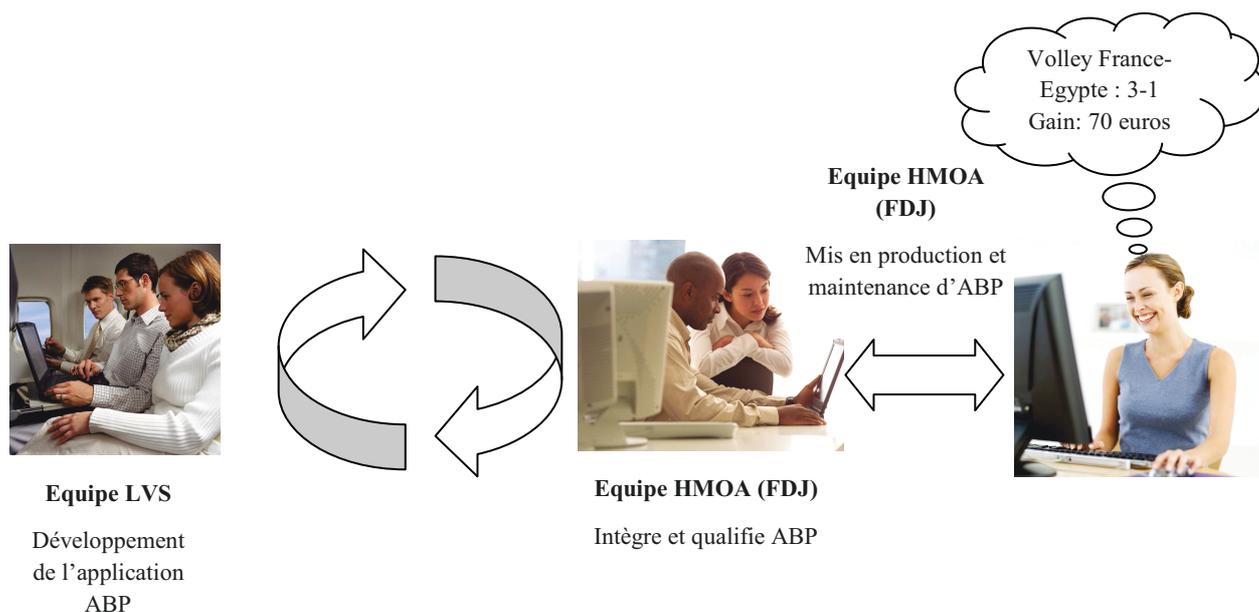


Figure 2-4 Schéma simplifié du rôle de l'équipe HMOA Multimédia Sport

2.3 Description du projet *Modern*

Le projet *Modern* est la création du site de paris sportifs en ligne *ParionsWeb*. La création et la gestion du site se fait grâce à l'application ABP fournie par LVS.

L'application ABP se compose :

- d'une application « ABP ToolsBar » développée en Java pour la gestion de l'offre⁷ sportive appelée *back office*,
- d'une base de données pour la gestion des données de l'offre et des prises de paris⁸,

⁷ La gestion de l'offre sportive concerne la création, la cotation, l'ouverture, la clôture, la promulgation, et le paiement des Markets. Dans la suite du document, nous parlerons d'offre pour désigner un ensemble de Markets.

⁸ Le joueur parie sur un support de pari sportif en choisissant la nature (pari simple, combiné ou système) et les paramètres de l'opération.

- de services pour la gestion et la diffusion d'offres sportives vers le front office⁹,
- de services pour la gestion des paris,
- de nombreux autres services pour la gestion de l'ensemble de l'application,
- d'un SGC¹⁰,
- d'un front office.

Le schéma ci-dessous, est mis en ligne sur le site de LVS et résume le fonctionnement de l'application ABP.

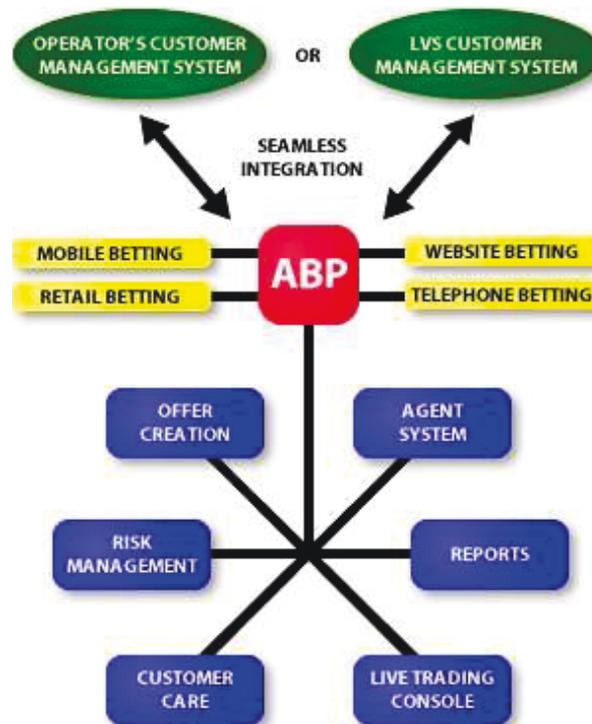


Figure 2-5 Schéma explicatif de l'application ABP

2.3.1 Présentation de « ParionsWeb » et du fonctionnement des paris sportifs

Le site ParionsWeb est une application en ligne de paris sportifs accessible par différents canaux. Ce site offre la possibilité de parier sur de nombreux sports tels que le football, le rugby, le handball, le basket, le volley, le tennis, les sports mécaniques, etc. Après avoir créés un compte et s'être authentifiés, les joueurs peuvent effectuer des paris sur les offres en ligne. Les offres de paris proposées peuvent prendre la forme de pronostics portant soit sur un événement à venir, soit sur un événement en cours. Dans ce dernier cas, ils sont appelés « pronostics live ». L'Arjel formalise cette

⁹ Le « front office » correspond à la partie hypertexte (web) de l'application.

¹⁰ Système de Gestion de Contenu : logiciel destiné à la conception et à la mise à jour dynamique de sites Web ou d'applications multimédia.

notion de pronostic en expliquant qu'un « pronostic est défini comme l'union de pronostics élémentaires » et qu'une « cote peut être associée au pronostic. » Elle définit également qu'un pronostic élémentaire est « caractérisé par un type de résultat et un choix de résultat. » (Arjel, Annexe au dossier des exigences techniques, 2012, p53).



Figure 2-6 Page d'accueil du site ParionsWeb (Juin 2011)

Afin de mieux appréhender l'analyse fonctionnelle et technique étudiée ultérieurement, il est important de décrire le fonctionnement des paris sportifs en général, et plus particulièrement sur le site de ParionsWeb.

a Trois types de prise de paris possibles

Les joueurs ont accès à plusieurs centaines de paris en simultané sur de nombreux sports. Trois types de « prises de paris » sont alors à leur disposition : les paris simples, combinés et multiples.

Pour les *paris simples*, une fiche de paris peut contenir jusqu'à huit paris simples.

Pour les *paris combinés*, le joueur gagne, si les pronostics sont tous exacts.

Pour les *paris multiples*, les joueurs ont le droit à l'erreur.

b La période de jeu

Quelque soit le type de prise de paris fait, les paris se font sur des pronostics liés à la période d'une manifestation sportive. Celle-ci représente l'intervalle de temps sur lequel peut porter un pari.

c Les types de résultats

Différents types de résultats existent¹¹ sur lesquelles les personnes peuvent jouer. Le plus courant est le 1 N 2

Ce type de résultat portant sur le résultat à la fin du temps réglementaire, consiste à déterminer le résultat d'une manifestation sportive. Pour chaque pari, le joueur choisit un pronostic élémentaire et un seul parmi les trois pronostics élémentaire possibles :

- le pronostic élémentaire «1 » correspond à la victoire de la première équipe (ou du premier sportif) citée dans le pari ;
- le pronostic élémentaire «N » correspond à un match nul ;
- le pronostic élémentaire «2 » correspond à la victoire de la deuxième équipe (ou du deuxième sportif) citée dans le « Résultat d'un match ».

Un type de résultat sportif est appelée « Market » dans l'application ABP¹².

d **Back office**

Le back office est l'élément central de l'application ABP. C'est une interface utilisateur développée en Java SWT¹³ qui offre la possibilité aux *traders* de créer et d'animer un pronostic en direct diffusé sur le front office. Un *trader* de paris sportifs est une personne physique qui établit des pronostics sur un résultat sportif. Il définit les cotes associées à ces pronostics. Il utilise le back office pour administrer des sports, des formules de paris et des cotes associées ainsi que le paiement des paris.

¹¹ Les types de résultats autorisés, et les phases de jeu sur lesquels ils portent, sont documentés et normalisés dans les ressources techniques du site de l'Arjel à l'adresse suivante : <http://www.arjel.fr/-Supports-de-paris-.html>

¹² Pour plus de facilité, nous utiliserons dans la suite de ce document le terme « Market » pour désigner un type de résultat sportif

¹³ Le Standard Widget Toolkit est une bibliothèque graphique libre pour Java.



Figure 2-7 Illustration de l'outil « event creator »¹⁴

¹⁴ L'outil « event creator » sert pour la création des « Markets » et est accessible depuis le back office d'ABP.

3 LE PROJET AUTOMATISATION

La première réunion sur l'automatisation a eu lieu le 17 septembre 2010. L'étude préliminaire a duré dix jours sur un intervalle de trois mois. La phase d'analyse détaillée a démarré mi-décembre et la phase de réalisation du projet a débuté en janvier 2011. Notre (l'équipe) rôle ainsi que le nombre de personnes rattachées à ce projet ont évolué. De nouveaux objectifs ont été définis au fur et à mesure de l'avancée du projet. Le projet d'automatisation s'est étiré sur la période de septembre 2010 à mars 2012. Cependant, ce mémoire ne concerne que *l'automatisation du test technique de non-régression comptable* et la période du démarrage du projet à fin juin 2011.

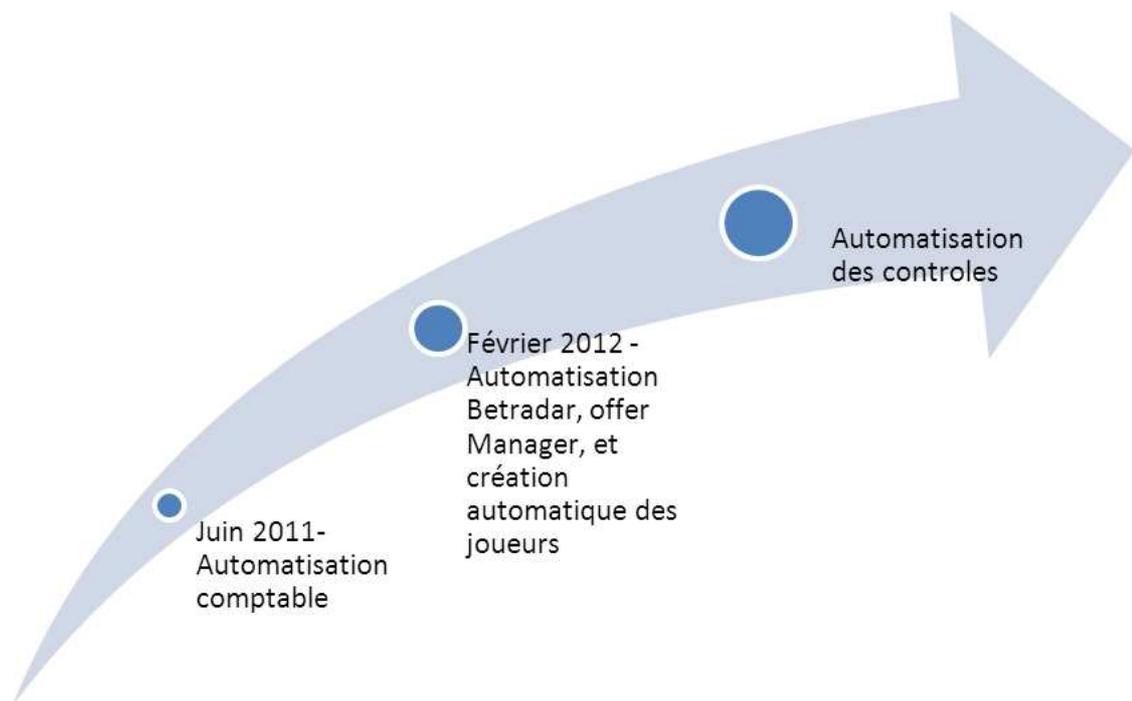


Figure 3-1 Schéma montrant l'avancée des développements à travers le temps

3.1 Problématique

« Avec une évolution permanente de l'application, une limitation des ressources et une grande fréquence des livraisons, les membres de l'équipe d'intégration HMOA Multimédia Sport n'ont pas le temps d'effectuer manuellement l'ensemble des tests de non-régression qu'ils ont à leur charge et des risques de dysfonctionnement peuvent apparaître en production. »

Pour éviter ces dysfonctionnements, la solution est de vérifier l'ensemble des nouvelles et anciennes fonctionnalités afin de garantir le bon fonctionnement de l'application.

Pour atteindre cet objectif, le donneur d'ordre, M. Tison, en l'occurrence, a décidé d'automatiser les tests techniques de non-régression que chaque membre de l'équipe effectue après chaque nouvelle intégration de l'application ABP. Certains de ces tests sont déjà en partie automatisés.

3.2 Etude préliminaire

A partir de l'expression du besoin de la Direction, nous avons décidé d'entreprendre une démarche projet. L'objectif est de dire si cela est réalisable et dans ce cas, de traduire un besoin en un produit utilisable. La première étape de la démarche projet est donc l'analyse préliminaire qui détermine si l'objectif est atteignable ou pas. Cette dernière est décrite ci-dessous.

3.2.1 Définition du besoin

A partir de l'objectif, nous avons effectué une première analyse qui a duré une dizaine de jours afin de mieux comprendre ses besoins et définir le cadre du projet. Le document fourni à l'issue de cette analyse était une note de cadrage.

3.2.2 Liste et analyse des contraintes

a Les contraintes fonctionnelles

- L'exécution automatique des tests sur plusieurs jours doit être possible.
- L'outil peut être lancé manuellement ou automatiquement. Cet outil doit être configurable et doit lancer un jeu de test de non-régression précis, indépendamment des autres jeux de tests.

b Les contraintes techniques

- Les contrôles d'intégrité doivent être faits, à la fois dans la base de données, de l'application ABP et dans la base de données Staging (Base de données alimenté par le traitement ETL (Extract-Transform-Load)).

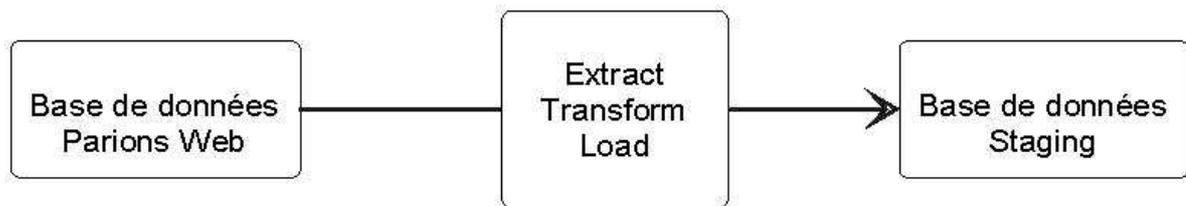


Figure 3-2 Schéma simplifié du transfert des données ABP

- Les outils existants, déjà utilisés par les équipes *Modern* et ceux développés pour automatiser une partie de certains tests de non-régression doivent être utilisés. Cela implique :
 - L'adaptation de l'outil développé à partir de Quick Test Pro¹⁵ (QTP) de Hewlett Packard qui automatise la création de « Markets »
 - L'adaptation de l'outil d'injection des prises de paris développé en langage PERL.
- La solution technique choisie doit tenir compte d'environnements techniques variés :
 - Le back office d'ABP utilisé pour la création de « Markets » est une interface Java SWT.
 - Le front office, utilisé pour l'authentification et les prises de paris, est développé pour le web et se base sur le protocole http, sur le html, le javascript ou encore le Flash.
 - L'exécution de script batch sur les environnements d'intégration peut être nécessaire dans certains cas.
 - L'interrogation de la BDD oracle des environnements d'intégration doit être possible.
- L'outil doit agir sur les environnements d'intégration.

c Les contraintes de planning

- Le projet automatisation peut se terminer à la fin du mois de février 2011, date à laquelle je termine potentiellement ma mission à la Française Des Jeux.
- Au-delà du mois de février, aucune contrainte de date n'est définie.

d Les contraintes de performances

Le niveau de performance de l'application doit respecter les exigences suivantes :

- Le temps d'exécution automatisé d'un test de non-régression doit être moins long que l'exécution manuelle de ce test.
- Pour chaque nouvelle livraison, la mise à jour d'un test automatisé doit être moins longue que l'exécution manuelle de ce test.

¹⁵ HP QuickTest Professional assure l'automatisation des tests fonctionnels et de non-régression. Grâce à une technologie de capture spécialisée, elle permet aux testeurs de concevoir des jeux d'essai en capturant directement les flux à partir des écrans de l'application.

- L'outil doit améliorer ou aider à l'amélioration de la couverture des tests manuels.

3.2.3 Risques

Le risque le plus important est l'arrêt du projet. En effet, ce projet n'est pas autonome et dépend directement du budget de qualification technique alloué au projet *Modern* en cours. Le projet peut donc être arrêté par décision du chef de projet: M. Blondeau Jérémie.

L'autre risque est l'échec du projet d'automatisation. Il est lié à l'utilisation des outils QTP et PERL que nous ne maîtrisons pas à ce stade et dont nous ne connaissons pas les limites techniques. Même si une analyse préalable de ces outils est faite, ce n'est réellement qu'au fur et à mesure de l'utilisation et de l'adaptation de ces outils que nous pouvons découvrir ses limites.

3.2.4 Analyse des outils existants

a Quick Test Professional (QTP)

QTP est l'outil utilisé lors de la qualification technique par l'équipe de qualification pour créer, clôturer et payer de manière automatisée l'offre sportive *via* le back office de l'application ABP. QTP est un outil qui agit sur l'interface graphique en simulant les actions des utilisateurs. Le développeur définit ainsi un scénario d'actions (clique, saisie de texte, etc.). Une fois lancée, l'application reproduit ses actions de manière automatisée. Le langage de développement est Visual Basic Script (VBScript)¹⁶.

Avantages

- La licence est achetée par la Française Des Jeux et l'outil est utilisable.
- QTP possède des bibliothèques intégrées capables de lire les fichiers Excel. Format utilisé pour les fichiers d'entrée de l'application de création de l'offre.
- Le logiciel appartient à HP Software, département logiciel du groupe Hewlett-Packard, une des plus grosses compagnies mondiales de logiciels.
- HP assure encore la maintenance.
- Une mise à jour importante par année¹⁷ est faite.
- La communauté est importante et plusieurs didacticiels sur le sujet existent.
- QTP supporte les technologies sur lesquelles nous travaillons ou nous sommes susceptibles de travailler :
 - Web pour accéder au front office.
 - « Java » (Core and Advanced) et add-ins¹⁸ SWT pour accéder à l'interface du back office de l'application

¹⁶ **VBScript** (diminutif de **Microsoft Visual Basic Scripting Edition**) est un sous-ensemble de Visual Basic utilisé en tant que langage de script d'usage général.

¹⁷ Voir : http://en.wikipedia.org/wiki/HP_QuickTest_Professional

- Oracle (base de données)
- « Terminal Emulator » pour se connecter de manière sécurisée sur une machine d'intégration au travers d'un terminal et exécuter des scripts batch sur la machine distante.
- « Flex » pour accéder aux objets Flash
- Web Services

Inconvénients

- L'outil se base sur l'interface graphique pour effectuer des actions.
- HP Quick Test Professional fonctionne uniquement sous « Windows ».
- Il ne fonctionne pas avec Internet Explorer d'une version antérieure à 8.

b PERL (Practical Extraction and Report Language)

PERL est un langage de programmation et plus particulièrement de script¹⁹.

Avantages

- C'est un langage interprété, polyvalent, et particulièrement adapté au traitement et à la manipulation de fichiers textes.
- Ce langage est très complet et multiplateforme.
- Ce langage a un statut de logiciel libre, distribué sous double licence : Artistic Licence et GPL.
- Pour les besoins de l'injecteur, nous utilisons uniquement les bibliothèques sous licence GPL.
- Il est utilisé dans de nombreuses entreprises et de nombreuses bibliothèques de base y sont associées. Des didacticiels et exemples de codes existent sur Internet.

Inconvénient

- La version de langage PERL installée actuellement ne permet pas de créer des objets²⁰. Or, les développements de l'injecteur ont été effectués avec cette version. La mise à jour du langage nécessitera la mise à jour des scripts associés.

3.2.5 Proposition d'une architecture de type modulaire

Nous avons effectué un travail de réflexion sur l'architecture modulaire qui s'est ensuite accompagné d'une analyse des tests réalisés par l'équipe HMOA. Plusieurs entretiens ont été nécessaires pour comprendre comment ces tests fonctionnaient et comment nous pourrions les intégrer dans cette architecture. Cette compréhension du fonctionnement des tests m'a aidé à trouver quelles fonctions et sous-fonctions étaient communes entre chaque test. Sans ce travail de compréhension global du

¹⁸ *add-in* ou *add-on* est un paquet qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités.

¹⁹ Voir : [http://fr.wikipedia.org/wiki/Perl_\(langage\)](http://fr.wikipedia.org/wiki/Perl_(langage))

²⁰ Les objets sont utilisés dans la programmation orientée objet.

fonctionnement des tests, nous aurions difficilement pu appliquer le modèle architectural souhaité au projet.

a Définition de l'architecture modulaire

Les éléments constitutifs du système sont indépendants. La modification de l'un ou le retrait de l'autre n'a pas d'influence sur les autres éléments.

b Classification Arborescente

Afin d'arriver à ce type d'architecture, il est nécessaire de décomposer les fonctions en sous-fonctions qui peuvent par la suite être réutilisées. Plus le découpage est fin, plus la réutilisation est facile et plus l'indépendance entre les modules est possible.

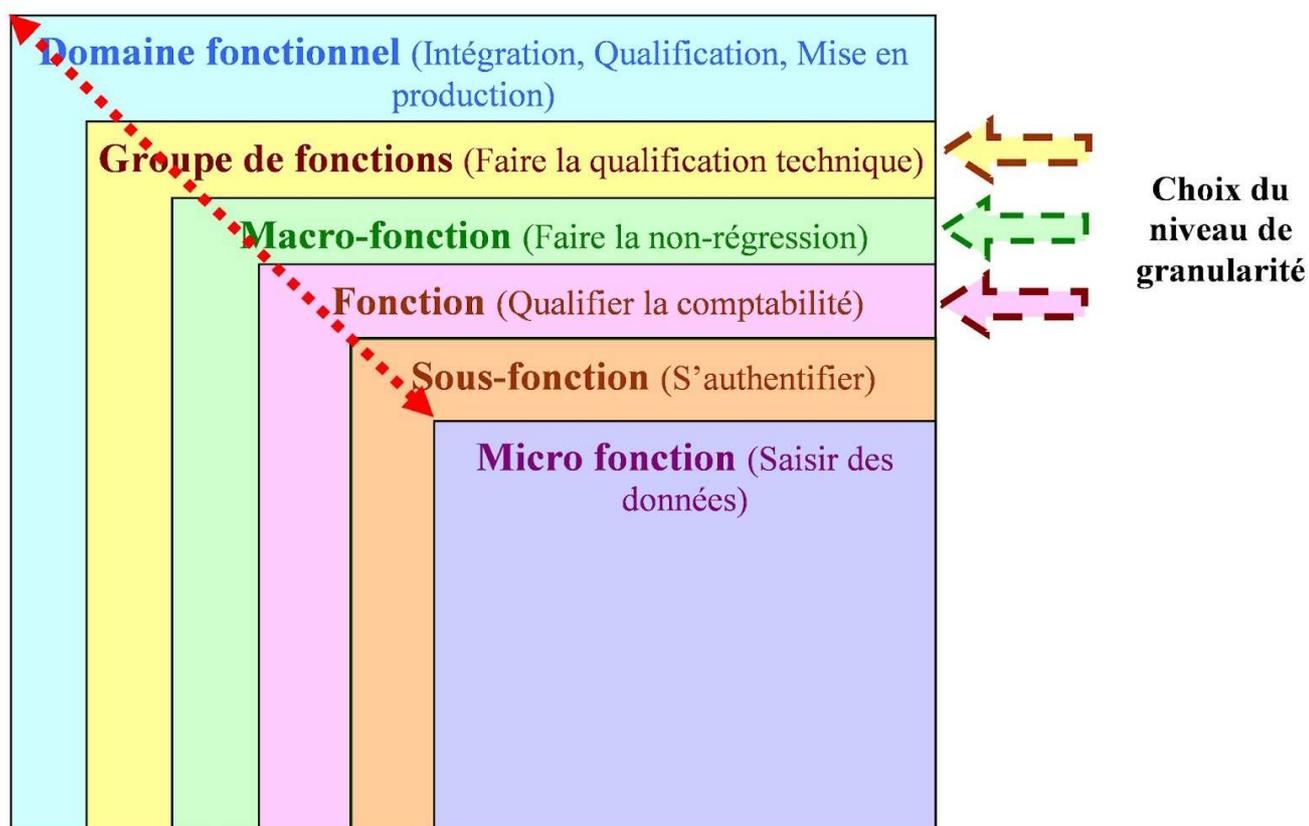


Figure 3-3 Schéma des différents niveaux de granularité utilisés pour l'architecture

Si nous adaptons cette architecture pour les besoins de l'automatisation, nous obtenons le schéma ci-dessous qui liste l'ensemble des tests faits actuellement par l'équipe. L'ensemble des tests de non-régression peut être assimilé à une macro-fonction dans l'architecture modulaire. Les fonctions représentent chacun des tests de non-régression.

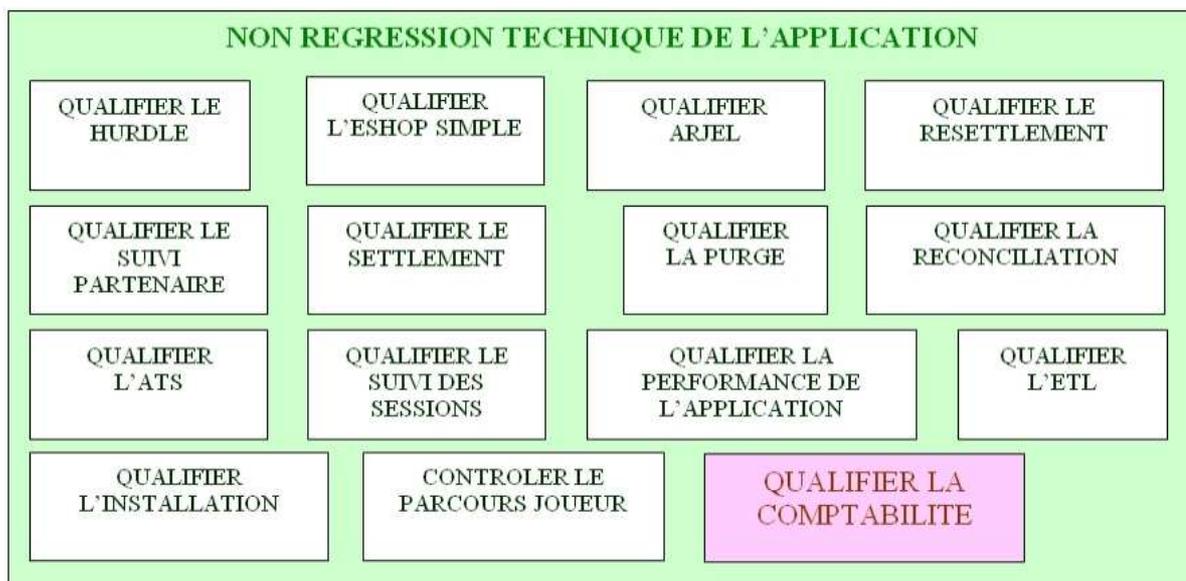


Figure 3-4 Modèle fonctionnel de la qualification technique de non-régression d'ABP

Pour exemple, lorsque nous décomposons la fonction « Qualifier la comptabilité » en sous-fonctions, nous obtenons le schéma ci-dessous, comprenant l'ensemble des sous-fonctions nécessaires à la comptabilité.

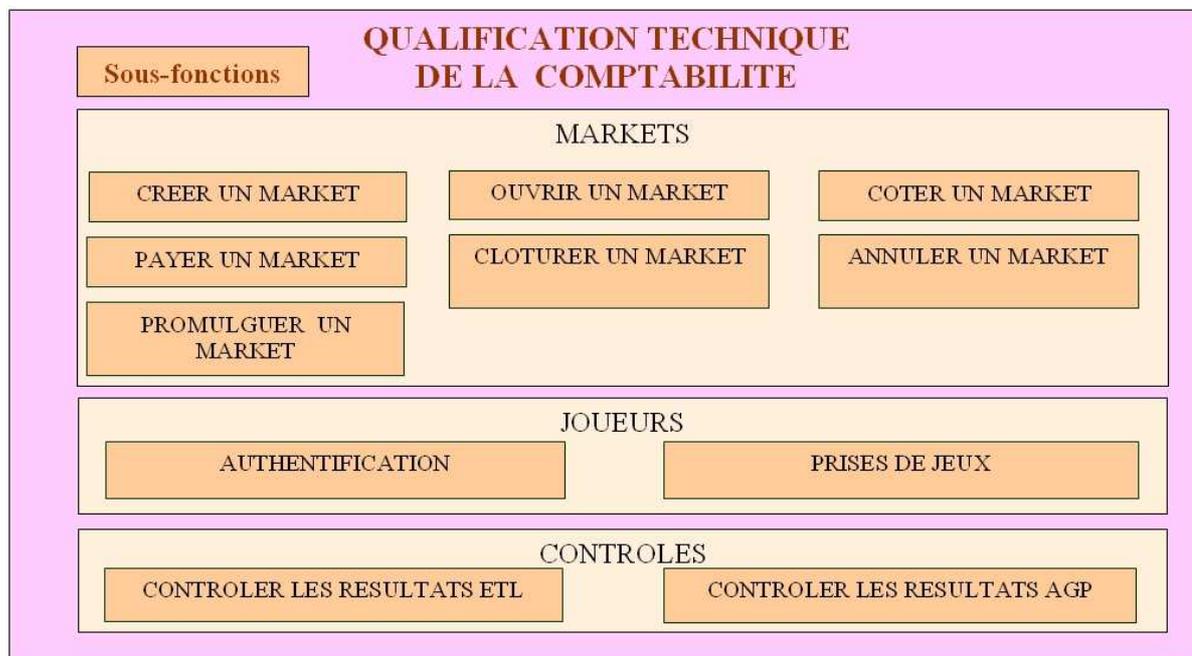


Figure 3-5 Sous-fonctions utilisées pour qualifier la comptabilité

Le schéma ci-dessous montre une partie des sous-fonctions utilisées par le processus de qualification comptable :

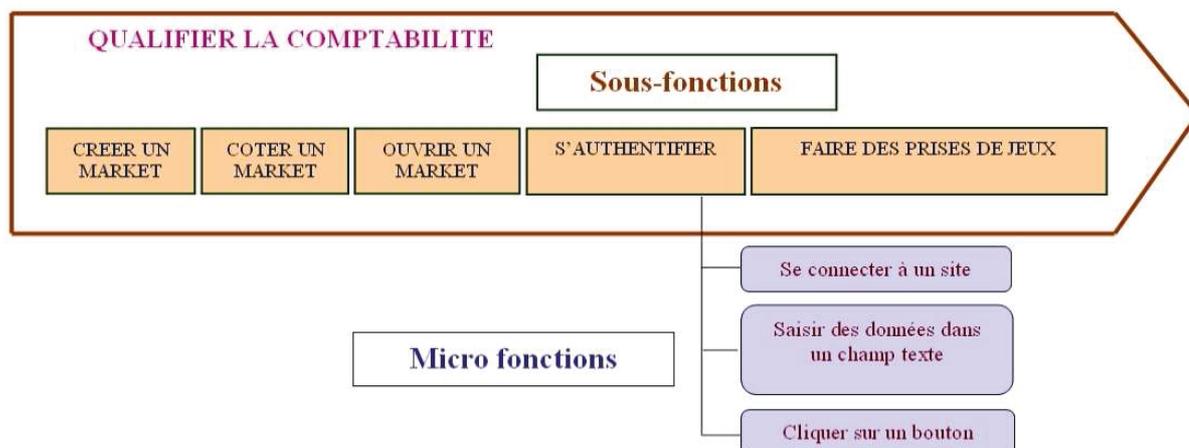


Figure 3-6 Extrait du processus de qualification de la Comptabilité

c **Tableau de correspondance sous-fonctions/fonctions**

Le tableau ci-dessous est nécessaire pour :

- savoir quelles sont les sous-fonctions à développer,
- et savoir quelles sont celles qui seront réutilisables.

Fonctions Sous-fonctions	Qualifier la Comptabilité	Qualifier la bascule	Qualifier l'Arjel ²¹	Qualifier le paiement	Qualifier la réconciliation	Qualifier Betradar
S'authentifier	X		X	X	X	
Faire une prise de paris	X		X	X	X	
Créer un « Market »	X	X	X	X	X	
Coter un « Market »	X	X	X	X	X	
Ouvrir un « Market »	X	X	X	X	X	

²¹ L'Autorité de régulation des jeux en ligne (Arjel) est une autorité administrative indépendante, spécialement prévue pour réguler les jeux d'argent sur Internet en France.

Clôturer un « Market »	X		X	X	X	
Promulguer un « Market »	X		X	X	X	
Annuler un « Market »	X			X		
Contrôler en base de données	X		X	X	X	
Lire un fichier CSV	X			X	X	
Générer un fichier CSV	X	X	X	X	X	
Exécuter un script					X	
Lire un fichier XML						X

A partir de ce tableau, nous pouvons voir que des sous-fonctions sont réutilisables pour différentes fonctions. Ce tableau donne une liste non exhaustive de l'association sous-fonctions et fonctions.

d Fonctionnement d'une Qualification

Qualifier, revient à effectuer des actions et à contrôler que le résultat obtenu suite à ces actions est bien celui attendu. La qualification se compose de plusieurs phases :

- La première phase est appelée *prérequis*. Celle-ci s'effectue avant de démarrer la qualification d'une partie de l'application. Elle est nécessaire pour préparer les environnements aux conditions requises pour le test. C'est dans cette phase par exemple que l'on crée les joueurs nécessaires aux prises de paris et que l'on vérifie les connexions à la base de données (BDD).
- Lors de la deuxième phase, la personne en charge de la qualification effectue une suite d'opérations logique en utilisant les outils de l'application. C'est le résultat obtenu par cette suite d'opérations logique que la personne contrôle. Par exemple, la personne crée un « Market » et effectue une prise de paris dessus.
- La troisième phase sert à contrôler l'intégrité des données, généralement en base de données. Cependant il arrive que certains contrôles doivent se faire dans des fichiers
- La dernière phase sert à l'écriture du rapport. Celui-ci indique l'état du test : Si ce dernier est en erreur ou non et pourquoi.

Lors de la deuxième phase, entre chaque opération, le testeur effectue des contrôles fonctionnels. Par exemple, le testeur ne commencera pas à faire des prises de paris si l'opération précédente, de création des « Markets », a échoué.

Pour chaque phase, nous pouvons y associer des sous-fonctions et les sous-fonctions listées précédemment pourront servir à réaliser une action dans une ou plusieurs phases en particulier.

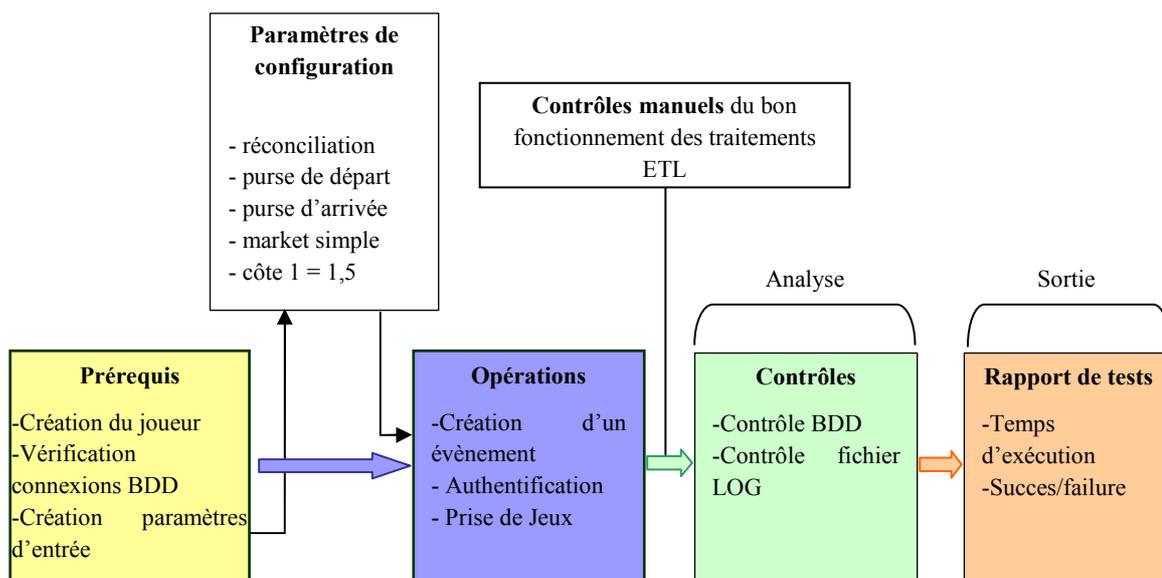


Figure 3-7 Schéma du processus de test comptable et ses principales actions

En violet, les sous-fonctions représentent des actions à réaliser préalablement au contrôle. Selon le type de contrôle à effectuer, elles peuvent être requises ou non. Un fichier de configuration fournit tous les paramètres nécessaires à ce type de sous-fonctions, ainsi qu'aux sous-fonctions d'analyses.

En vert, les sous-fonctions représentent les actions effectuées pendant un contrôle.

En orange, le résultat de l'analyse globale doit être récupéré, enregistré et diffusé aux personnes concernées par ces contrôles.

Ces trois actions correspondent au niveau de maturité du test automatisé. En effet, dans un premier temps, avant de pouvoir effectuer des contrôles, toutes les actions liées aux prérequis et aux opérations doivent être terminées : Les données liées à des prises de paris ne peuvent être contrôlées si le module n'a pas pu ouvrir une session utilisateur et faire ces prises de paris.

3.2.6 Conclusion de l'étude

A partir de l'analyse des besoins, nous avons réalisés cette analyse préalable. Celle-ci a duré une dizaine de jours et a montré que l'analyse approfondie de tous les tests automatisés empêcherait, dans le temps imparti, la réalisation d'une partie opérante de l'outil d'automatisation.

Cette analyse a également définie les bases de l'architecture modulaire applicable à l'automatisation des tests de non-régression grâce à une étude de l'urbanisation du système d'information de ces derniers. Cette architecture modulaire rend possible le découpage du projet d'automatisation de l'ensemble des tests en plusieurs sous projets dont chacun a pour objectif l'automatisation d'un des tests de non-régression.

a Définition d'un nouvel objectif

Nous avons pu mettre en avant, grâce à l'étude préliminaire, que la charge de travail pour analyser l'ensemble des tests techniques avant de pouvoir commencer les développements de l'automatisation, apparaît alors trop importante. En choisissant de procéder de la sorte, aucun test n'aurait raisonnablement pu être automatisé dans les délais impartis (pour mémoire, avant fin février). Lors d'une réunion, la décision a donc été prise de ne pas engager une analyse détaillée pour chaque test, mais davantage de se concentrer sur un objectif atteignable pour la fin février. Il s'agit alors d'assurer le développement du ou des modules les plus stratégiques, et de fournir rapidement un retour sur les outils existants et imposés pour l'automatisation.

L'objectif principal en passe dès lors par **l'automatisation des tests techniques de non-régression comptable**.

b Anticipation sur les objectifs futurs

Les futurs objectifs ne sont, en conséquence, pas définis, ni même priorisés. Cependant, il est clair pour la Direction que ceux-ci ne le seront qu'à partir du moment où le premier objectif sera atteint et dans la mesure où les conditions budgétaires seront favorables.

3.3 Initialisation du projet

3.3.1 Analyse de la situation

a Besoin d'un retour rapide sur les technologies utilisées

Le nouvel objectif exprimé est clairement défini en termes d'automatisation comptable. Il doit non seulement respecter les délais fixés pour la fin février 2011, mais aussi rendre efficient l'automatisation des tests de régression comptable.

Les contraintes imposées dès le départ par la Direction portent sur l'utilisation des outils existants et sur leur adaptation dans le cadre de l'automatisation comptable.

Cependant, nous avons mis en évidence lors de l'analyse des risques que ne disposons que très peu de connaissances sur ces outils et nous ne savons pas si ces derniers répondront à nos attentes. En effet,

la personne spécialisée dans l'usage de ces outils a quitté la société seulement deux semaines après le démarrage de la phase de réalisation du projet. D'autre part, les besoins et les contraintes de celle-ci, en termes d'automatisation, n'étant pas les mêmes que les nôtres, le transfert de connaissances a été incomplet.

Le choix imposé d'utiliser la technologie QTP et, surtout, de reprendre un outil existant sans avoir le moindre recul concernant les limitations techniques ne garantit alors en aucun cas l'atteinte de l'objectif. Ce risque, connu dès le démarrage du projet, est toutefois jugé acceptable par la Direction.

Pour ces raisons, nous avons décidé de faire un retour sur ces technologies. Dès le démarrage du projet, nous avons distingué deux étapes pour l'automatisation comptable. Ces deux étapes représentent des livrables et il a été possible de définir deux itérations pour chacune d'elles.

b Penser aux objectifs futurs

Le fait de se fixer un objectif précis à la suite de l'analyse préliminaire, n'exclue pas la volonté de définir, par la suite, de nouveaux objectifs. La demande d'ajout de nouvelles fonctionnalités doit être prise en compte, celle-ci pouvant toujours, par la suite, survenir. Il apparaît donc important d'assurer la mise en place d'une gestion de projet à même d'intégrer, de façon continue, ces nouvelles fonctionnalités dont nous n'avions alors aucune connaissance.

c Au-delà de l'objectif d'automatisation comptable

La mise à disposition de personnes (M. Al Oumami et moi-même) au commencement du projet par la Direction ne dépasse pas la fin février et la fin de la première étape de l'automatisation comptable. Comme précisé plus haut, le risque d'arrêt du projet n'est alors pas à exclure et ne permet pas d'avoir de certitudes concernant la durée de celui-ci, notamment quant à sa poursuite au-delà du mois de février.

La mise en place d'une gestion de projet avec des itérations successives constitue de ce fait une réponse bien ajustée à ce type de situation. En effet, à tout moment, dès la fin d'une itération courte, l'arrêt du projet est possible ; mais avec l'assurance d'avoir finalisé et livré un module. A cela, dans l'hypothèse où le projet se poursuit, une gestion itérative intègre alors avec plus de facilité les futures demandes.

d Proximité avec les utilisateurs finaux

Mon interlocuteur principal est M. Tison, celui-ci ayant souhaité être impliqué et suivre l'avancement de l'automatisation comptable. Toutefois, des échanges s'opèrent également, de façon ponctuelle, avec des membres de l'équipe HMOA Multimédia Sport. Il s'agit souvent de personnes possédant des connaissances sur tel ou tel test manuel devant être analysé et éventuellement automatisé par la suite. Étant dans le même « open-space » que l'équipe, ces échanges se trouvent facilités du fait même de cette proximité.

3.3.2 Choix de la méthode

a Définition d'une méthode agile

Une méthode agile est une méthodologie de gestion de projet.

« Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du besoin du client et non les termes d'un contrat de développement. »²².

Une méthode agile est avant tout itérative. Elle procède par un affinement du besoin mis en œuvre dans des fonctionnalités en cours de réalisation ou déjà réalisées. Cet affinement, se réalise en matière de génie logiciel sous deux aspects :

- fonctionnellement, par adaptation systématique du produit aux changements du besoin détecté par l'utilisateur lors de la conception-réalisation du produit ;
- techniquement, par remaniement régulier du code déjà produit.

Elle répond aux 4 valeurs et aux 12 principes du manifeste *Agile* lancé en 2001 par 17 personnalités de l'industrie logicielle (Beck, Cockburn, Fowler, Jeffries, Marick, Schwaber, Sutherland...), en réponse à un malaise et aux insuffisances, ainsi qu'aux difficultés liées aux méthodes de développement traditionnelles :

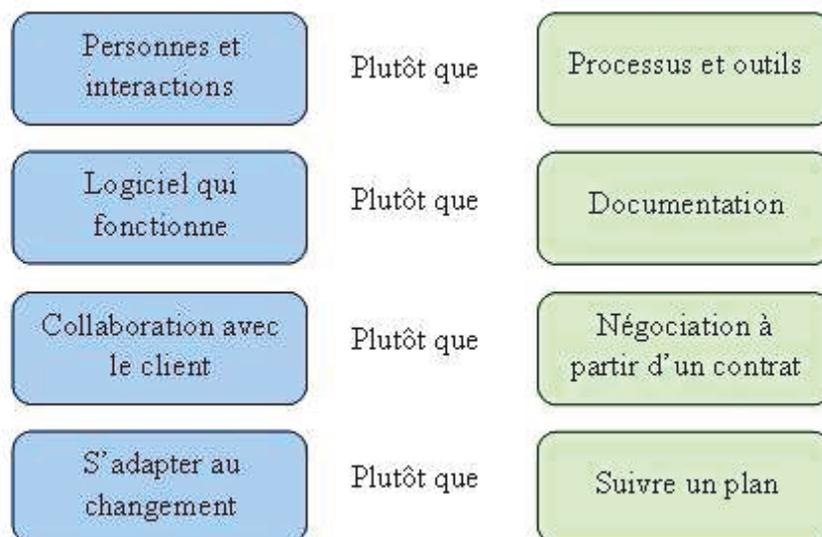


Figure 3 6 Les 4 valeurs du manifeste Agile

²² Source : Wikipedia.

b Un Besoin d'agilité pour ce projet

Au vu de l'analyse de la situation précédemment décrite, utiliser une méthodologie de type Agile apparaît plus opportune pour mettre en place l'automatisation comptable.

En effet, cette dernière semble plus adaptée pour ce type de situation. Elle répond au besoin d'un retour rapide sur la faisabilité de l'automatisation en utilisant les outils existants ; et elle s'adapte au contexte du projet, à savoir : pas de visibilité sur la suite du projet au-delà du mois de février, tout en maintenant la possibilité d'intégrer de nouvelles fonctionnalités au fur et à mesure de l'avancée du projet.

c Livraisons rapides

Une méthodologie de type Agile fournit rapidement une première partie fonctionnelle d'un produit. Dans le cadre du projet d'automatisation, et notamment dans la première étape du projet, cette méthode offre la possibilité de savoir rapidement si les outils existants répondent aux besoins. Cela évite de s'engager sur une durée de réalisation trop longue avec des outils, qui au final ne pourraient pas être utilisables. Le tout associé à une architecture modulaire, nous pouvons rapidement décider de changer un module en cours de projet.

d Réagir rapidement aux circonstances changeantes

L'autre avantage de la méthodologie *Agile* est de ne pas avoir de contraintes d'engagement sur le long terme. Dans le cadre de notre projet, cette méthode se révèle particulièrement adéquate dans l'hypothèse où notre projet prendrait fin en février.

Elle nous offre ainsi la possibilité d'effectuer une itération pour chaque nouvelle fonctionnalité souhaitée par la Direction. Nous pouvons ainsi stopper les développements de l'automatisation, mettre fin au projet à la fin d'une itération et libérer les ressources dédiées.

e Mise en place de l'Agilité avant tout

Il reste que notre choix ne s'est cependant pas porté initialement sur l'utilisation d'une méthodologie dite « Agile » en particulier. Deux raisons à cela :

- D'une part, l'équipe d'automatisation est, au départ, composée de seulement deux personnes : un chef de projet (nous-même) et un développeur (M. Al Oumami). La communication était directe. Nous (l'équipe et la Direction) communiquions ensemble sur l'architecture et sur l'avancée des développements très facilement puisque nous étions à proximité. La gestion du projet était simple.
- D'autre part, le client n'avait pas pour habitude d'utiliser les principes d'agilité. La plupart des méthodologies dites « agiles » demande une implication plus importante de la part du client que pour des méthodes traditionnelles de gestion de projet. Il apparaît plus évident, que pour faire adhérer le ou les clients à cette méthodologie de gestion de projet, il soit préférable que sa mise en place soit progressive.

Néanmoins, nous verrons dans la suite de ce document de quelle manière nous avons, dès le départ, respectés les valeurs de l'agilité.

f **Choix de Scrum**

Le projet d'automatisation ne s'est pas arrêté à la fin de la première étape en février 2011. De nombreuses itérations ont été faites par la suite et plusieurs d'entre elles sont décrites en détail dans la suite de ce document. C'est lors de ces autres itérations, lorsque l'équipe s'est renforcée avec l'arrivée sur le projet d'une troisième personne, que j'ai décidé d'adopter et d'utiliser au fur et à mesure les outils fournis par Scrum²³ : **graphique d'avancement** (appelé « Burdown Chart »), **carnet du produit** (appelé « Backlog de produit ») et **la Mêlée quotidienne** (appelée « Daily Scrum »). Partant ainsi de principes agiles, élémentaires et nécessaires au projet, nous nous sommes au fur et à mesure organisés autour de Scrum.

Grâce à ses outils, Scrum propose un cadre pour la gestion du projet.

Scrum offre également, la possibilité de trouver une place plus adaptée à la situation de l'équipe. Ce fut le cas lors de la deuxième étape du projet (automatisation de la création et de la clôture des « Markets » via l'API REST). M. Huerta Stéphane ayant rejoint l'équipe en cours ; ce dernier, a alors une grande expérience et de très bonnes compétences en gestion de projet. Mon rôle de chef de projet, lors de la première étape de l'automatisation, risque alors d'être difficilement accepté par M. Huerta, même en étant soutenu par la Direction. Le rôle de ScrumMaster est un rôle de facilitateur, garant de l'application des processus. Dans une gestion de projet Scrum, il n'y a pas de chef de projet directif et tous les membres de l'équipe contribuent à l'avancement du projet. L'équipe est alors autonome. Avec ce nouveau rôle, il est plus facile pour moi de m'appuyer sur les compétences de l'équipe, de responsabiliser les membres et de garantir le besoin de reconnaissance de chacun.

La venue de M. Huerta dans l'équipe, de par ses compétences en management de projet agile a également contribué au choix de cette méthode.

L'intérêt y est triple :

- nous gagnions une personne experte dans les méthodes Agiles, connaissant très bien Scrum et à même d'aider à la mise en place de cette méthode ;
- nous gagnions une personne plus encline à utiliser une méthode qu'elle connaît bien ;
- nous répondions à son besoin de reconnaissance, puisque son rôle est de former et d'accompagner toutes les personnes du projet dans l'utilisation de la méthodologie Scrum.

Nous verrons par la suite de quelle manière nous avons inclus progressivement les principes de Scrum dans la gestion de projet ainsi que ses outils.

g **Avant de commencer la phase de réalisation**

Avant de démarrer la phase de réalisation d'un projet Agile, une première analyse globale des besoins doit être conduite pour que l'équipe s'imprègne du contexte métier. Dans notre projet, en plus de l'analyse des besoins, une analyse détaillée est réalisée afin de bien comprendre et éclairer le fonctionnement du test comptable. Cette analyse précise également quels modules seront nécessaires

²³ Scrum est un framework Agile. C'est-à-dire qu'il fournit des outils utiles pour une gestion de projet de type Agile. Scrum est détaillé dans le paragraphe 3.5.1

et comment techniquement le test sera mis en place. La création de modules indépendants, ne permet pas à elle seule la réalisation du test comptable. Les modules doivent pouvoir s'enchaîner les uns après les autres. L'analyse détaillée, que nous allons à présent présenter, sert à préciser comment vont s'enchaîner ses modules et quel est l'outil qui va être utilisé pour ses enchaînements.

3.4 Analyse détaillée

3.4.1 Analyse du fonctionnement du test comptable

a *Comment fonctionne le test comptable ?*

Le test de régression comptable manuel s'effectue sur 4 jours. Il a pour but de contrôler les données comptables vis-à-vis d'un jeu de données précis en entrée. On retrouve dans le schéma ci-dessous le processus de test comptable.

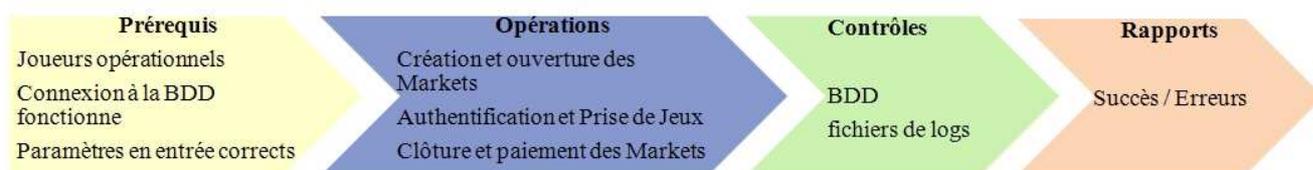


Figure 3-8 Schéma du processus comptable

L'utilisateur a sauvegardé un jeu de données d'entrée dans un fichier Excel et l'utilise comme référence pour rejouer le même jeu de données à chaque test de régression. L'avantage de rejouer le même jeu de données d'entrée est que le jeu de données en sortie ne change pas.

b *Ordonnancement du test sur plusieurs jours*



Figure 3-9 Schéma représentatif de l'enchaînement des actions du test comptable sur 4 jours

Jour 1

- L'utilisateur crée l'offre sportive depuis le back office d'ABP.
- L'utilisateur se connecte à différents comptes joueurs et effectue plusieurs prises de paris sur des pronostics dans l'application ABP.
- L'utilisateur clôture, promulgue, annule et/ou paie des pronostics.

Jour 2

- L'extraction des données ABP est effectuée de manière automatique dans la nuit grâce aux deux principaux traitements : ETLExpport et ETL.
- L'utilisateur contrôle et compare les résultats comptables de la première journée
- L'utilisateur se connecte à différents comptes et effectue plusieurs prises de paris sur des pronostics bien définis.
- L'utilisateur clôture, promulgue, annule et/ou paie des pronostics.

Jour 3

- L'extraction des données ABP est effectuée dans la nuit grâce aux deux principaux traitements : ETLExpport et ETL.
- L'utilisateur contrôle et compare les résultats comptables de la seconde journée
- L'utilisateur se connecte à différents comptes et effectue plusieurs prises de paris sur des pronostics bien définis.
- L'utilisateur clôture, promulgue, annule et/ou paie des pronostics.

Jour 4

- L'extraction des données ABP est effectuée dans la nuit grâce aux deux principaux traitements : ETLExpport et ETL.
- L'utilisateur contrôle et compare les résultats comptables de la troisième journée.

c Les traitements ETL

Le schéma ci-dessous synthétise les informations en entrée et en sortie du traitement ETL. L'avantage de ce tableau est de donner une vue d'ensemble.

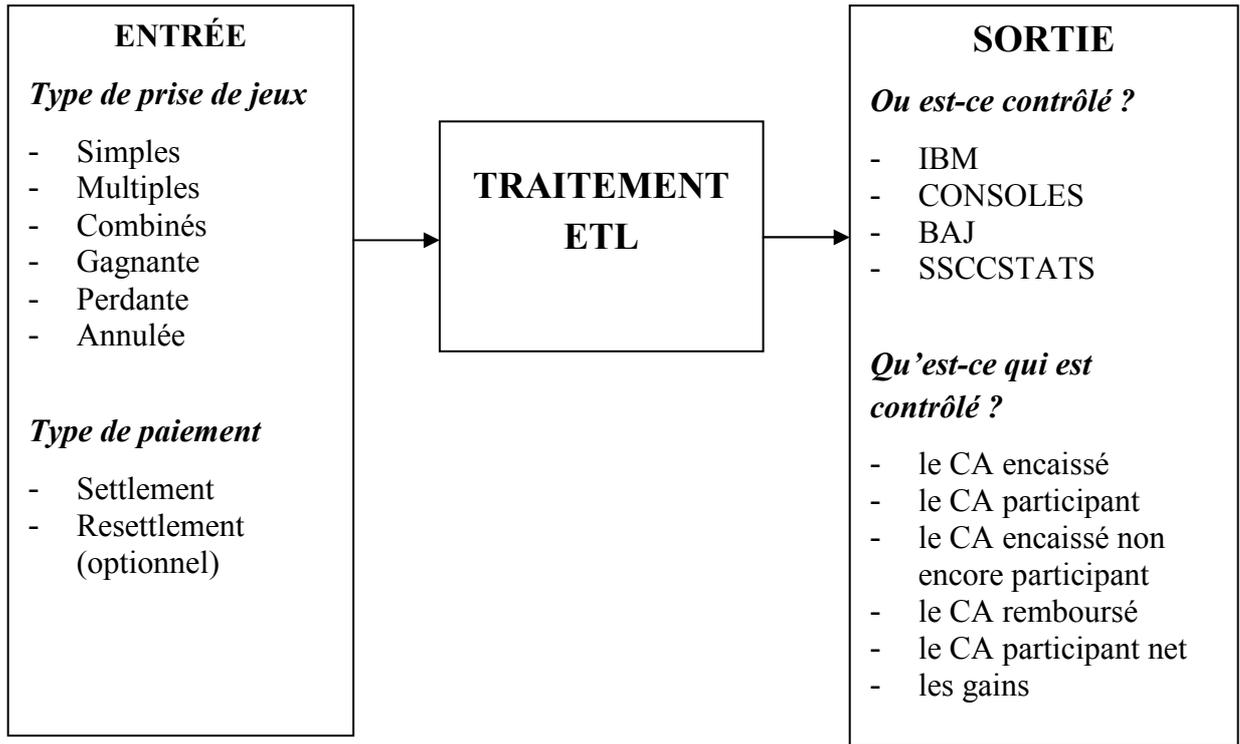


Figure 3-10 Schéma d'analyse des données comptables

Si l'on reprend le processus de test comptable, les traitements ETL passent avant que l'on puisse effectuer les contrôles comptables.



Figure 3-11 Schéma détaillé du processus comptable entre les Opérations et les contrôles

Le schéma ci-dessus reprend le modèle d'architecture modulaire décrit plus haut dans ce document. Nous retrouvons l'enchaînement des actions de type opérations et contrôles.

d Où sont les données comptables?

Le contrôle des données comptables s'effectue à quatre endroits différents, appelés *supports comptables* : Consoles, IBM, BAJ et Statistiques.

Les consoles sont des fichiers générés tous les jours et accessibles depuis une URL interne. Les données de statistiques, de la BAJ et d'IBM sont remontées dans des tableaux de bord grâce à un logiciel d'informatique décisionnelle nommé « Business Object ».

e Les contrôles

Deux types de contrôles doivent être pris compte lors de l'automatisation :

- Le premier concerne les contrôles effectués avant et après chaque module et à l'intérieur même des modules. Ils servent à vérifier la conformité des données fournies en retour d'un programme, vis-à-vis de règles fonctionnelles. Ces contrôles seront appelés *contrôles clients* ou *contrôles fonctionnels*.
- Le second concerne les *contrôles d'intégrité des données* et nécessite la plupart du temps le développement de modules spécifiques. C'est le cas par exemple des modules de contrôle des données comptables en base de données.

Les deux schémas ci-dessous détaillent l'emplacement des différents types de contrôles :

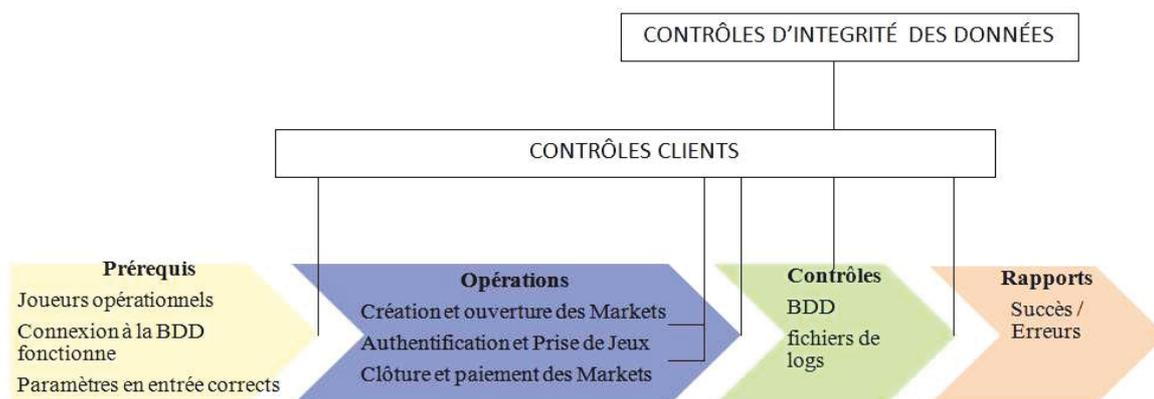


Figure 3-12 Schéma du processus comptable et les types de contrôles associés

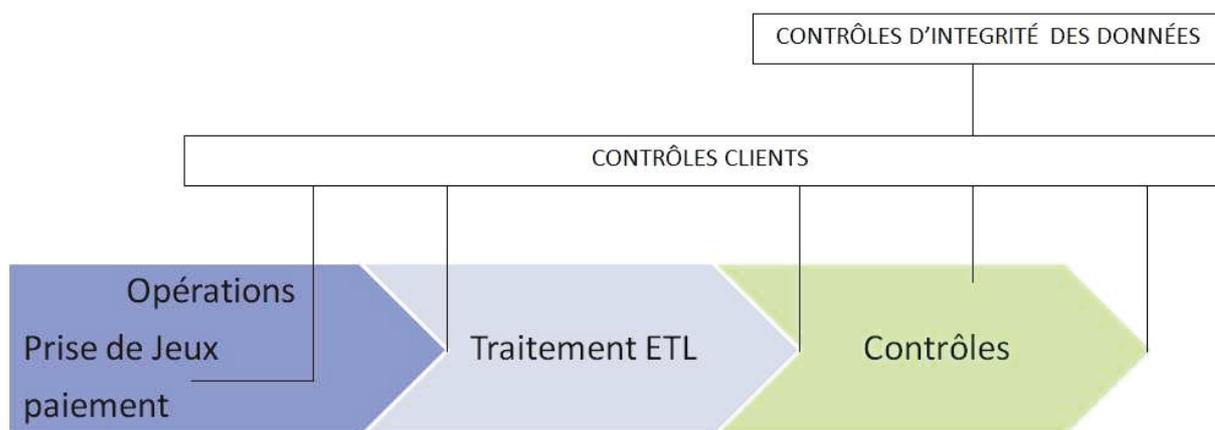


Figure 3-13 Schéma détaillé du processus avant contrôles et les types de contrôles associés

Le schéma technique, ci-dessous, montre le parcours de l'information au travers de l'infrastructure de la Française Des Jeux. Les données du site ParionsWeb sont remontées vers le service de comptabilité. Ce schéma nous montre où nous devons effectuer les contrôles d'intégrité des données.

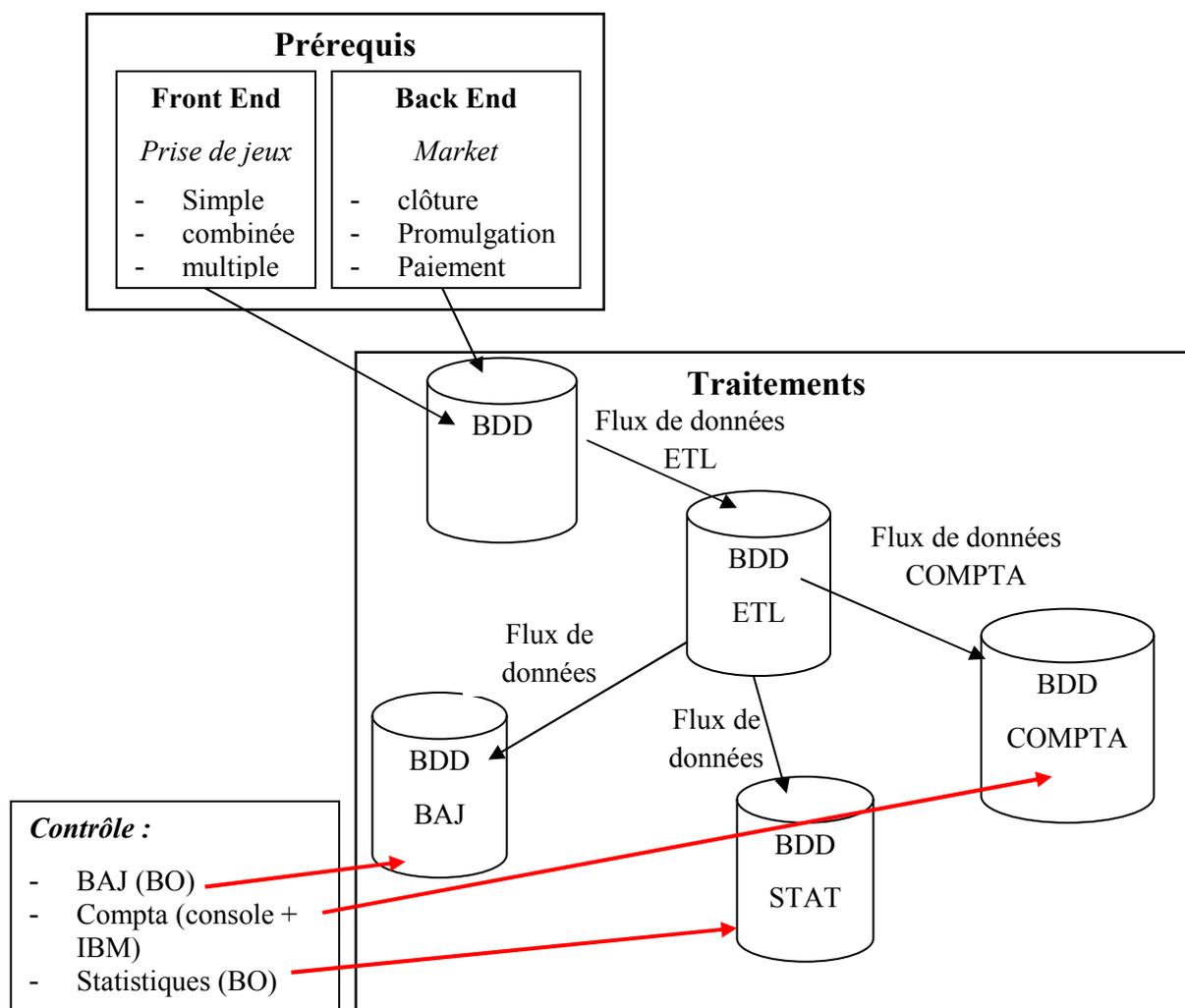


Figure 3-14 Schéma technique du transfert des données comptables

f **Détails du processus de test comptable manuel**

Liste des Prérequis

1. **En base de données**

Les données affichées sur les différents supports comptables, ne donnent pas d'information sur le joueur ayant fait une prise de paris. Ces dernières sont regroupées par « région ». Ainsi lors des tests comptables, comme nous ne pouvons distinguer un joueur d'un autre, l'environnement sur lequel

nous effectuons des prises de paris comptables est réservé à la comptabilité afin d'éviter de fausser les résultats comptables. Il est également nécessaire de nettoyer l'environnement de toutes les prises de paris résiduelles. Les prérequis au test comptable sont donc :

- Purger les données pour éviter des conflits de données sur les prises de paris encaissées non participantes.
- Aucune prise de paris autres que celles prévues dans ce contrôle ne doivent être effectuées sur l'environnement de test choisi.

2. Pour effectuer des prises de paris

- La personne en charge du test passe par des comptes joueurs créés manuellement, *via* le site marchand²⁴. Puis, elle valide ces joueurs, sur l'environnement d'intégration, pour les régions suivantes²⁵ : Métropole, DOM, St Pierre et Miquelon, Monaco, St Bart, St Martin. Le choix des régions a été défini dès le départ dans le test comptable.
- Ces comptes joueurs doivent avoir assez d'argent pour pouvoir effectuer toutes les prises de paris nécessaires au test.
- Les limites de prises de paris de ces joueurs sont assez hautes pour pouvoir effectuer toutes les prises de paris nécessaires au test. L'exemple le plus probant est la limite hebdomadaire. Prenons le cas où l'on souhaiterait, sur une durée inférieure à 7 jours, faire plusieurs prises de paris avec un même joueur pour une somme totale dépassant les 1500 euros. Dans ce cas, le montant maximum des dépenses hebdomadaires que peut faire ce joueur est fixé à 1000 euros. Ce dernier est alors bloqué dès que cette limite est atteinte et le joueur ne pourra pas effectuer toutes les prises de paris souhaitaient.

Opérations manuelles

1. Coté back office d'ABP

- Des événements doivent être créés.
- Pour chaque événement, des « Markets » doivent être créés.
- Une cotation doit être effectuée sur chaque pronostic de chaque « Market ».
- Ces « Markets » doivent être ouverts. Une fois ouverts, ils seront automatiquement mis en ligne sur le front office de l'application. (un exemple le jeu de données est fourni dans le fichier nommé « plan de recette V.3.xls » donné dans l'annexe : Plan de recette comptable).

2. Coté front office d'ABP

- L'utilisateur doit être authentifié.
- L'utilisateur doit faire des prises de paris simples, combinées et multiples en fonction d'un jeu de test bien précis.
- Les prises de paris doivent être gagnantes, perdantes et annulées.

²⁴ Le site marchand est la plate-forme contenant toutes les informations « joueur » ainsi que l'historique de prise de paris.

²⁵ Pour créer un compte, un joueur doit être français et fournir une adresse valide en France ou dans ses DOM TOM. En intégration, la création des comptes respecte ces deux critères.

Liste des traitements

Les traitements se composent de deux parties :

- La première concerne les traitements réalisés par l'application ABP qui extrait les données de la base ABP vers une autre base AGP_ETL en fonction de « règles métiers » définies dans ABP.
- La seconde partie concerne le reste des traitements effectués tout au long de la chaîne de l'ETL. Elle est composée de différents flux de travaux (workflows)²⁶ qui en fonction des règles métiers, remontent les informations jusqu'à la base de données de comptabilité, de statistiques ou encore la Base Analyse Joueur.

Liste des contrôles de données à effectuer pour les tests manuels

1. Où sont contrôlées les données ?

Les données sont contrôlées sur 4 environnements différents :

- Fichier Consoles (fichier plat contenant des informations comptables),
- Fichier IBM (second fichier plat contenant des informations comptables),
- B.O (accès base de données),
- BAJ (accès base de données).

2. Quelles sont les données qui sont contrôlées ?

Les données contrôlées sont les mêmes d'un environnement à l'autre :

- le CA encaissé,
- le CA participant,
- le CA encaissé non encore participant,
- le CA remboursé,
- le CA participant net,
- les gains.

Un comparatif est fait entre les données fournies pour les prises de paris en entrée et les données récupérées sur les différents supports comptables. Les informations d'entrées sont stockées dans un fichier de format Excel.

Règles de gestion

1. CA encaissé

Il correspond aux mises totales encaissées

2. CA participant

²⁶ On appelle workflow ou « Flux de travail » est la représentation d'une suite de tâches ou opérations effectuées par une personne, une suite de personnes, un logiciel, etc. Le terme flow renvoie au passage du produit, du document, de l'information, etc., d'une étape à l'autre.

Il correspond au montant des prises de paris dont les pronostics ont été **clôturés, promulgués** ou **annulés, et payés**.

3. CA encaissé non encore participant

Il correspond au montant des prises de paris dont les pronostics sont **non clôturés** donc **non promulgués** ou dont une partie est **non promulguée** donc **non payée**.

4. CA remboursé

Il correspond au montant des paris dont les pronostics ont été **clôturés, annulés, et payés**.

5. CA participant net

Il correspond aux prises de paris dont les pronostics ont été **clôturés, promulgués** mais **pas annulés** et payés.

6. Gains

Ils correspondent au montant des prises de paris qui sont **clôturés, promulgués, payés et gagnants**. Les prises de paris annulées ne sont pas prises en compte.

3.4.2 Analyse technique comptable

a **Accessibilité des données (contraintes techniques)**

IBM

- 1^{ère} solution : Données accessibles *via* interface logicielle
- 2nd solution : Données accessibles en amont *via* des fichiers rapport générés par le workflow et déposés dans le répertoire : `../../../../infa_shared/cjeu/TgtFiles/save`

CONSOLES

- 1^{ère} solution : Données accessibles sous forme de fichiers sur un url web accessible depuis un navigateur
- 2nd solution : Données accessibles sous forme de fichiers dans le répertoire : `../../../../infa_shared/cjeu/TgtFiles`

BAJ

- Solution : Données accessibles depuis la base de données <nom de la base de données BAJ>

Statistiques (SSCCSTATS)

- Solution : Données accessibles depuis la base de données <nom de la base de données SSCCSTATS>

b Actions nécessaires pour automatiser de bout en bout la comptabilité

Environnement	Solution	Type d'accès	Actions à prévoir
IBM	2	Fichier	Demande d'accès en lecture aux fichiers
CONSOLES	2	Fichier	Demande d'accès en lecture aux fichiers
BAJ	1	BDD	Demande d'accès en lecture à la base de données <nom de la base de données BAJ>
SSCCSTATS	1	BDD	Demande d'accès en lecture à la base de données <nom de la base de données SSCCSTATS>

c Risques potentiels

Environnement	Solution	Type d'accès	Actions à prévoir
BAJ	1	BDD	Les requêtes sql dans B.O ne sont pas toujours accessibles car des modifications peuvent avoir lieu entre temps dans B.O. D'autre part, certains champs récupérés sont propres à B.O, ils ne sont pas directement récupérés depuis la base de données
SSCCSTATS	1	BDD	Les requêtes sql dans B.O ne sont pas toujours accessibles car des modifications peuvent avoir lieu entre temps dans B.O. D'autre part, certains champs récupérés sont propres à B.O, ils ne sont pas directement récupérés depuis la base de données

3.4.3 Automatisation

a Détails du processus de test comptable automatisé

Prérequis nécessaires

- Les données n'ont pas besoin d'être purgées sauf pour les comptes utilisés pour la comptabilité.
- Pour pouvoir effectuer toutes les prises de paris nécessaires au test, les joueurs doivent avoir assez d'argent sur leur compte.
- Les prises de paris effectuées doivent être gagnantes, perdantes et annulées.

Actions automatisées

1. Coté back office d'ABP

- Le programme crée 7 « Markets » de type 1N2 dont deux pour du Live. La création de l'offre sportive peut se limiter à la création de « Markets » de type 1N2²⁷ pour un seul sport donné. Pour rappel, une formule de paris sportifs est appelée Market dans l'application ABP.
- Le programme fait la cotation de chaque pronostic pour chaque « Market ».
- Le programme met en ligne ces « Markets ».

2. Coté front office d'ABP

- Le programme utilise un compte joueur pour s'authentifier.
- Le programme fait des prises de paris simples, combinées et multiples en fonction d'un jeu de test bien précis.

Liste des traitements

Les traitements de diffusion des données ne changent pas. Avant de continuer le processus d'automatisation, nous devons nous assurer que ces traitements ont bien été effectués.

Les traitements sont :

- Traitement d'export ETL.
- Workflow ETL: _STAG_DETAIL et AGGREGAT²⁸.

Liste des contrôles de données à effectuer

D'après les éléments fournis par l'analyse fonctionnelle et technique précédente, nous avons délimité le périmètre des tests. La recherche d'informations et la demande d'accès à ces environnements semblaient dans un premier temps compliquées. Nous avons donc décidé de limiter les contrôles uniquement à la base de données d'ABP ainsi qu'à la partie ETL à la charge du département HMOA Multimédia Sport.

Les 3 bases de données concernées par les contrôles sont :

- ABP : Base de données du site Internet *ParionsWeb*.
- AGP_ETL qui reçoit les données extraites de la base de données ABP par le processus d'export.
- ETL qui contient les données extraites de la base de données AGP_ETL *via* les traitements workflow de traitement ETL.
- Le contrôle ne tiendra pas compte du cas du « resettlement »²⁹.
- Les contrôles en base de données se limiteront au login utilisé pour se connecter.

²⁷ Une explication du fonctionnement d'un « Market » 1N2 a été donnée au début du document (p 15).

²⁸ Nom donné aux workflows responsables de l'export des données

²⁹ Le « resettlement » est un processus qui intervient lorsqu'il y a eu erreur sur le résultat d'un match terminé (erreur de saisie), que les joueurs ont été payés et que l'on souhaite modifier le résultat et régulariser les gains. Suite à ce processus, certains gagnants peuvent devenir perdants et certains perdants peuvent devenir gagnants.

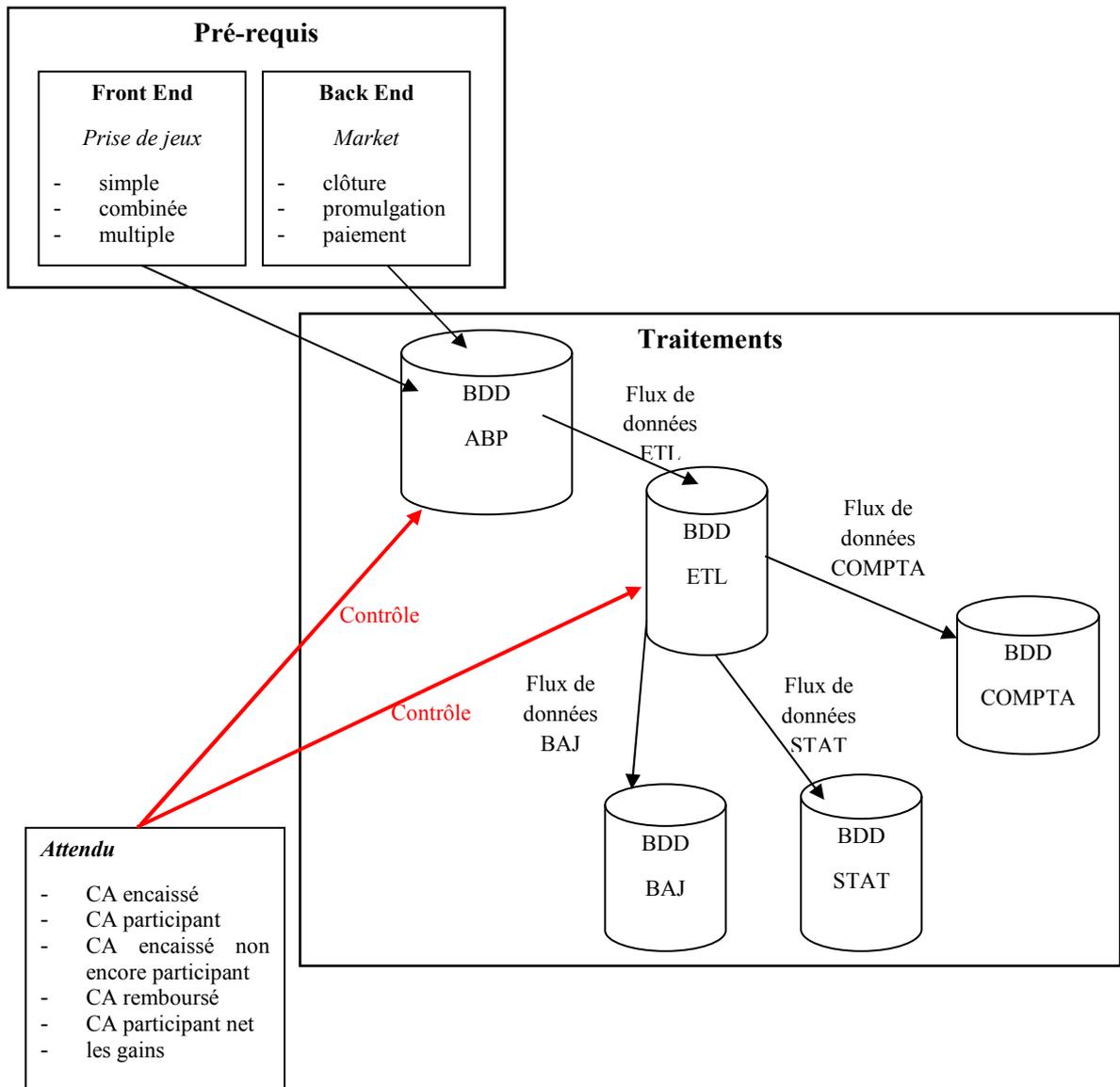


Figure 3-15 Schéma technique du transfert des données comptables et des contrôles associés

b *Architecture*

Modules Utilisés

Fonctions /Modules	Create Market	Generate injecter	Injecter	Cloture Market	Quality center	Compare Reporter	Injecteur Accounting
Authentification			X				
Prise de paris			X				
Créer un « Market »	X						
Coter un « Market »	X						
Ouvrir un « Market »	X						
Clôturer un « Market »				X			
Promulguer un « Market »				X			
Annuler un « Market »				X			
Payer un « Market »				X			
Ordonnancer les différents modules					X		
Planifier l'exécution des tests					X		
Contrôler les résultats comptables ABP							X
Contrôler les résultats comptables ETL						X	

c *Mise en place*

Détail du processus

Le test automatisé se fait sur 4 jours comme pour le test manuel. Chaque nuit, les traitements tournent, exportent les données de la base de données ABP vers la base de données ETL et les contrôles s'effectuent le lendemain dans la base de donnée ETL.

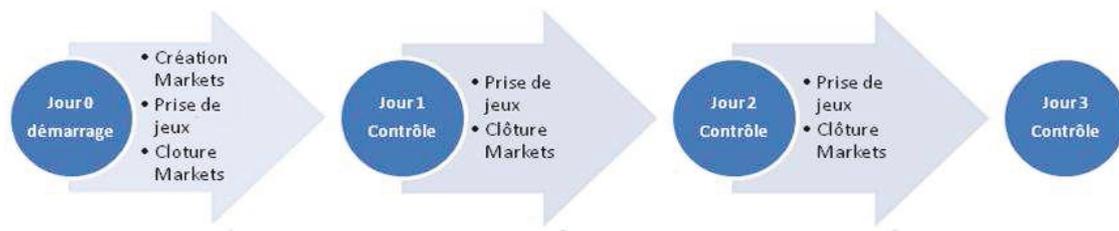


Figure 3-16 Schéma représentatif de l'enchaînement des actions du test comptable sur 4 jours

Relancer plusieurs fois les tests d'automatisation

Une réflexion sur l'architecture de chaque module a été faite pour qu'ils puissent être rejoués sans devoir à chaque fois modifier les données en entrée. L'exemple le plus probant est la date de création et de clôture des « Markets » qui est calculée en fonction de la date du jour et d'un offset fourni en entrée.

Ordonnancer les différentes actions

Le choix d'une architecture modulaire, et la décomposition des actions de création de « Markets », de prises de paris, de clôture de « Markets » en modules imposent l'utilisation d'un outil capable d'enchaîner ses différents modules (actions) pour créer un processus : *le processus comptable*.

Cet outil devra ordonnancer, planifier et exécuter les différents modules.

3.4.4 L'« ordonnanceur »

L'« ordonnanceur » est la véritable colonne vertébrale de l'automatisation du test comptable. Cette fonction qui est réalisée à l'aide d'un outil a pour but d'effectuer une suite logique d'opérations, d'enchaîner les uns après les autres les différents modules créés pour l'automatisation. Ces différents modules indépendants les uns des autres communiqueront par un échange d'informations tout au long du processus d'automatisation.

Plusieurs choix s'offrent à nous :

- L'outil HP Quality Center (QC) peut créer des tests dits « automatisés », c'est-à-dire appeler des scripts externes, les ordonnancer et les exécuter.
- Control'M peut correspondre. Il est déjà utilisé à la Française Des Jeux pour l'ordonnancement des traitements de l'ETL. L'inconvénient à utiliser cet outil est que nous n'avons pas les droits d'administration. Son utilisation reste toutefois intéressante dans le cas où nous souhaitons lancer le test comptable à date fixe. A cela, un tel lancement ne peut être fait que lorsque toutes les actions du processus comptable sont automatisées y compris les actions de la phase de prérequis.
- L'utilisation d'un logiciel externe.

Nous avons décidé d'utiliser l'outil d'ordonnancement fourni par QC pour trois raisons :

- La première, est que cet outil est complémentaire à l'outil Quick Test Professionnel(QTP) et offre la possibilité d'appeler directement des scripts QTP, d'échanger très facilement des informations telles que les paramètres d'entrée, de récupérer les valeurs et commentaires de sortie des scripts.
- La deuxième, est que l'outil QC offre la possibilité d'exécuter à distance depuis n'importe quel ordinateur des scripts présents sur une machine dédiée à l'automatisation et appelée serveur QTP.
- La troisième, est la facilité de réutilisation des scripts pour d'autres besoins.

a Fonctionnement de Quality Center (QC)

L'outil QC est une plateforme utilisée par l'équipe HMOA Multimédia Sport pour préparer et réaliser les tests manuels de non-régression techniques. Lors de la qualification d'une partie de l'application, plusieurs étapes (« steps » en anglais) sont effectuées manuellement. Un test dans QC est le regroupement de plusieurs étapes. Il est rédigé par une personne dans la partie appelée « TestPlan », puis chargé dans la partie « TestLab » pour être exécuté par une ou plusieurs autres personnes. Au fur et à mesure que la personne avance dans les étapes, cette dernière indique si l'étape s'est correctement déroulée ou non. Pour savoir si une étape s'est correctement déroulée, la personne compare avec ce qui est attendu.

The screenshot shows the Quality Center interface with a test plan. A table of test steps is visible, with callouts pointing to specific parts of the interface:

- Nom du test:** Points to the 'Subject' field in the test plan tree.
- Détail des étapes du test:** Points to the 'Step Name' column in the table.
- Liste des actions à effectuer dans cette étape:** Points to the 'Description' column in the table.
- Résultat attendu suite aux actions de cette étape:** Points to the 'Expected Result' column in the table.

Step Name	Description	Expected Result
Preamble: login	Fill the Username and Password fields with the values "Caroline" (password "pass"). Then click the "Login Now" button.	
Preamble: enter Time	Identify the line matching the task "Phoning". Then enter "5" in the box corresponding to the date Sunday.	The displayed message is equal to "MODIFICATIONS NOT SAVED". The remaining time for Sunday must be 5 hour(s) Time used on task Phoning for Sunday must be 5 hour(s)
Body: saveTimeTrack	Click the "Save Time-Track" button.	The displayed message is equal to "Time-Track was successfully saved".
Postamble: removeTime	Identify the line matching the task "Phoning". Clear the box corresponding to the date Sunday.	The displayed message is equal to "MODIFICATIONS NOT SAVED". The remaining time for Sunday must be 0 hour(s)
Postamble: saveTimeTrack	Click the "Save Time-Track" button.	The displayed message is equal to "Time-Track was successfully saved".
Postamble: logout	Click the "Logout" button in the upper right corner of the current page.	

Figure 3-17 Illustration des étapes d'un test dans Quality Center

En plus de pouvoir effectuer des tests manuels, l'outil offre la possibilité de créer des tests dit « automatisés »; c'est à dire d'exécuter directement des scripts PERL, VBScript, ou encore Quick

Test Professionnel, et donc indirectement de lancer tout type de script³⁰. Par exemple, l'appel d'un script « Windows » peut se faire grâce à un script VBScript.

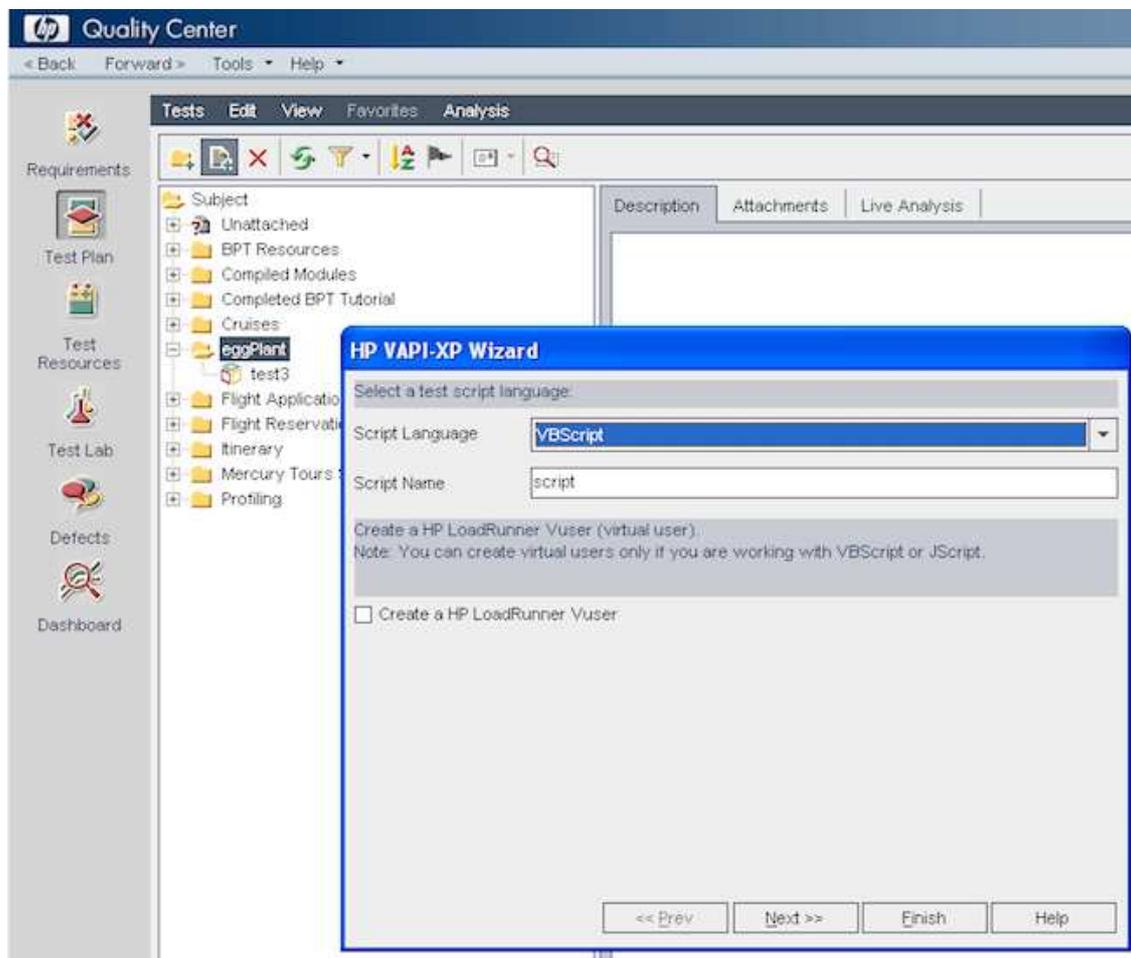


Figure 3-18 Illustration création d'un test automatisé VBScript dans Quality Center

« TestPlan » est la partie dans QC où l'on crée les tests manuels ou automatiques.

« TestLab » est la partie dans QC où l'on charge puis exécute les tests créés dans le « TestPlan ». Ces tests sont regroupés dans ce que l'on appelle un « TestSet » ou ensemble de tests où l'on regroupe les tests qui servent à qualifier une même partie de l'application, une même fonctionnalité.

³⁰ Un exemple de création de test automatisé est disponible à l'URL suivante :
<http://www.testplant.com/support/2011/03/integrating-eggplant-with-hps-quality-center/>

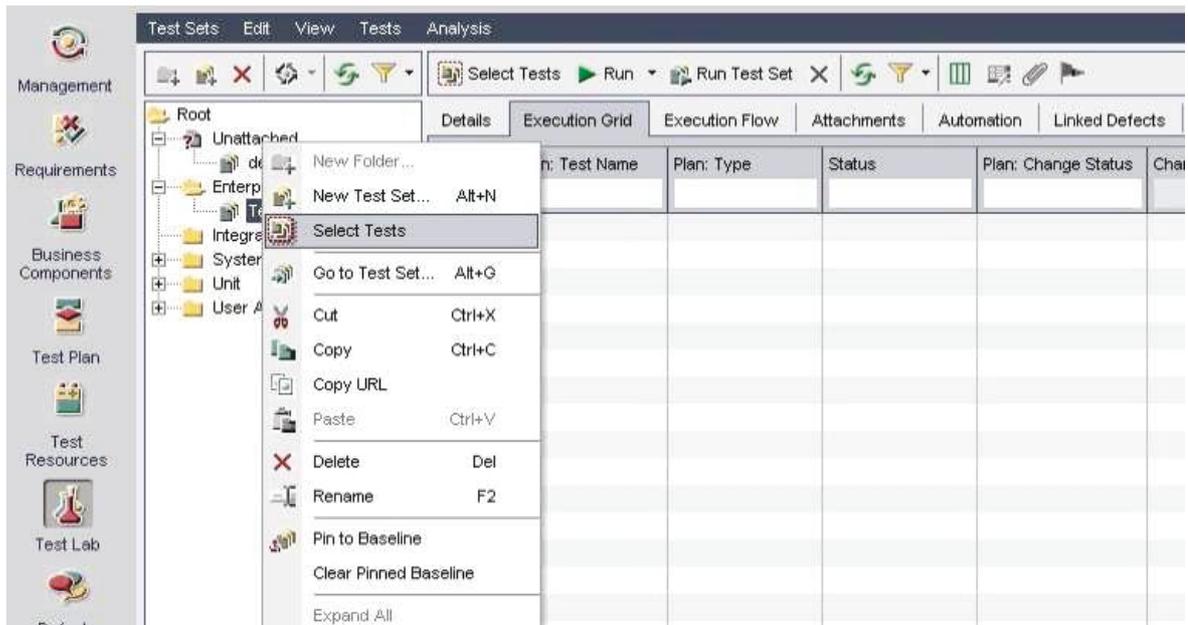


Figure 3-19 Illustration du chargement d'un test dans un « TestSet »

Pour l'automatisation comptable, deux types de test automatisé sont nécessaires :

- des tests automatiques VBScripts créés dans QC et qui appellent des scripts sur la machine locale du testeur ;
- des scripts QTP qui sont importés dans QC.

Pour des raisons d'indépendance vis-à-vis de la plateforme QC, l'importation des scripts dans QC ne contient pas les bibliothèques associées. Les bibliothèques sont appelées depuis la machine locale du testeur.

Le schéma ci-dessous donne une représentation simplifiée du processus d'exécution des tests automatisés de QC.

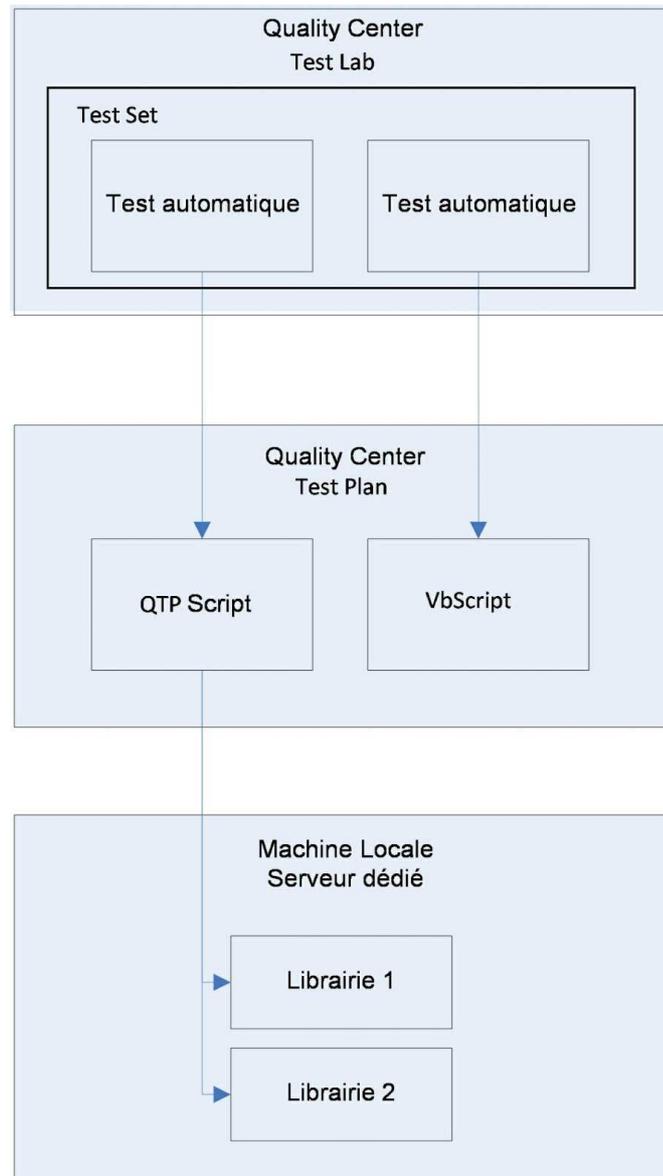


Figure 3-20 Echange des données entre Quality Center et la machine locale

1ère étape : Création de tests automatiques dans le « TestPlan »

Lors de la création des tests, et si cela est nécessaire, un appel aux scripts QTP, ou « Windows » peut être fait. Les paramètres propres aux tests sont enregistrés à cette étape et leurs valeurs sont fournies lors de la création du « TestSet ».

Dans le cas de l'automatisation, un test automatisé ne possède qu'une seule étape servant à appeler le module automatisé.

2ème étape : Création du « TestSet » dans le « TestLab »

Le « TestSet » est un enchaînement de plusieurs tests (ou modules pour notre cas).

Pour le créer, nous chargeons les instances³¹ des tests déjà créés au préalable dans le « TestPlan ». L'objet test possède des propriétés propres et la valeur de chacun de ses paramètres est définie à cette étape. Plusieurs « TestSet » peuvent charger un même test et définir différentes valeurs pour ses paramètres.

3ème étape : Ordonnement des tests dans le « TestSet »

L'ordonnement se définit lors de la création du « TestSet ».

Une campagne de non-régression utilise un ou plusieurs « TestSet » dont l'ordonnement est défini à l'avance. Lors de cette étape, nous définissons les contrôles clients entre chaque test. Ces contrôles gérés au niveau de QC servent pour l'enchaînement des modules. Ils permettent de contrôler le bon déroulement d'un test et sous quelle condition le suivant doit être lancé. Nous pouvons choisir ici qu'un test ne s'exécute si et seulement si le test précédent n'a pas échoué.

4ème étape : Lancement de l'exécution du « TestSet »

Une fois les « TestSet » choisis, l'utilisateur doit définir la date et l'heure de lancement du premier test de chaque « TestSet ». L'enchaînement des autres tests se fait de manière automatique et suivant les règles d'enchaînement choisies. Exemple : le test (module) d'injection de l'offre ne doit pas s'exécuter si le test (module) de création de l'offre a échoué.

b Les Paramètres utilisés en entrée

Paramètres définis dans les tests du « TestSet »

Ces paramètres sont saisis manuellement et définis à l'avance dans Quality Center pour chaque test du « TestSet ». Ces paramètres sont transmis aux scripts appelés par le test. La modification de ces paramètres est manuelle, donc ces paramètres sont de préférences globaux.

Par exemple : le paramètre « fileAuthParam » sera le fichier contenant les paramètres dont le script d'authentification aura besoin pour fonctionner.

Paramètres nécessaires au fonctionnement des scripts

La valeur des variables propres au fonctionnement des scripts QTP ou PERL se doit à la fois de pouvoir être modifiée rapidement et rester inchangée entre deux exécutions. L'utilisation d'un fichier contenant le paramétrage de ces variables doit par conséquent être mis en place à cet effet. Un fichier de configuration devra être mis en place pour définir les variables des scripts PERL et un autre pour définir les variables utilisées par les scripts QTP.

³¹ En programmation objet, une classe sert à définir toutes les fonctionnalités et les comportements d'un objet, l'instance les réalise. On peut créer plusieurs instances d'une même classe.

c Le traitement des données en sortie

Toutes les erreurs survenues durant l'exécution des scripts doivent être stockées et accessibles une fois les traitements terminés.

Dans le cas de l'utilisation d'un script QTP, le rapport d'erreur est visible depuis QC.

Le choix a été fait d'arrêter le « TestSet » lorsque le module associé tombe en erreur. La valeur de succès ou d'échec doit donc être remontée des modules vers les scripts du « TestPlan » définis dans QC.

Ce test étant lié au module de création de l'offre, si le module (appelé script dans le schéma ci-dessous) tombe en erreur alors le test le détecte et tombe aussi en erreur.

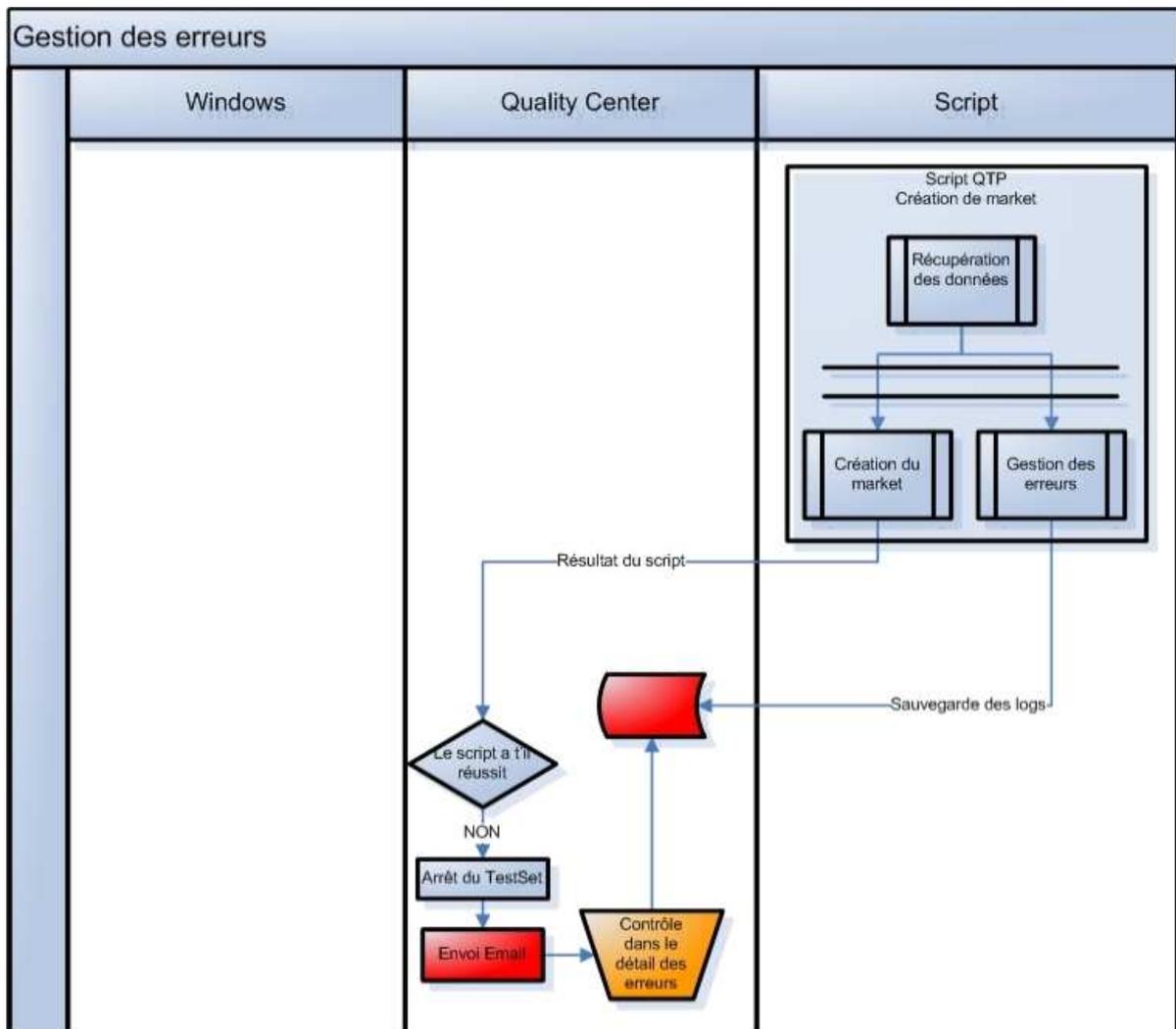


Figure 3-21 Diagramme de flux gestion des erreurs

d Ordonnement des Tests

La campagne de tests de non-régression comptable s'étale sur 4 jours consécutifs avec des actions qui doivent être effectuées quotidiennement. L'ordonnement des tests se base sur le contrôle de la valeur des tests lancés depuis QC : succès ou erreur.

Les tests prévus au début d'une nouvelle journée de traitement forment le point de départ des traitements de la journée.

Deux conditions sont nécessaires à leur démarrage :

- leur date de planification. Cette date est définie manuellement avant le lancement du « TestSet » ;
- la condition de succès du test précédent.

e *Détail des processus mis en place*

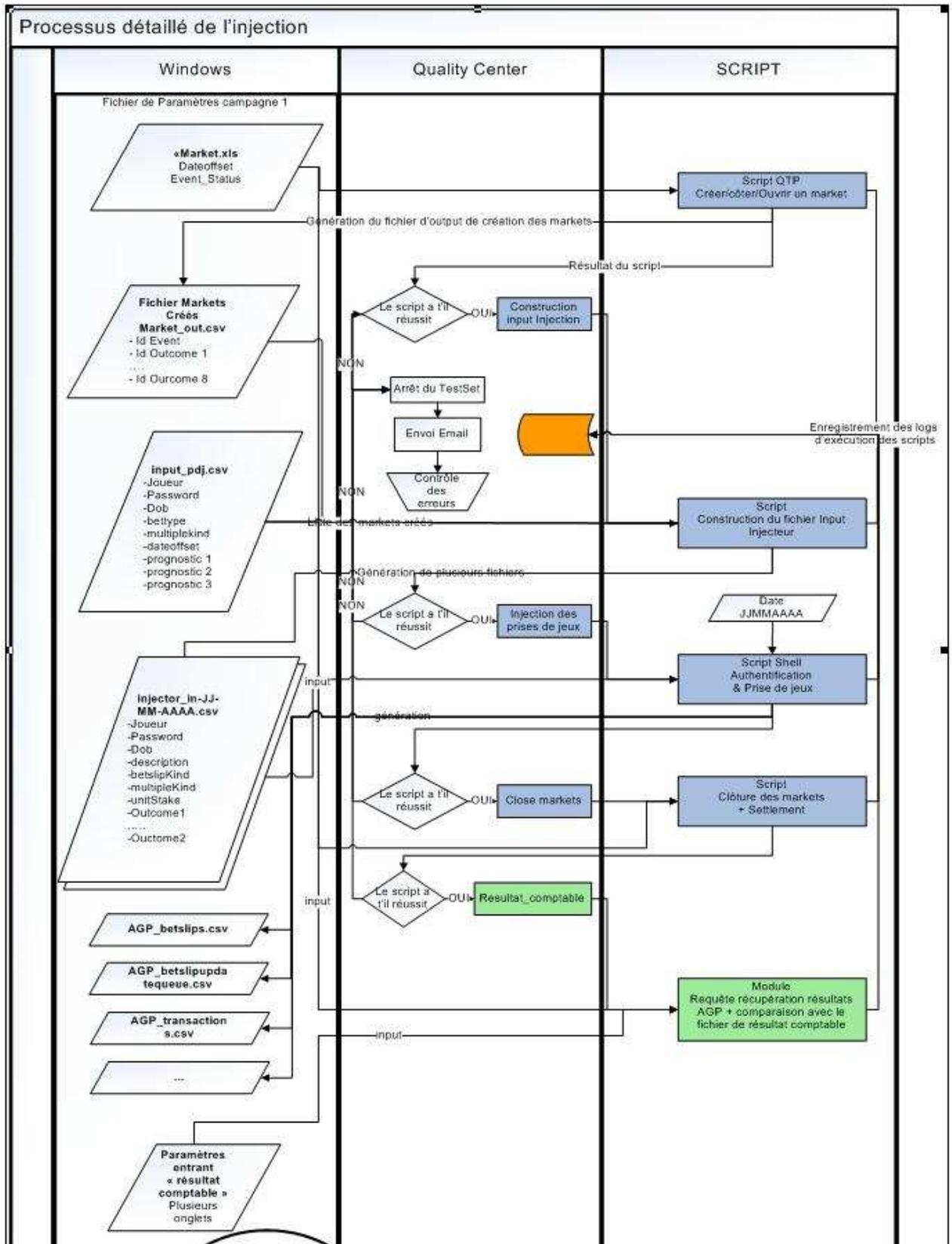


Figure 3-22 Diagramme de flux du processus du premier jour d'automatisation comptable

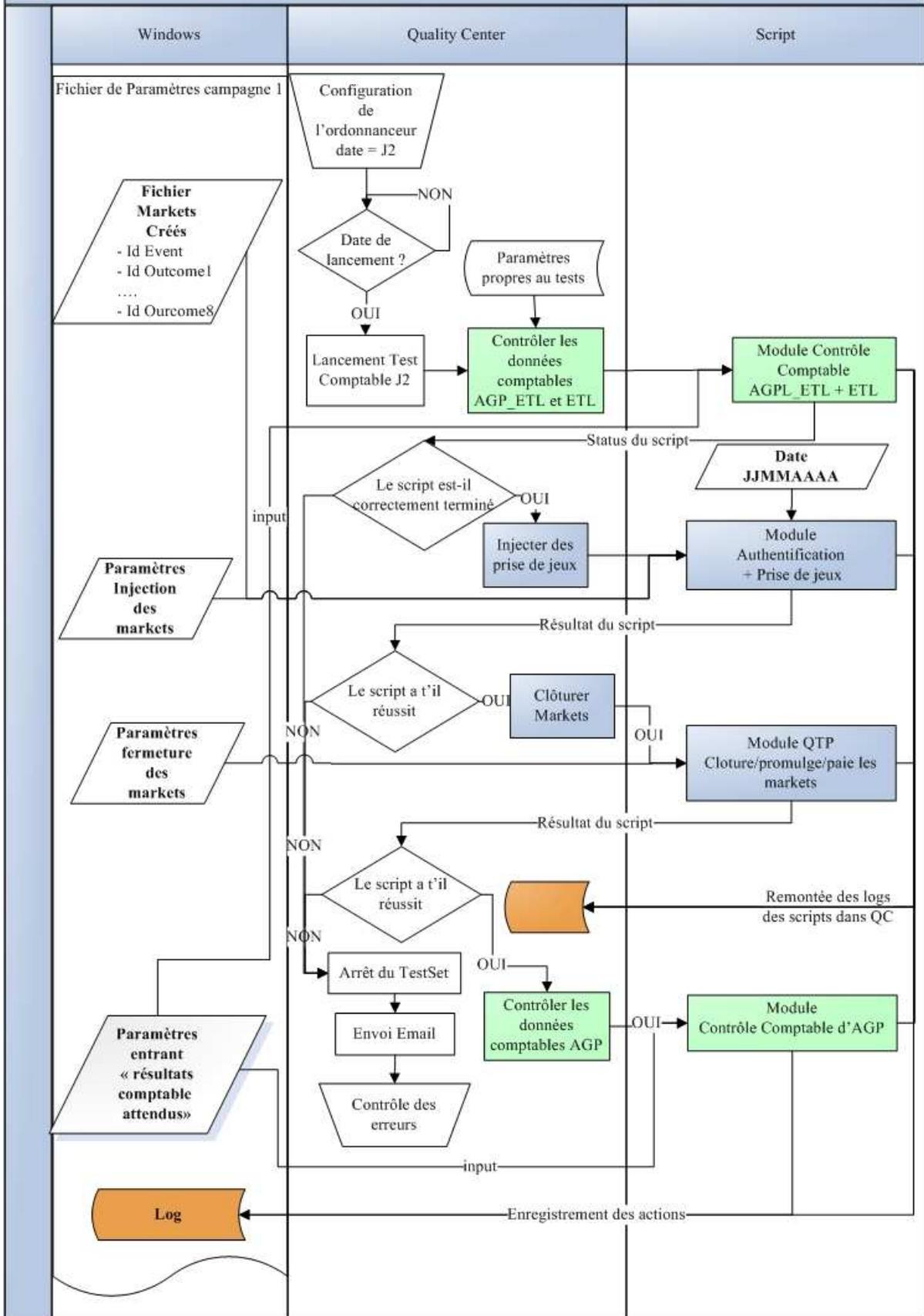


Figure 3-23 Diagramme de flux du processus du deuxième jour d'automatisation comptable

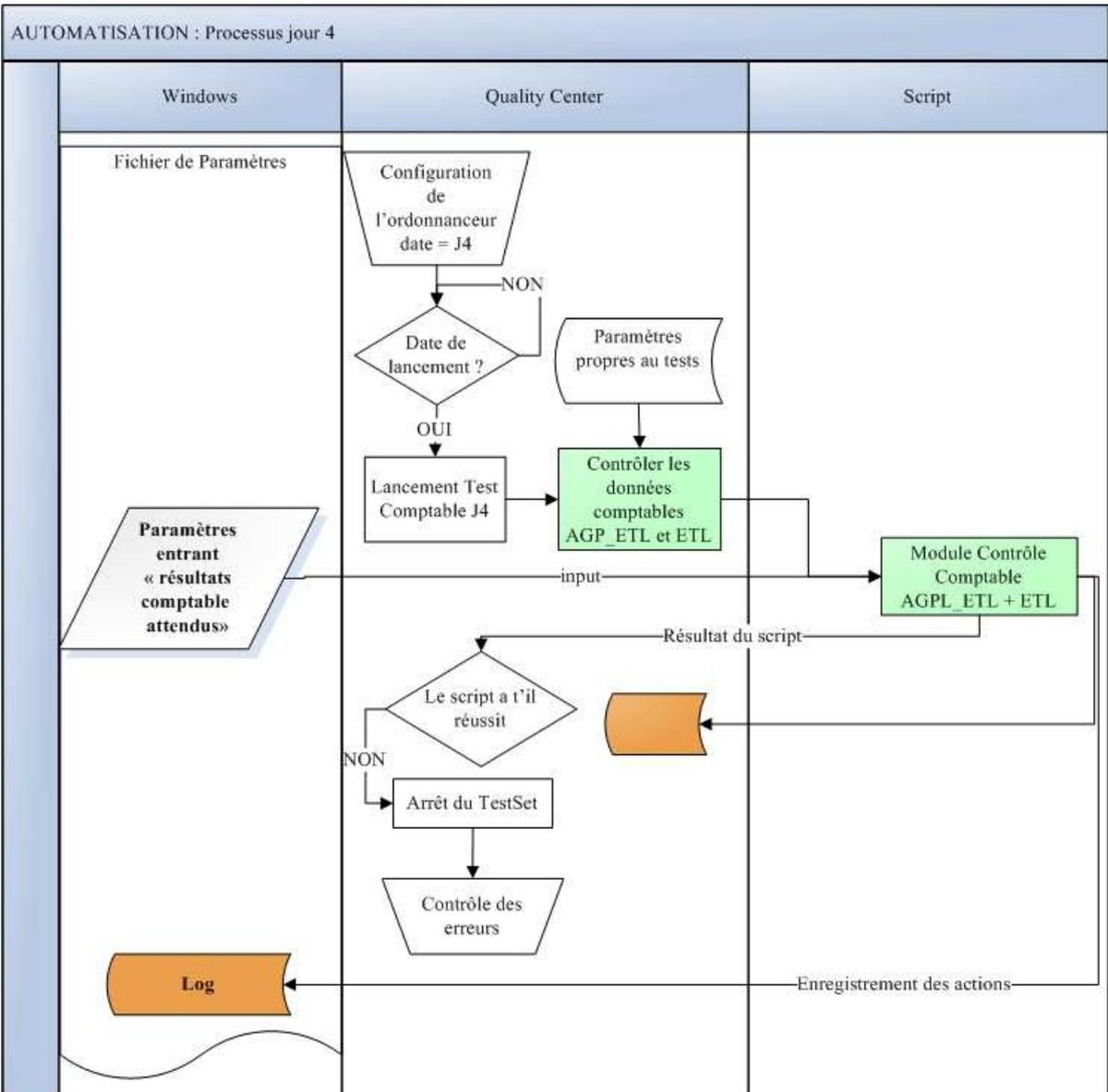


Figure 3-25 Diagramme de flux du processus du quatrième jour d'automatisation comptable

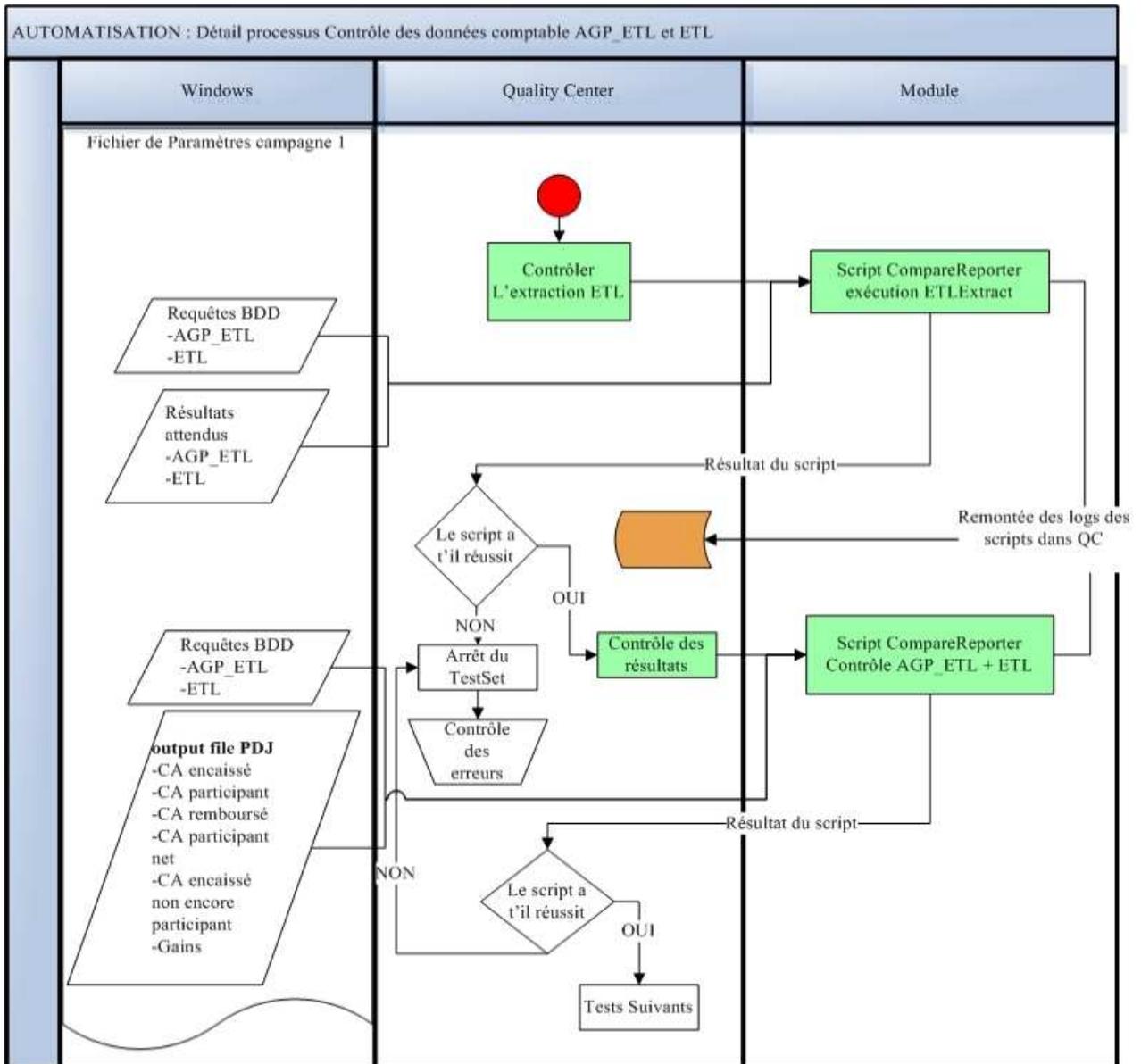


Figure 3-26 Diagramme de flux détaillé du processus de contrôle Comptables AGP_ETL et ETL

Cette étude a apporté toutes les informations nécessaires aux développements de l'ensemble des modules à réaliser. Néanmoins pour plus d'agilité, nous avons commencé le développement des premiers modules sans attendre que cette étude ne soit achevée. La décomposition des fonctions en sous-fonctions, le regroupement par module ainsi qu'une ébauche de l'enchaînement des modules ont fourni assez d'informations pour que l'on puisse commencer les développements. Le reste de l'étude s'est déroulée parallèlement aux premiers développements précisant la manière dont les modules doivent être ordonnancés.

Les étapes de réalisation sont détaillées dans le paragraphe suivant.

3.5 Phase de réalisation, d'intégration et de qualification continue

La phase de réalisation du projet s'est étalée sur plusieurs itérations et l'utilisation de Scrum s'est faite de manière progressive avec la mise en place des outils et actions associés à ce framework.

Le premier outil mobilisé a été le planning Scrum mis en place au début de la deuxième itération. Ce dernier comprend le **graphique d'avancement** de l'itération (Burdown Chart) et le **carnet du produit** contenant la liste des fonctionnalités à développer (Backlog du produit).

Ensuite, la mêlée quotidienne (ou Daily Scrum) a été mise en place au démarrage de la cinquième itération.

3.5.1 Définition de Scrum

Nous avons vu précédemment, les raisons pour lesquelles l'utilisation du framework Scrum a été choisie dans la gestion du projet d'automatisation. Désormais, nous allons voir quels sont les outils qui ont été utilisés pour basculer vers une gestion de projet de type Scrum. Mais avant d'explicitier plus en détail la mise en place de ces outils, il est nécessaire de connaître Scrum et son fonctionnement.

a ***Qu'est-ce que Scrum ?***

Scrum est un framework agile dédié à la gestion de projet. Scrum tire son nom du terme anglais « mêlée », au Rugby. Ce nom a été choisi par analogie entre les réunions quotidiennes de Scrum et la mêlée, cette technique de reprise du jeu après une faute qui remet une équipe sur de bons rails par un effort collectif. Conçue en 1993 et formalisée en 1995, cette méthode de développement (orientée gestion de projet) inclut souvent des pratiques venant de XP. Dans notre cas, nous n'avons pas associé Scrum à une autre méthodologie de gestion de projet de type XP. Nous verrons cependant dans la suite de ce document que cette association pourrait se révéler bénéfique.

L'idée de Scrum est de tenir compte de la réalité de la plupart des projets pour lesquels il n'est pas possible de tout définir dès le début : les spécifications seront modifiées et précisées, des outils ou technologies inconnus entreront en jeu, etc. De fait, pour s'adapter aux changements, Scrum ne suit pas un processus prédictif et les travaux à faire sont ajustés régulièrement au cours du projet, notamment à la fin de chaque itération, appelée le « Sprint ».

b **Fonctionnement de Scrum**

Scrum suppose donc une intense collaboration entre les différentes personnes impliquées.

Le propriétaire du produit (« Product Owner ») est le représentant des clients et des utilisateurs, il définit les priorités pour la réalisation.

Le « gestionnaire » est nommé « ScrumMaster » : il a pour charge de faciliter l'application de Scrum par l'équipe.

L'équipe s'engage pour la réalisation de fonctionnalités et le ScrumMaster la motive pour y arriver.

Le tout représentant l'équipe Scrum.

Le processus Scrum repose sur deux journaux ou « Backlog » :

- Backlog de produit : une liste des fonctionnalités pour le produit, définie par le directeur de produit (« Product Owner »). Dans le cadre de notre projet, les fonctionnalités ont été définies après discussion avec l'équipe de développement et proposées au Product Owner.
- Backlog de Sprint : recense les tâches du Sprint en cours.

Un projet utilisant Scrum a son cycle de vie composé de Sprints successifs. Un Sprint dure au plus quatre semaines. Pendant un Sprint, des réunions quotidiennes de moins de 15 minutes (appelées Scrum) permettent à toute l'équipe de faire le point sur le travail accompli par chacun depuis la dernière réunion Scrum, les obstacles rencontrés, et le travail prévu d'ici la prochaine réunion - rien de plus.

Pendant un Sprint, l'équipe développe un produit partiel. Elle déroule toutes les activités nécessaires pour cela : analyser, concevoir, développer, tester, documenter et intégrer. Chaque Sprint se termine par une revue de Sprint, pour que le directeur de produit évalue, au cours d'une démonstration, le produit partiel obtenu et modifie au besoin le Backlog de produit.

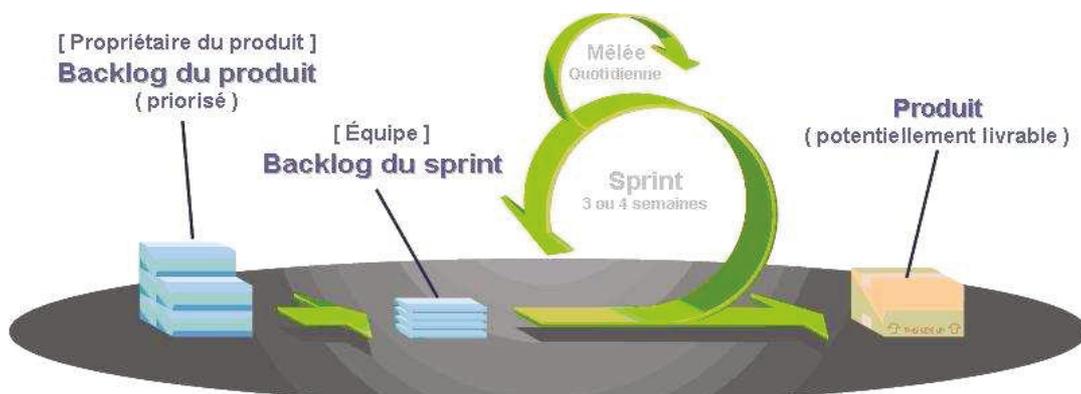


Figure 3-27 Vue synthétique processus Scrum

3.5.2 Mise en application de Scrum : utilisation des outils

a *Outil pour la gestion administrative d'un projet*

Le choix du planning a été très simple au départ. Le projet étant agile, un planning classique avec l'aide d'un logiciel tel que Microsoft Project n'a pas été retenu. En effet, un tel planning nous semble plus utile pour des projets classiques et de plus grosse envergure.

Pour ma part, du fait de notre faible maîtrise initiale du langage et des outils développés en QTP et PERL, l'évaluation de la charge de travail pour la réalisation de telle ou telle tâche peut être remise en cause à la première complication se présentant. Sa durée nécessitera d'être rallongée ou une nouvelle tâche devra être créée. Bien qu'utile pour l'ordonnancement des tâches, un outil comme *Microsoft Project* n'apporte pas la souplesse attendue. C'est pourquoi il a été décidé d'utiliser un planning créé depuis Excel et de le partager sur un serveur de subversion³² grâce auquel il est aisé de rajouter des tâches sans définir leur date de démarrage.

D'autre part, la centralisation de toutes les informations relatives au projet dans un unique et même endroit est souhaitable afin de faciliter sa gestion. Les onglets Excel ont été très utiles pour séparer les différentes informations.



Nous avons :

- un onglet pour le Backlog du produit ;
- un onglet par Sprint ;
- un onglet pour le Bug Review (utilisé dans la deuxième étape du projet) ;
- un onglet pour comptabiliser le temps de gestion de projet car le planning de Sprint est uniquement un planning de production et non d'analyse ou de gestion ;
- un onglet pour les congés ;
- un onglet pour le calcul du budget ;
- un onglet TODO qui a été très vite remplacé par l'onglet Backlog.

Les développeurs peuvent ainsi mettre à jour le planning simplement en cochant la cellule concernée. Il est possible par exemple de réaliser une tâche à la place d'une autre, cette dernière pouvant être initialement prévue en début de semaine mais accomplie en fin de semaine. Simple d'emploi et facilement mis à jour, une telle centralisation répond aux attentes de notre mode de gestion.

³² *Subversion* est un logiciel de gestion de versions, distribué sous licence Apache et BSD.

Ce planning a été amélioré au fur et à mesure du projet pour convenir aux besoins inhérents. Pour le second Sprint, ce dernier a été complètement modifié afin de faciliter :

- le rajout de tâches ;
- l’affichage de la courbe d’avancement du projet et plus particulièrement du reste à faire ;
- l’attribution d’une tâche ;
- le statut d’une tâche ;
- l’autonomie de l’équipe vis-à-vis du planning ;
- et pour toutes ces raisons, la mise en pratique de la méthodologie Scrum.

Tout ceci, a été fait dans l’idée que l’équipe de développement soit autonome vis-à-vis du planning et se charge de le mettre à jour quotidiennement. En effet, une fois que le ScrumMaster a défini et rajouté les tâches pour le Sprint, généralement en début de celui-ci, les développeurs se chargent, en fonction de leurs avancements, de s’attribuer les tâches de développement restantes. L’équipe est ainsi autonome dans le travail qu’elle effectue.

Néanmoins, une règle est à respecter : si une tâche doit être rajoutée pendant une itération, ce rajout est fait en accord avec le ScrumMaster et le Product Owner. L’équipe de développement n’a pas le droit de rajout ou de suppression des tâches du planning du Sprint.

b ***Backlog du produit***

Le Backlog du produit³³ contient toutes les tâches auxquelles l’équipe Scrum a pensé.

Afin de faciliter les choses, la charge de travail pour une tâche est estimée en jours. Un responsable est associé rétroactivement à une tâche. Lorsque les tâches sont en attente dans le Backlog, celles-ci ne sont pas encore associées à un responsable.

Tâche	Module	Type tâche	Priorité	Poids	Responsable	Statut	Sprint
Documentation technique Sprint 1 - Amine	Tous	Documentation	-	3,5	Amine Al Oumami	Testé et Validé	
Gestion multi joueur - Ajout fonctionnalité dans l’injecteur	Entrées / Sorties	Développement	-	3,5	Amine Al Oumami	Testé et Validé	
Génération des fichiers d’inputs pour l’injecteur - Création composant	Entrées / Sorties	Développement	-	2,5	Amine Al Oumami	Testé et Validé	
Analyse comptable - Création d’un Composant	Entrées / Sorties	Développement	-	3	Amine Al Oumami	Testé et Validé	

³³ Expliqué page 102

c **Backlog du Sprint**

Le Backlog du Sprint n'est rien d'autre que l'ajout au Sprint en cours des tâches en attente dans le Backlog du produit en accord avec le Product Owner.

Tâche	Module	Type tâche	Responsable	Statut
Nouvelle version 10,4,2 - Mise à jour composant QTP-PERL	Entrées / Sorties	Développement	Amine Al Oumami	Terminé
Analyse Comptable - Analyse et création requêtes pour ETL	Entrées / Sorties	Analyse	Philippe Amerio	Terminé
Analyse Comptable - Création composant pour analyse comptable ETL	Entrées / Sorties	Développement	Amine Al Oumami	Terminé

Les tâches contenues dans le Backlog de Sprint sont associées à un responsable lorsqu'un développeur s'approprie la tâche. Il met à jour le statut de la tâche et la passe en statut « Terminé » lorsque celle-ci est achevée. Lors du Sprint 6, un statut a été rajouté pour les tâches de développement. Celui de « Testé et validé » qui garantit qu'une tâche est faite à 100%.

Ci-dessous, une vue d'ensemble du planning mis en place lors du deuxième Sprint et utilisé jusqu'à présent :

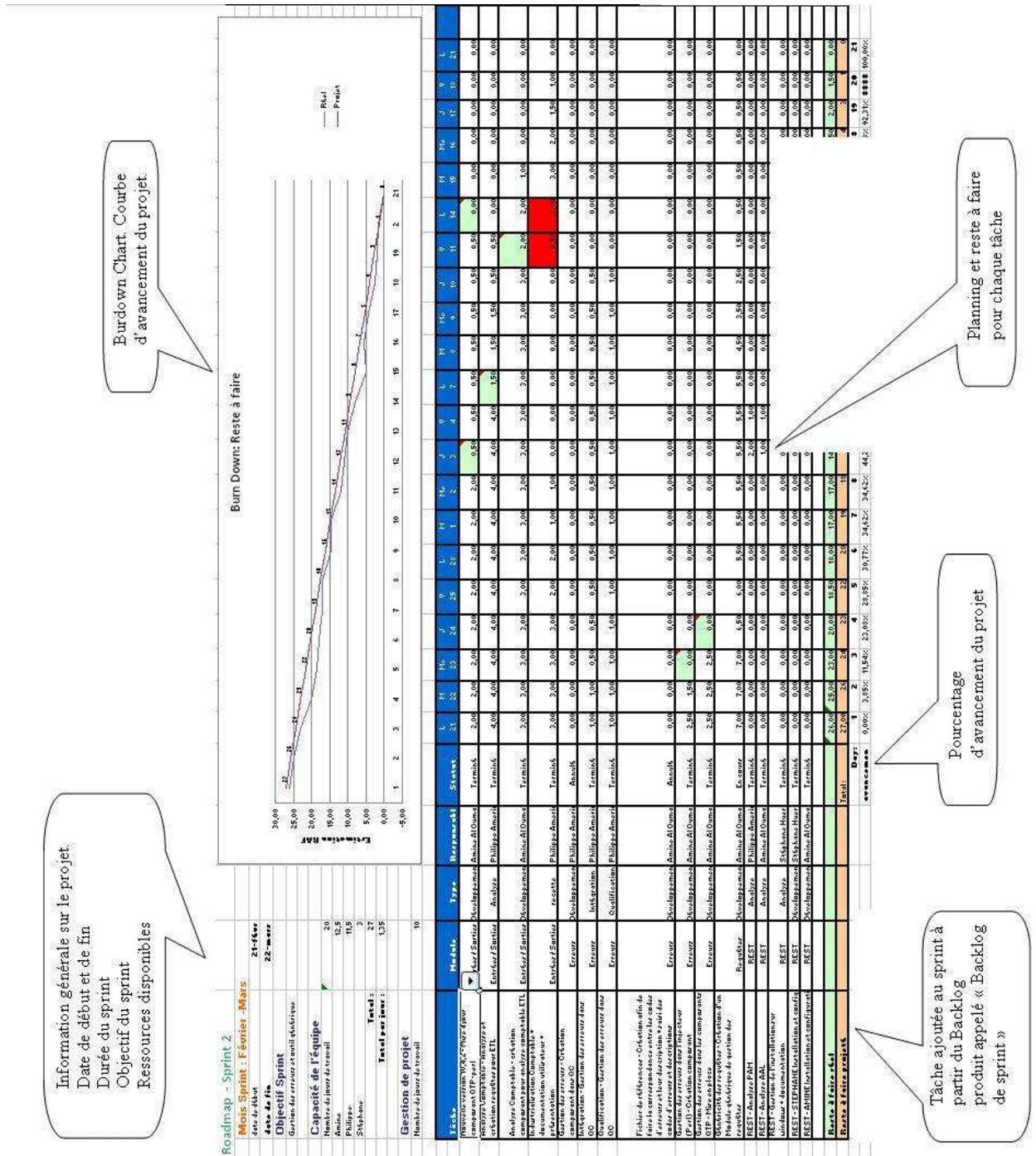


Figure 3-28 Illustration outil de suivi de projet

d **Burdown Chart**

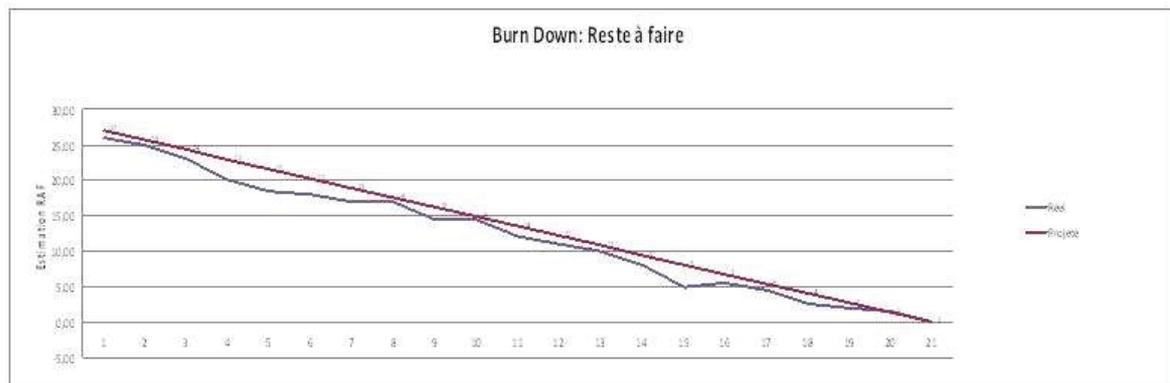


Figure 3-29 Illustration Burdown Chart

Dans la courbe ci-dessus nous pouvons remarquer que :

- sur l'axe des abscisses s'affiche le nombre de jours du Sprint ;
- sur l'axe des ordonnées s'affiche la charge de travail ;
- la droite en rouge représente la « projection » de la charge de travail qu'il reste à faire chaque jour avant la fin du Sprint;
- la courbe en bleu représente la charge de travail « réelle » qu'il reste à faire avant la fin du Sprint. Elle varie en fonction de l'avancée de l'équipe.

Si la courbe bleue passe au-dessus de la droite rouge, cela signifie que l'équipe de développement est surchargée de travail vis-à-vis de ce qu'elle est capable de faire. Ceci peut être dû au fait qu'une tâche ait pris plus de temps que prévu ou qu'une personne soit moins disponible que prévue (maladie par exemple). Dans ce cas, deux actions sont nécessaires :

- La première est de comprendre pourquoi la tâche a mis plus de temps que prévu. Cela peut être dû à :
 - une sous-estimation de la difficulté de la tâche,
 - une difficulté pour le développement à réaliser le travail,
 - une baisse de performance du développeur.

Dans tous les cas, une recherche et une compréhension des causes sont faites lors de la réunion quotidienne (Mêlée). S'il le faut, cela peut faire l'objet par l'équipe de développement d'une analyse plus approfondie du problème.

- La seconde action, effectuée par le ScrumMaster et en fonction des explications de l'équipe, sera de :
 - réévaluer cette tâche si cela est nécessaire,
 - ou d'annuler cette tâche au profit d'autres tâches,
 - et dans le cas où la charge de travail reste toujours supérieure à la capacité de l'équipe : supprimer les tâches les moins prioritaires afin de tenir la date de fin Sprint.

Si la courbe bleue reste en dessous de ligne rouge, cela signifie que l'équipe est en sous charge de travail. Le ScrumMaster peut donc rajouter des tâches du Backlog après avoir les avoir priorisés avec le Product Owner.

3.5.3 Vue d'ensemble du projet

Du fait que le projet n'ait pas encore l'importance qu'il aura par la suite, il a été convenu au départ avec la Direction que j'effectue la gestion du projet d'automatisation des tests sans abandonner les autres tâches annexes. Ci-dessous, la liste des tâches qui m'ont été allouées et leurs priorités.

N° tâche	Libellé de la tâche	Type de tâche	Priorité
1	Correction problèmes de production	ponctuelle	1
2	Contrôle processus ETL <i>Modern</i>	quotidienne	2
3	Réalisation de tests techniques	ponctuelle	2
4	Gestion du projet automatisation des tests	quotidienne	3

Ceci explique pourquoi, nous apercevons, dans le planning global ci-dessous, une longue période avant le démarrage du projet (Août – Décembre) durant laquelle nous avons effectué pendant une dizaine de jours les principales tâches de l'analyse préliminaire décrite plus haut dans ce document.

La période suivante, de janvier à juin 2011 est la période pendant laquelle les développements ont été effectués. L'étude détaillée a commencé en amont de la phase de réalisation, puis elle a continué parallèlement à cette phase.

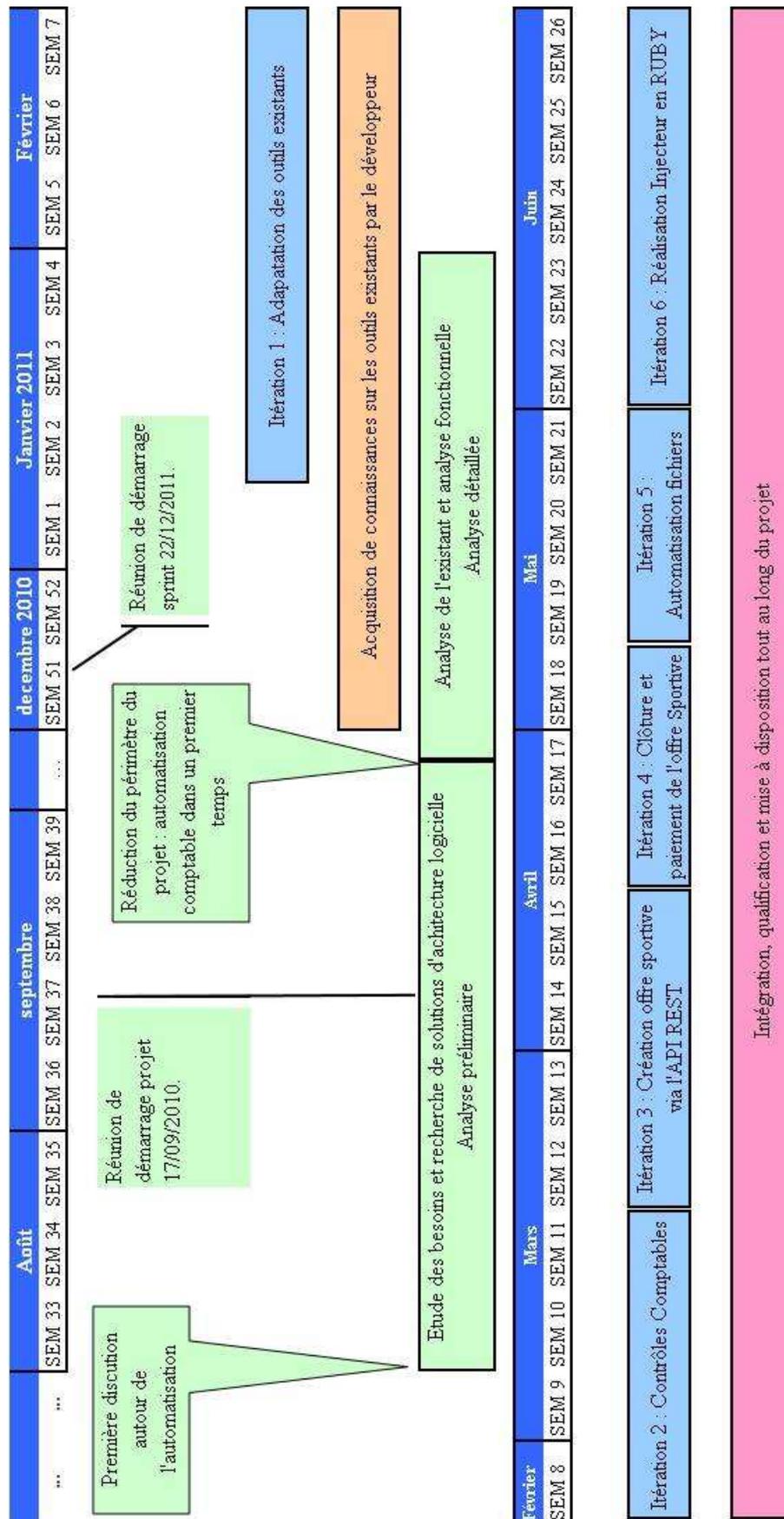


Figure 3-31 Illustration planning global du projet

a Résumé des étapes

Le projet peut se décomposer en 3 grandes étapes de développement avec en amont une période d'analyses préalables :

- La première étape correspondant à deux itérations :
 - La première concerne, la compréhension par l'équipe de développement des outils et l'adaptation des outils de création et de clôture de l'offre sportive développés en QTP, ainsi que l'adaptation de l'outil d'injection des prises de paris, développée en PERL.
 - La seconde concerne, la gestion des erreurs, ainsi que la création de deux modules de contrôle comptable sur les bases de données ABP, AGP_ETL et ETL : « CompareReporter » et « Injecteur_Accounting ».
- La deuxième étape cible le remplacement des deux modules QTP en Ruby. Elle se déroule en deux itérations :
 - La première, pour la création de l'offre sportive *via* l'interface API REST,
 - La seconde, pour la clôture de l'offre sportive *via* l'interface API REST.
- Durant la troisième et dernière étape du projet d'automatisation comptable, nous avons remplacé le module d'injection développé en PERL par un module développé en Ruby.

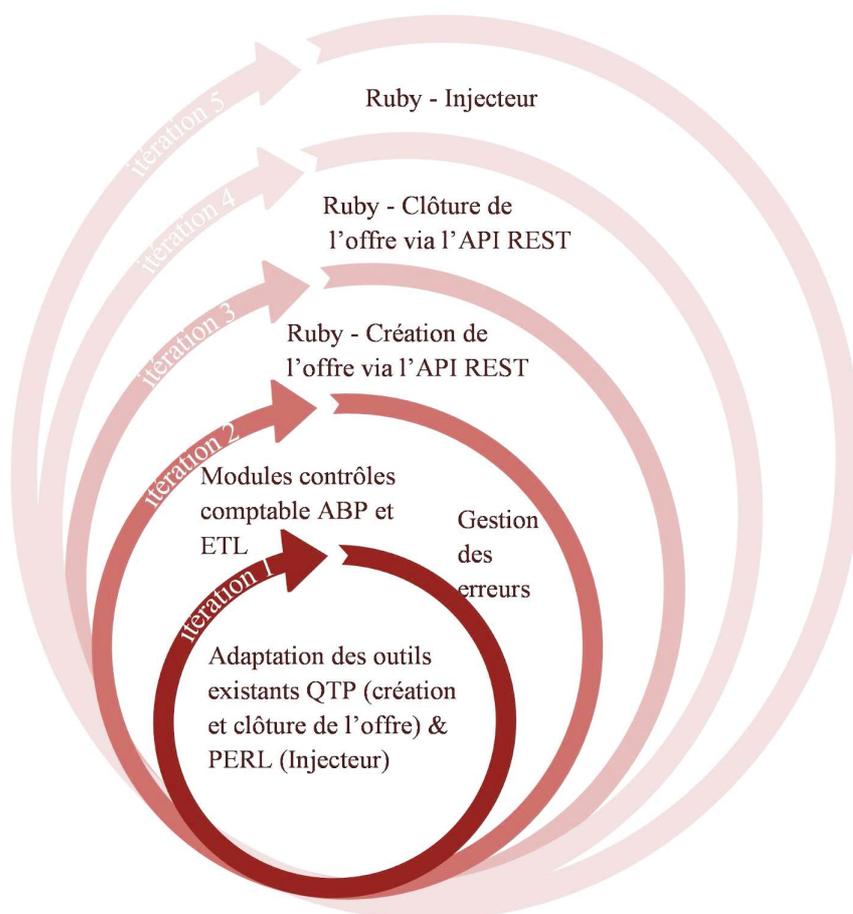


Figure 3-32 Schéma des itérations du projet d'automatisation comptable.

3.5.4 Première étape

a Objectifs

Cette première étape est la plus importante du projet. Son objectif est double : d'une part, de donner un retour rapide sur la faisabilité de l'automatisation comptable à partir des outils existants ; d'autre part, de mettre en place l'ordonnancement des différents modules.

Objectifs du premier Sprint : adapter les outils existants à l'automatisation et faire un retour sur les limitations découvertes.

Objectif du second Sprint : finaliser l'automatisation comptable en créant les modules nécessaires aux contrôles d'intégrité en base de données.

b Rôles et responsabilités

Chef de projet, architecte

Pour la première étape du projet, l'équipe d'automatisation se compose de deux personnes. Pour ma part, j'ai eu en charge la définition de l'architecture fonctionnelle et la gestion du projet ; ce qui m'a amené à échanger sur les choix d'architecture, la définition et la priorité des tâches avec le développeur, en l'occurrence M. Al Oumami. Celui-ci vient d'intégrer la Française Des Jeux en décembre 2010 et est depuis en charge du développement ou de l'adaptation des applications automatisées. Il a été choisi en raison de ses compétences en développement parce qu'il peut s'occuper du développement et de la maintenance des outils déjà créés en PERL et en VBScript sous QTP.

En janvier 2011, l'organisation du projet se présentait comme suit :

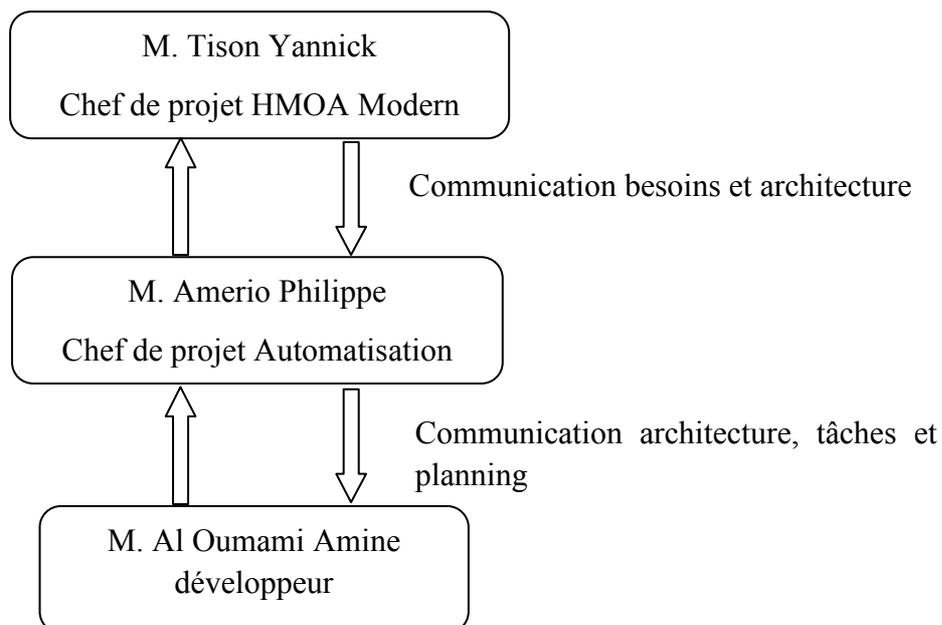


Figure 3-33 Schéma organigramme de l'équipe lors de la première étape du projet

c **Gestion de projet**

Durant cette étape, le projet respecte ces 3 notions essentielles de l'Agilité à savoir :

- satisfaire le client en livrant un produit fini au plus tôt et régulièrement, ce qui est la priorité du projet afin d'avoir un retour rapide sur la faisabilité du produit ;
- composer avec le changement : des changements d'architecture fonctionnelle et technique peuvent être nécessaires pour adapter les outils existants ;
- collaboration continue entre le métier et le développement : nécessité d'informer la Direction sur l'avancée et les éventuelles difficultés rencontrées durant le projet.

Réunion de planification de Sprint

Avant de pouvoir démarrer les premiers développements, il faut au préalable nous mettre d'accord sur ce qui sera fait. Cela est possible grâce aux réunions de démarrage de chaque Sprint. A ce stade, nous disposons des éléments fournis en amont par la phase préliminaire et en partie par la phase détaillée qui a démarré peu de temps auparavant. Le développeur a également pour objectif d'améliorer ses connaissances sur les outils déjà existants afin de commencer à proposer des solutions techniques. A l'issue de ce travail, les points suivants ont été abordés lors de cette réunion :

1. Gestion des Entrées/Sorties

Ce point est essentiel pour l'ordonnancement des tâches. En effet, il décrit comment les différents modules vont s'interconnecter entre eux. Il détermine comment la sortie d'un module doit être formatée pour pouvoir être ensuite utilisée en entrée par le module suivant.

2. Gestion des gagnants

Lorsqu'un joueur est payé par la Française de Jeux, et qu'il se connecte sur son compte joueur, un message apparaît pour lui préciser qu'il est gagnant. Le joueur doit ensuite confirmer manuellement en cliquant sur « Valider » avant de pouvoir revenir sur la page d'accueil du site ParionsWeb. Toutefois, le fonctionnement de l'injecteur, tel qu'il a été développé ne prend pas en compte cette page de validation et s'attendait à être directement dirigé vers la page d'accueil du site de ParionsWeb pour effectuer les prises de paris. Nous avons donc dû adapter l'outil pour que celui-ci prenne en compte cette page intermédiaire qui pouvait alors apparaître.

3. Gestion des erreurs

Pour plus de simplicité dans l'enchaînement logique des modules, des contrôles fonctionnels ont été mis en place pour n'autoriser l'enchaînement de deux modules qu'à la condition que le premier soit en succès.

Pour cela, nous avons dû faire remonter l'état du module et mettre en place une gestion des erreurs. En effet, si les modules QTP remontent les informations de succès ou d'échec directement à l'ordonnanceur QC, ce n'était pas le cas du module d'injection PERL. Une gestion des erreurs a dû être mise en place pour cela.

4. Gestion des documentations

Une mise au point sur les documentations techniques et fonctionnelles que l'on doit fournir à la Direction a été faite.

5. Passage de connaissances ancien développeur / nouveau développeur

Cette réunion a été l'occasion de faire un point sur les connaissances des outils sur lesquels le développeur allait devoir travailler. Ceci afin de prévoir un planning plus large dans le cas où sa compréhension ne serait pas complète.

A l'issue de cette réunion, un compte-rendu de réunion a été réalisé. Ci-dessous la liste des tâches de gestion effectuées :

Personne	Tâches	cadre
Philippe	Détailler précisément les entrées/sorties de chaque module. Mettre à jour les documentations	
Philippe	Lister les tâches et faire un RACI	
Philippe	Programmer une réunion pour estimer le temps de développement de chaque évolution	

Le RACI³⁴ a été fait, les tâches définies et allouées et le planning du Sprint défini.

« Priorisation » des développements

Une fois l'ensemble des tâches définies, quatre macro-tâches se distinguaient, priorisées de cette façon:

1. les développements axés autour des Entrées / Sorties,
2. le développement pour la gestion des gagnants,
3. le développement de la gestion des erreurs des modules PERL et QTP,
4. le développement des Modules pour les contrôles comptables,

La priorisation des tâches a été faite en fonction des risques liés à la réalisation du projet. Il est important de vérifier dans un premier temps que les outils existants puissent être automatisés; ensuite que l'automatisation ne soit pas bloquée par la gestion des gagnants³⁵; et enfin que l'ordonnancement de bout en bout des modules « création de l'offre », « injection des prises de paris », « clôture » et « paiement des Markets » soit possible. Il faut pour cela s'assurer que les erreurs soient correctement remontées et que le contrôle du succès ou de l'échec d'un module soit possible.

En plus des tâches de développement, des tâches d'intégration, de qualification et de rédaction de documentation sont prévues.

Outil de suivi de fichiers : subversion

Dès la première étape, nous avons demandé à avoir un outil pour le suivi des versions de fichiers. Nous l'avons utilisé, pour les fichiers de suivi de projet, les documentations ainsi que les développements effectués en PERL et QTP.

³⁴ **RACI** représente une matrice des responsabilités qui indique les rôles et les responsabilités des intervenants au sein de chaque processus et activité.

³⁵ Détail page précédente

d **Développements effectués**

Développements effectués lors du Sprint 1

- Les développements sont axés autour des Entrées/Sorties des modules de « création de l'offre » et de « l'injection » déjà existants afin de rendre possible l'automatisation.
- Le développement de la gestion des gagnants.

Développements effectués durant le Sprint 2

- Les développements qui concerne la gestion des erreurs pour les modules « d'injection » (PERL) et de « création de l'offre » (QTP),
- Le développement du module de contrôle comptable ABP,
- le développement d'un outil générique de requête (utilisé pour les contrôles comptables AGP_ETL et ETL).

e **Problèmes rencontrés**

Limitation de l'outil QTP

1. L'outil dépend de l'interface graphique :

QTP est un outil qui se base sur l'interface graphique. Les deux modules développés sous QTP effectuent des actions sur le back office. Ainsi, à chaque modification de l'interface du back office, le risque de voir les objets graphiques changés ou être simplement déplacés est important. A chaque livraison d'une nouvelle de version du site ParionsWeb, des modifications de l'interface sont constatées et provoquent des erreurs dans l'exécution des scripts QTP. Cela nécessite de mettre à jour, de rechercher les erreurs et de redéfinir les objets. Le temps de remise à niveau des scripts QTP vis-à-vis d'une nouvelle version ne peut se prévoir et est souvent trop long.

2. Le Java SWT n'est pas correctement pris en charge par le module d'extension "QTP Java Plugin" :

Cette extension est utilisée par tous les objets créés spécifiquement afin d'accéder à l'interface du back office ABP. La reconnaissance des objets graphiques Java n'est cependant pas fiable à 100%. D'une exécution à l'autre, sans que l'interface graphique ait été changée, certains objets Java ne sont plus reconnus.

3. QTP gère mal les accès aux ressources :

La mise en place de chemins relatifs n'est pas correctement prise en compte. Ce qui nous oblige, lors d'une installation sur une nouvelle machine, à redéfinir tous les chemins d'accès, à savoir :

- le chemin vers l' « object repository » : objets que QTP utilise pour effectuer les actions automatisées ;
- le chemin vers les « reusable actions » : actions qui définissent une suite d'opérations en particulier. Par exemple, pour se connecter au back office, on appelle une action particulière

qui n'a pour seul rôle de se connecter au back office, les autres opérations étant enregistrées dans d'autres « reusable actions » ;

- le chemin vers le fichier des paramètres d'entrée.

Si l'on avait dû installer l'un des deux modules QTP : création ou clôture de l'offre, sur une machine différente de celle du développeur, nous aurions dû compter une demi-journée de travail.

4. Nécessité d'une ressource ayant des connaissances QTP très pointues :

L'utilisation d'un outil spécifique comme QTP pose des problèmes pour la continuité du projet. L'équipe HMOA Multimédia Sport est obligée d'avoir, dans ses effectifs, une personne spécialisée en VBScript et QTP pour pouvoir maintenir les modules QTP.

Cela pose un problème pour la distribution des ressources humaines. En effet, aucune autre personne de l'équipe n'est en mesure d'intervenir sur ces modules. La gestion des ressources s'en trouve du coup restreinte.

5. L'industrialisation est difficile à mettre en place :

L'exécution des modules QTP nécessite au préalable l'installation du logiciel QTP. Nous pouvons difficilement créer un exécutable sous « Windows » et pas du tout sous linux.

6. Suivi de version difficile :

QTP utilise des fichiers binaires et il n'est pas possible d'effectuer un suivi de version sur un fichier binaire.

7. Limitation à « Windows » :

Bien que connu au départ du projet, cette limitation est à prendre en compte.

La reprise de l'existant est difficile et beaucoup de travail reste alors à faire.

1. L'architecture déjà mise en place n'est pas optimisée :

L'utilisation de deux projets différents : un pour les objets réutilisables et l'autre pour les actions du contrôle compliquent le débogage. En effet, il faut fermer un projet, ouvrir le second, le corriger, le sauvegarder, le fermer et ouvrir le premier projet.

D'autre part, il y a beaucoup de dépendance entre les fonctions : une mise à jour complète du code est obligatoire si nous voulons faire évoluer l'outil.

Le langage n'étant pas de l'objet, la mise en place d'une architecture correcte pour l'application est plus difficile et la maintenance aussi.

2. Des règles fonctionnelles liées aux cas d'utilisation sont mal ou non implémentées :

Des problèmes se sont manifestés lors de la création des « rankEvents »³⁶, les « opposants »³⁷ et la recherche sur les « Markets » n'est pas fiable. Par exemple, lorsque nous cherchons un « Market » avec une description « TOP 14 orange », l'outil s'arrête sur le premier « Market » trouvé commençant par « TOP 14 ». D'autre part, la création d'un « Market » dans un événement en particulier se base sur la description et non sur son identifiant.

3. Certaines règles de gestion liées aux back office ne sont pas implémentées :

- Lors de la création d'un événement dans le back office, le « Market » par défaut est créé automatiquement. Le module QTP de génération des « Markets » ne doit alors pas créer de « Market » si celui-ci est un « Market » par défaut puisqu'il est créé en même temps que l'évènement. Or, ce dernier se crée. Ce qui provoque une erreur lors de l'exécution de QTP.
- La cotation avec l'outil du back office permettant de gérer la cotation des Markets : « Risk Manager » tombe en erreur si le premier « Market » du même nom qu'il souhaite coter est déjà fermé. Dans le cas de l'automatisation, cela peut arriver car deux « Markets » de même nom peuvent être créés et cela cas n'est pas pris en charge.

Limitation liés à QC

L'exécution sur une machine distante, appartenant à autre utilisateur, d'un script depuis QC n'est pas possible. Cette fonctionnalité nécessite en effet l'ouverture de ports et de droits spécifiques qu'il faut demander à la sécurité. Cette fonctionnalité n'est toujours pas présente.

Initialement, il a été prévu d'avoir pour chaque jour un « TestSet » avec les Tests du jour associés. Mais les traitements du deuxième jour ne peuvent démarrer que si les traitements du premier jour se sont correctement déroulés. Seulement, QC n'ordonne que les tests à l'intérieur d'un « TestSet » et pas les « TestSets » entre eux. Il n'offre donc pas la possibilité de signifier à un « TestSet » que le « TestSet » précédent et donc l'ensemble de ces tests se sont correctement déroulés. C'est pour ces raisons que l'ensemble des tests prévus pour les quatre jours se sont retrouvés rassemblés dans un même « TestSet ».

³⁶ Les rankEvents sont des « Markets » particuliers.

³⁷ Pour chaque Market créés, des opposants sont utilisés. Par exemple, pour un match de tennis les opposants sont Nadal et Federer

Limitation du Script PERL

L'appel d'un script PERL se fait au travers d'un Script VBScript, puis au travers d'un fichier de commandes MS-DOS (.bat). Nous ne pouvons donc pas communiquer directement à QC le log des erreurs et le code « sortie » du script PERL. Les logs s'enregistrent alors dans un fichier de log au format texte et le résultat du test s'affiche sur la sortie standard. Cette sortie standard est récupérée et analysée par le script VBScript, coté QC, qui a lancé l'exécution du script PERL. Une notation spécifique « ERR : » a alors été mise en place dans le cas où les scripts PERL terminent en erreur. Lorsque le script VBScript parcourt la sortie standard et lit « ERR : », ce dernier définit le test avec un statut « Fail ». Pour correctement contrôler le résultat du script, la solution a été d'avoir un code de sortie de script sur lequel QC peut se baser pour mettre l'état du test à « fail » ou à « success ». Pour afficher les erreurs, la solution aurait été de faire en sorte que le script VBScript lise le fichier de log généré par les scripts PERL et l'affiche dans le résultat de QC.

Intégration, qualification et mise à disposition des modules

Pour cette première étape, j'ai été en charge de l'intégration et des tests d'intégration dans QC des modules QTP et PERL. La phase d'intégration s'est correctement déroulée mais la phase de qualification a mis en évidence une partie des erreurs précédemment décrites.

f Conclusion du Sprint

La communication auprès du Product Owner des problèmes rencontrés avec cet outil s'est faite dès la phase de qualification des outils. Certaines erreurs n'ont pas pu en effet être découvertes avant cette phase.

Néanmoins, l'adaptation des outils pour les besoins de l'automatisation s'est révélée un succès. En effet, aucune limitation de la part des modules n'a empêché leurs ordonnancements. L'architecture mise en place a parfaitement convenu à la communication entre modules.

Les limitations techniques ont donc été liées à QTP, ainsi qu'à une mauvaise conception initiale de ces outils. L'utilisation de l'outil QTP n'a, au départ, pas la même utilité. Le niveau de performance de l'outil est beaucoup moins important et l'utilisateur de départ ne doit pas être dérangé par le fait de devoir relancer l'application si cette dernière tombe en erreur.

D'autre part, cette architecture a permis de ne pas perdre les efforts entrepris pour adapter le module d'authentification et d'injection des prises de jeux ainsi que les modules de contrôle comptable. Même si une partie des modules apparaissent à la suite de cette itération très peu exploitables, d'autres l'ont été complètement.

Après discussion avec le responsable HMOA Multimédia Sport et à partir des problèmes remontés, nous avons décidé qu'il n'était pas possible, vu l'importance des limitations, d'utiliser QTP pour nos besoins. Nous avons donc été contraints de remplacer les deux modules QTP par deux autres ne présentant pas les inconvénients de QTP.

3.5.5 Deuxième étape

a Objectifs

Remplacer les deux modules création et clôture des « Markets » développés en QTP par deux autres modules développés dans un nouveau langage objet : Ruby³⁸ ; et qui se serviront d'une nouvelle application mise en place par LVS : l'API³⁹ REST⁴⁰. Une des conditions de réussite de cette étape est que les deux nouveaux modules doivent au moins faire ce que faisaient déjà les modules QTP.

b Solution de remplacement

Solution de remplacement de QTP

Une réunion avec LVS a eu lieu début mars 2011. Cette réunion a pour ordre du jour la présentation par LVS de l'outil qu'ils ont développé et utilisé afin d'effectuer des tests simplifiés de création d'offre, d'injection, de clôture et de paiement automatisé. LVS a en effet développé une architecture de type REST qui permet d'accéder à des fonctions sur un serveur distant *via* le protocole http.

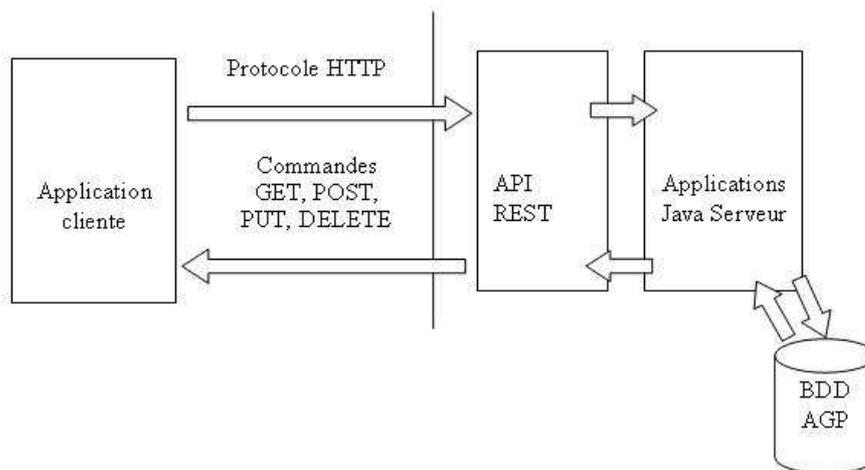


Figure 3-34 Schéma de l'architecture REST mise en place par LVS

A l'issue de cette réunion, nous avons estimé que l'utilisation de l'API REST développé par LVS répondrait aux besoins de l'automatisation. J'ai donc dressé une liste des avantages et inconvénients de l'API REST.

³⁸ **Ruby** est un langage de programmation libre. Il est interprété, orienté objet.

³⁹ **Application Programming Interface (API)** est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres.

⁴⁰ **REpresentational State Transfer (REST)** est un style d'architecture basé sur l'appel d'une ressource(URI) avec laquelle le programme va communiquer grâce au protocole HTTP. Le Web est construit sur ce style d'architecture.

Avantages	Inconvénients
Accès aux données sans passer par l'interface graphique.	Nous perdons les règles métiers liées au back office
Certaines règles métiers sont tout de même incluses dans l'API.	L'API REST n'est pas un service. Nous devons donc redéfinir dans le code les règles métiers manquantes.
La mise en place des règles métiers oblige l'équipe à connaître davantage le fonctionnement de l'application.	Aucun engagement écrit de la part de LVS pour le support et la maintenance de cette application.
Le directeur de LVS s'est engagé à faire évoluer l'application.	L'automatisation est dépendante de l'API REST.
Une personne de chez LVS est dédiée à cette application et est disponible pour répondre à nos questions et corriger les erreurs.	L'API REST utilise des services d'ABP comme le « forceFeed ». Nous ne pouvons pas utiliser un module si les services appelés par ce module fonctionnent mal.
L'accès au serveur se fait facilement <i>via</i> le protocole http et les commandes associées. La compréhension de ce protocole est facile.	
Une documentation est fournie.	

Les deux risques liés à l'API REST et menaçant la réussite du projet sont :

- le manque de réactivité de LVS vis-à-vis des demandes d'évolution et de correction puisqu'aucun engagement écrit n'avait été fait. Mais, nous avons pu constater que cela n'a pas été le cas.
- La dépendance de l'automatisation vis-à-vis de l'API REST. Cependant, à part quelques bogues rencontrés, de manière générale l'API REST a été stable et a répondu correctement. D'autre part, l'utilisation de l'API REST, a eu l'avantage de nous fournir quelques règles métiers. Sans cela, nous aurions été obligés de redéfinir complètement ces règles ; par exemple, dans le cas où nous aurions choisi d'accéder et d'effectuer des modifications en accédant directement *via* des requêtes SQL⁴¹ à la base de données. L'autre avantage a été de pouvoir demander à LVS des évolutions pour nos propres besoins et notamment la mise en place de services.

Choix du langage

Le langage de développement utilisé par LVS pour automatiser leurs tests a été Ruby. Notre choix s'est également porté sur ce langage pour plusieurs raisons :

1. c'est un langage Objet,
2. ce langage est déjà utilisé par LVS pour le Tiers 2 d'ABP et pour l'automatisation, et en cas d'incompréhension, nous pouvions demander de l'aide à LVS,
3. c'est un langage facile à comprendre et à mettre en œuvre,

⁴¹ *Structured Query Language* est un langage informatique normalisé servant à effectuer des opérations sur des bases de données

4. la compréhension du langage Ruby permet de mieux comprendre le fonctionnement du Tiers 2 d'ABP, un langage commun qui permet la redistribution des ressources,
5. les équipes acquièrent des connaissances sur des nouvelles technologies.

c ***Rôles et responsabilités***

Migration vers le rôle de ScrumMaster

Lors de la seconde étape du projet, une troisième personne est venue renforcer l'équipe. Il s'agit d'un consultant externe de la Française des Jeux, qui fait déjà partie de l'équipe HMOA et qui a souhaité rejoindre l'équipe d'automatisation. Celui-ci dispose alors des compétences organisationnelles en gestion de projet Agile et plus particulièrement en méthodologie Scrum. Il a également des compétences en architecture de développement Objets. Grâce à ses compétences, il a pu jouer deux rôles principaux :

- Développeur senior et architecte technique des modules qui ont été développés en objets,
- Coach ScrumMaster.

L'équipe de développement Scrum s'est constituée de deux personnes: un développeur senior et un développeur junior.

Le développeur senior a des connaissances en bonne pratique de développement objet, très profitable pour mener à bien le projet de refonte des modules QTP en modules développés en Ruby. Le développeur junior a déjà connaissance du fonctionnement des deux modules de création et de clôture de l'offre.

Après avoir été rassuré sur la capacité de l'équipe à définir une architecture logicielle cohérente, à analyser et remonter les problèmes rencontrés, j'ai décidé de laisser progressivement l'équipe de développement s'autogérer. Auparavant plus impliqué dans la définition de l'architecture et des solutions techniques à mettre en place, mon rôle a évolué et s'est rapproché de celui de ScrumMaster : je tenais un rôle d'accompagnateur et non plus un rôle directif de chef de projet.

Les rôles du ScrumMaster :

- accompagner l'équipe dans le développement des modules en supprimant les obstacles,
- être l'interface entre les besoins de la Direction et l'équipe de développement, mais également entre le fournisseur de l'application REST et l'équipe de développement,
- définir les priorités,
- définir et estimer avec l'équipe les tâches de développement,
- définir le planning des Sprints,
- réaliser une revue de Sprint en fin de Sprint,
- établir les rapports d'avancement de projet.

d **Gestion de projet**

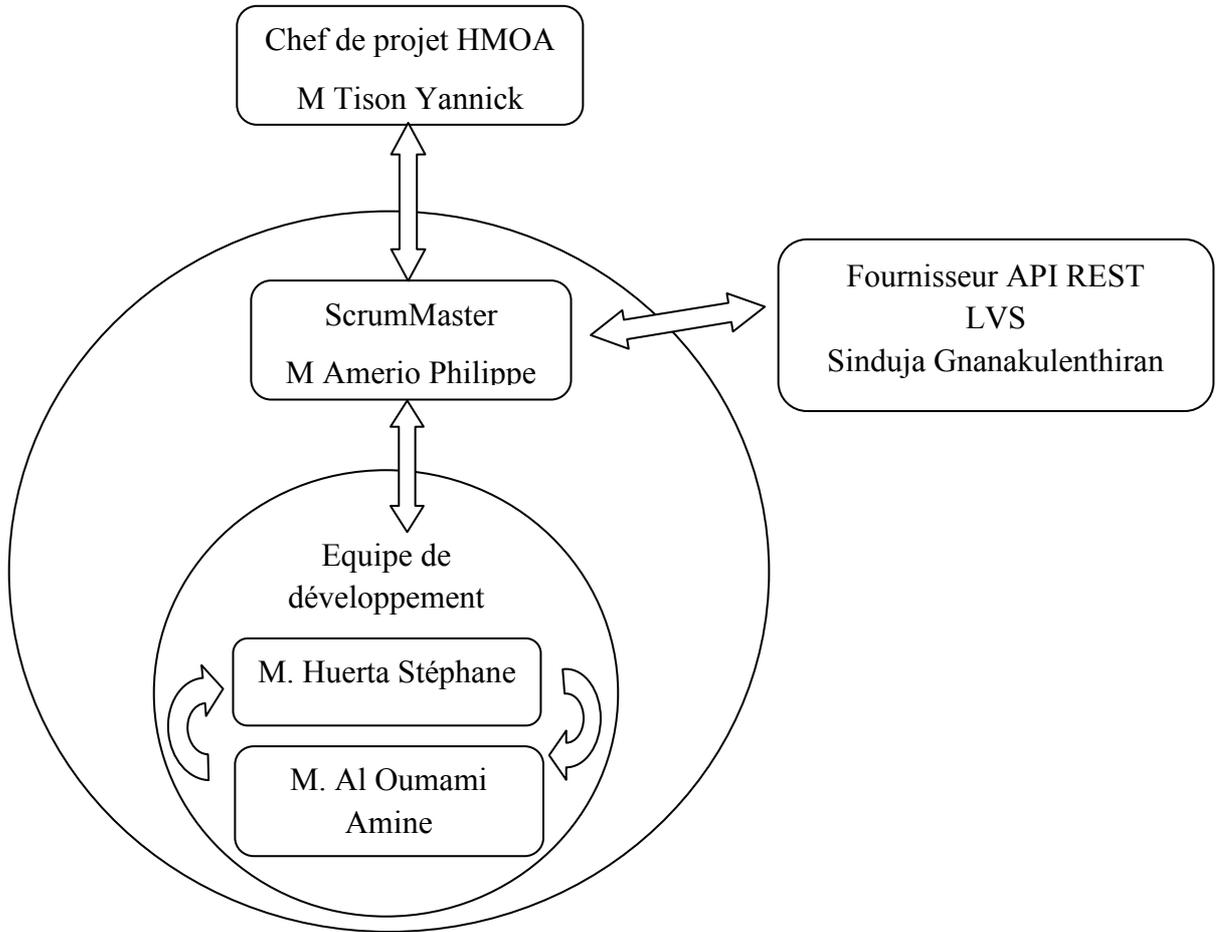


Figure 3-35 Schéma de l'organisation de l'équipe lors de la deuxième étape du projet

Outil de suivi de bogue LVS : « bug reviews »

Une partie de mon travail était de suivre les anomalies et de les remonter à LVS afin qu'ils les corrigent. Même si la prise en charge par LVS de la correction des bogues n'a pas été officielle, Sinduja a corrigé les dysfonctionnements que nous lui remontions et a intégré les corrections aux nouvelles livraisons d'ABP. Le Bug review a été un tableau de référencement des bogues que nous avons mis en place pour le suivi. Il était intégré au fichier du planning.

ID	bugs	Module	Comments	date	STATUS	Release	Priority
1	problem when we try to create a rank event	REST	this problem is fix by LVS	01/04/2011	waiting delivery	11.1.2	1

e **Développements effectués**

Comme expliqué précédemment, cette étape a compris deux Sprints. Le premier renvoie à la création de l'offre sportive et le second, à la clôture de l'offre sportive.

Cette étape a été l'occasion pour l'équipe de développement d'apprendre le langage Ruby et l'architecture REST. Ils ont construit une architecture objet en tenant compte du fait que les modules utilisés doivent pouvoir être réutilisables par la suite. L'équipe s'est servie de diverses pratiques d'architecture logicielle appelée « design pattern » pour développer la création et la clôture de l'offre sportive. Elle s'est par exemple servie du design pattern *Factory*⁴².

Ci-dessous, le schéma représentant l'architecture logicielle mise en place.

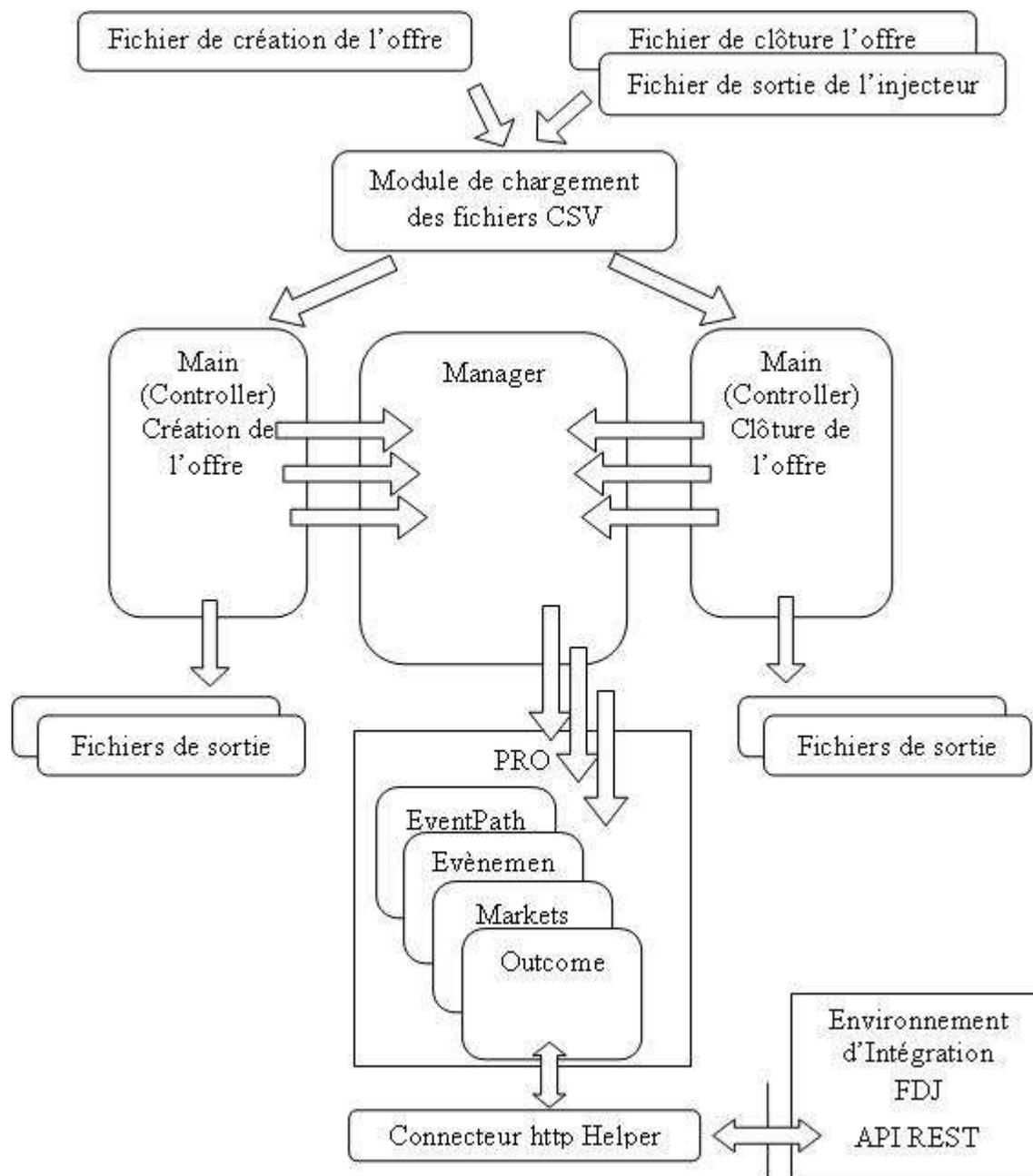


Figure 3-36 Schéma simplifié de l'architecture logicielle

⁴² En français *fabrique*, utilisé en développement objet pour regrouper des objets ayant les mêmes particularités

Environnement de développement

L'environnement de développement et d'intégration des nouveaux modules développés en Ruby, est « Windows ». Néanmoins, Ruby peut être facilement exportable sous linux et nous avons veillé à utiliser d'une part, la même version de Ruby que celle installée sur les environnements d'intégration à savoir la 1.9.2, et d'autre part, des bibliothèques compatibles avec un environnement Linux.

Afin de réaliser le projet, l'équipe s'est dotée de *NetBeans* comme environnement de développement. Cet EDI⁴³ est gratuit, s'installe et s'exécute en mode « standalone »⁴⁴. D'autre part, il prend en charge le langage Ruby. Il est muni d'un débogueur et permet également la recherche et le téléchargement de bibliothèques Ruby appelées « gem ».

f **Problèmes rencontrés**

Liés aux développements

Durant cette étape, les problèmes rencontrés par l'équipe de développement ont été liés au manque de connaissances sur l'API REST, sur le langage Ruby et sur des règles métiers de l'application ABP. Ces problèmes ont été surmontés au fur et à mesure de l'avancée du projet.

Liés à l'homme

Durant les développements, deux problèmes importants sont apparus. Ils sont dus au fait qu'un des développeurs n'a pas pour habitude de travailler en équipe et manque de compétences en développement.

Le manque de communication de ce dernier, notamment pour résoudre un problème qu'il ne comprend pas complètement et sur lequel il peut rester bloqué ; ses difficultés à suivre une consigne, nous a fait perdre du temps sur le développement. L'exemple le plus probant est lorsque, bien que d'accord sur l'architecture à mettre en place, ce dernier ne l'a pas du tout suivie.

Le manque de connaissances dans les bonnes pratiques de développement a nécessité une reprise de son code afin de corriger les bogues qui en découlaient.

Ces deux difficultés ont été surmontées assez rapidement par plusieurs actions de notre part.

Premièrement, afin de corriger le problème lié aux bonnes pratiques de codage, nous avons effectué :

- une revue du code par le développeur senior qui, connaissant les bonnes pratiques de développement a su corriger rapidement les bogues et mettre à jour le code ;
- nous avons encouragé le développeur concerné à utiliser des bonnes pratiques de développement en lui expliquant et lui montrant les avantages. Le code serait plus lisible, les erreurs moins fréquentes, et la maintenance plus facile ;
- nous avons fait en sorte que les deux développeurs travaillent la plupart du temps sur un même module afin que chacun est rapidement une idée du code de l'autre ;

⁴³ Environnement de Développement Intégré est un programme regroupant un ensemble d'outils pour le développement de logiciels

⁴⁴ Historiquement, standalone désigne un logiciel capable de fonctionner sans faire appel à des ressources du Système d'exploitation.

- nous avons alloué une tâche de mise à jour du code pour chaque Sprint généralement égale à 2 ou 3 jours pour, dans le même temps, améliorer l'architecture logicielle et pousser les développeurs à « nettoyer » le code.

Pour pousser l'équipe à mieux communiquer, plusieurs actions ont été entreprises.

Nous avons mis en place au début de la troisième étape, une réunion quotidienne (Mêlée dans Scrum) afin d'améliorer la communication au sein de l'équipe. Cette réunion a grandement amélioré la communication dans le sens où celle-ci oblige le développeur à :

- énoncer le travail effectué la veille ;
- expliquer les difficultés rencontrées ;
- dire sur quoi il compte travailler durant la journée.

Cette réunion a l'avantage de voir rapidement si un des développeurs a compris sur quoi il doit travailler, à l'inciter à parler de ses difficultés et à éviter que quelqu'un ne perde du temps inutilement sur un problème. Mais elle oblige aussi les développeurs à faire un effort d'explication sur le travail déjà effectué et à venir ; le tout dans une démarche de communication constructive.

Le ScrumMaster de son côté peut contrôler que les développeurs n'effectuent pas d'autres tâches que celles prévues dans le Backlog du Sprint. Il est au courant des problèmes en cours et effectue des actions correctrices si nécessaire.

Liés à limitation de l'API REST

Certains bogues de l'application REST, ne nous ont pas permis de reproduire à l'identique les fonctionnalités des modules développés en QTP. C'est le cas de la création des rankEvents et du paiement.

Pour les rankEvents, le problème a été remonté à LVS et une correction a été apportée par Sinduja et livrée en même temps que la version suivante d'ABP. A ce jour, la création de l'offre des rankEvents est opérationnelle.

Pour le paiement, le problème a résidé dans le fait que l'API REST ne fournissait pas la possibilité de payer un « Market » ou un événement. Nous n'avons pas souhaité redéfinir les règles métier du paiement des « Markets » ou des événements qui étaient compliqués. Une demande d'évolution officielle a été faite pour que LVS fournisse un service identique à ce qui a été réalisé pour le paiement des « Markets » dans le back office. Cette demande est remontée jusqu'au Chef de Projet de Modern afin que LVS inclut cette fonctionnalité dans une prochaine livraison (un extrait de cette demande se trouve dans l'annexe : « Demande d'évolution de l'API REST, rajout de la fonctionnalité de paiement »).

g Intégration, qualification, mise en application et maintenance

Pour chaque module développé, les développeurs ont effectué des tests unitaires avant de faire un « commits »⁴⁵ de leur travail sur SVN.

Pour chaque module, nous avons décidé de créer un exécutable « Windows ». D'une part, parce que l'utilisation des modules se font sur cet environnement et d'autre part, pour des raisons de simplicité lors du déploiement et de leur utilisation. La création de l'exécutable s'est appuyé sur un « gem »

⁴⁵ Les « commits », ou publications des modifications sont atomiques. Un serveur Subversion utilise de façon sous-jacente une base de données capable de gérer les transactions atomiques.

Ruby nommé *ocra* qui est appelé depuis un *rakefile*⁴⁶. Cet exécutable est ensuite mis à disposition dans un répertoire pour être partagé avec les personnes susceptibles d'avoir besoin de celui-ci afin d'automatiser leurs tests ou une partie de leurs tests.

Étant donné que l'équipe de développement était réduite à deux personnes, nous avons établi un processus de livraison simple :

- le développeur teste de façon unitaire son programme,
- une fois validée, il fait un commit des sources de son programme ;
- je récupère, les sources de ce programme sur mon poste de travail ;
- je qualifie le programme,
- j'archive l'ancien exécutable et génère le nouveau,
- je le mets à disposition des utilisateurs dans un répertoire partagé.

Ce processus simplifié de livraison convient à la taille de notre projet. En effet, si des erreurs remontent durant la qualification ou durant l'utilisation d'un module, la correction et la livraison du correctif peuvent être rapides.

Concernant la maintenance des modules, j'ai été la personne intermédiaire en charge de vérifier que les erreurs remontées par les utilisateurs étaient bien dues à un problème de programmation. Si c'était le cas, nous en discutons avec le développeur. L'avantage est ici de détourner le moins possible le développeur de son objectif.

h Conclusion du Sprint

L'utilisation de l'API REST s'est avérée correcte car l'utilisation d'un modèle transactionnelle évitait les problèmes rencontrés précédemment avec QTP.

Le risque de voir LVS négliger la correction des anomalies liées à l'API a été écarté. Les différents problèmes ont été corrigés ou sont en cours de correction. Les demandes d'évolution ont été officialisées afin que l'on ait une garantie sur leurs dates de livraison ainsi que sur la conformité avec les besoins que nous avons exprimés.

Le choix du langage Ruby a été judicieux. Ce dernier présente en effet non seulement les avantages cités précédemment, mais il répond aussi aux contraintes techniques de l'automatisation ; à savoir la création de module indépendant, qui puisse être automatisé pour la création et la clôture de l'offre.

Lors de cette étape, nous avons pu remplacer avec succès les modules existants par de nouveaux modules sans impacter les autres modules tels que l'injecteur ou les contrôles comptables.

⁴⁶ Les *rakefiles* (équivalents des *makefiles*) sont écrits en Ruby et spécifient comment construire automatiquement des fichiers souvent exécutables, ou des bibliothèques à partir d'éléments de base tels que du code source.

3.5.6 Troisième étape

a Objectifs

Les objectifs de cette troisième étape sont :

- d'automatiser la génération des fichiers XML Betradar,
- de remplacer l'injecteur PERL par un injecteur Ruby dans le but de n'avoir plus qu'un seul langage pour les trois principaux modules d'automatisation, afin de faciliter la maintenance et les évolutions.

Dans le cadre de ce mémoire, le périmètre se limite à l'automatisation du test comptable. C'est pour ces raisons que l'automatisation des fichiers XML de Betradar, qui n'est pas une fonctionnalité liée au test comptable, n'est pas détaillée dans le présent rapport.

Cependant, nous pouvons faire une remarque toute particulière sur le fait que pour produire les fichiers XML, nous avons réutilisé certains composants ainsi que les PORO⁴⁷ afin de garder la logique fonctionnelle mise en place lors de la création et la clôture de l'offre. C'est le cas par exemple du composant de chargement des fichiers aux formats « csv ».

Cela prouve ainsi que l'architecture répond correctement aux attentes et facilite la réutilisation des composants.

b Rôles et responsabilités

M. Tison a déjà effectué des tâches liées au rôle de Product Owner, comme :

- Représenter le client/utilisateur auprès de l'équipe,
- Définir et faire partager la vision du produit,
- Communiquer en externe sur l'avancement,
- Accepter ou refuser le travail réalisé,
- Être disponible sur demande de l'équipe,

D'autre part, il a manifesté un intérêt pour être davantage impliqué dans le projet. C'est donc tout naturellement qu'il a endossé le rôle de Product Owner.

Désormais, en plus des tâches qu'il effectue, il doit :

- Créer et maintenir le Product Backlog,
- Prioriser le Product Backlog,
- Participer aux meetings Scrum de l'itération,
- Ne rien changer durant l'itération.

⁴⁷ **Plain Old Ruby Object**, sont des objets basiques ne possédant aucune logique métier. La logique métier étant mise en place dans des objets appelés *managers*. L'avantage est de pouvoir réutiliser ces objets avec en mettant une logique métier différente à chaque fois.

La participation de M. Tison aux mêlées quotidiennes a grandement simplifié la communication. En effet, il était rapidement au courant des problèmes dans les développements et connaissait « au jour le jour » l'avancée du projet. Le planning et les mêlées ont suffi pour communiquer sur le projet.

Les tâches de création, de priorisation et de maintenance du Product Backlog ont été, à la différence d'une véritable gestion de projet Scrum, faites en collaboration avec le ScrumMaster. Avant chaque création d'items du Backlog de produit, le ScrumMaster a consulté l'équipe de développement afin d'établir une liste des fonctionnalités à développer en fonction des besoins exprimés par le Product Owner. Il a ensuite fait valider le Backlog de produit et la priorité de chaque tâche.

Composition de l'équipe durant la troisième étape

- 1 ScrumMaster (Philippe Amerio)
- 2 développeurs (Stéphane Huerta et Amine Al Oumami)
- 1 Product Owner (Yannick Tison)
- 1 coach Scrum (Stéphane Huerta)

c ***Gestion de projet***

La gestion de projet n'a pas changé durant cette étape. Elle a toujours été de type Scrum. Mais elle s'est améliorée grâce notamment à la mise en place de la réunion quotidienne et à l'attribution du rôle de Product Owner à M. Tison.

En effet, pour toutes les raisons citées précédemment, nous avons décidé de mettre en place une réunion quotidienne au début du Sprint 5 et, par la suite, il n'a plus été question de revenir en arrière au vu des avantages procurés par cette réunion.

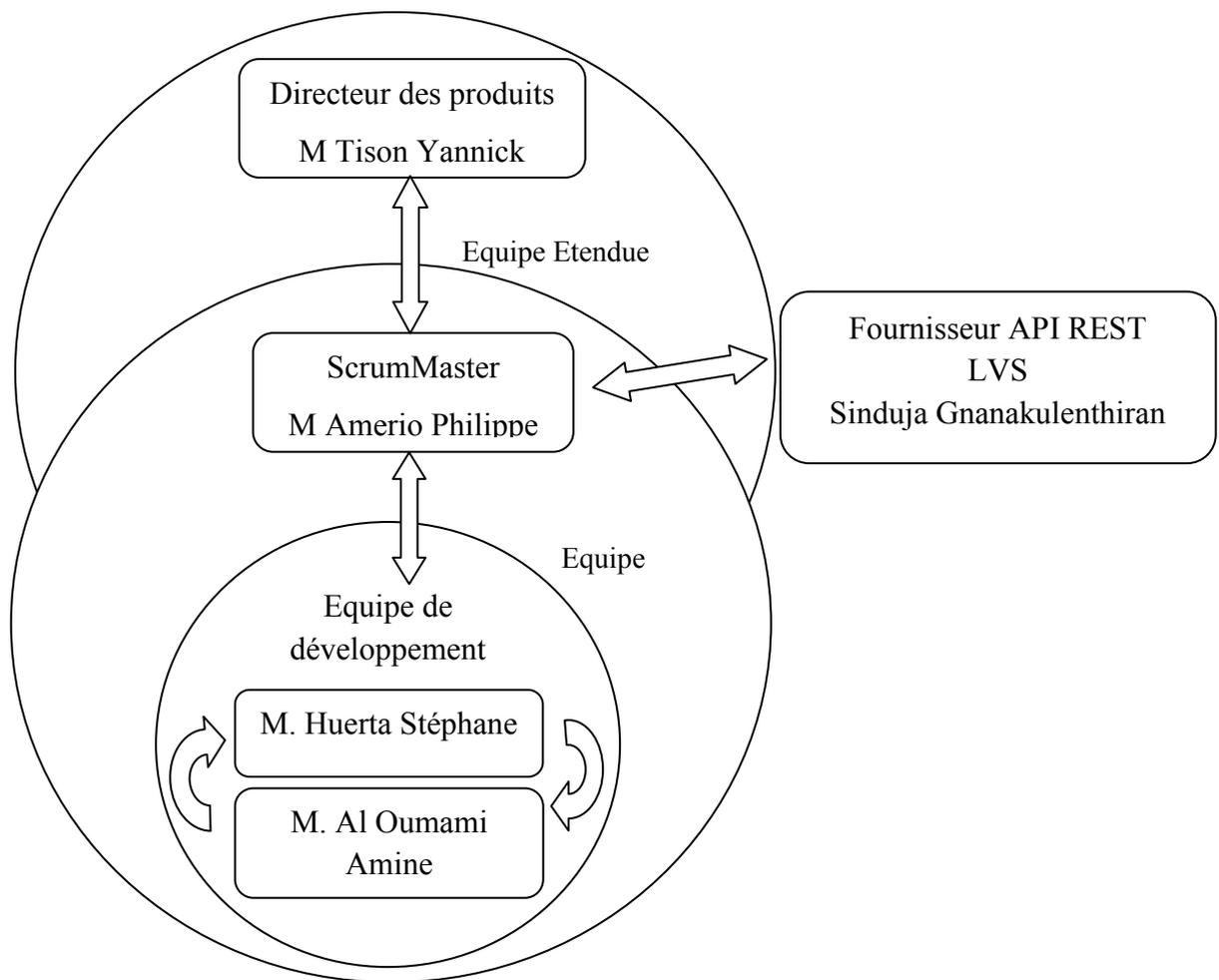


Figure 3-37 Schéma de l'organisation du projet, de type Scrum, mise en place

d ***Intégration, qualification, mise en application et maintenance***

L'intégration, la qualification, la mise en application et la maintenance est identique à l'étape précédente.

4 Mise en Application

4.1 Qualification Comptable

Au mois de juin 2011, la qualification comptable n'était pas entièrement automatisée. Les fichiers de configuration ont été créés et l'enchaînement des modules s'est correctement fait via Quality Center. Néanmoins, les prérequis nécessaires, avant chaque test automatisé ainsi que le contrôle des traitements comptables ont dû se faire manuellement. La personne en charge de la qualification a dû en amont :

- s'assurer que l'environnement soit fonctionnel,
- s'assurer aussi que les joueurs :
 - soient présents et encore actifs sur l'environnement utilisé,
 - aient une région définie qui correspondait à ce que l'on attendait pour la comptabilité,
 - disposent d'assez d'argent sur leur compte pour pouvoir faire toutes les prises de paris du test comptable,
 - aient des limites de prises de paris assez hautes pour effectuer des prises de paris sans être bloqué.
- s'assurer enfin que l'accès à la base de données soit opérationnel.

Une fois ces actions manuelles effectuées, le lancement de l'automatisation a pu commencer.

Chaque matin, avant que l'automatisation ne se lance, il a fallu contrôler que les traitements ETL se soient correctement passés.

Le schéma ci-dessous reprend le processus de qualification comptable définie lors de l'étude préliminaire, en précisant quelles sont les actions qui ont été automatisées.

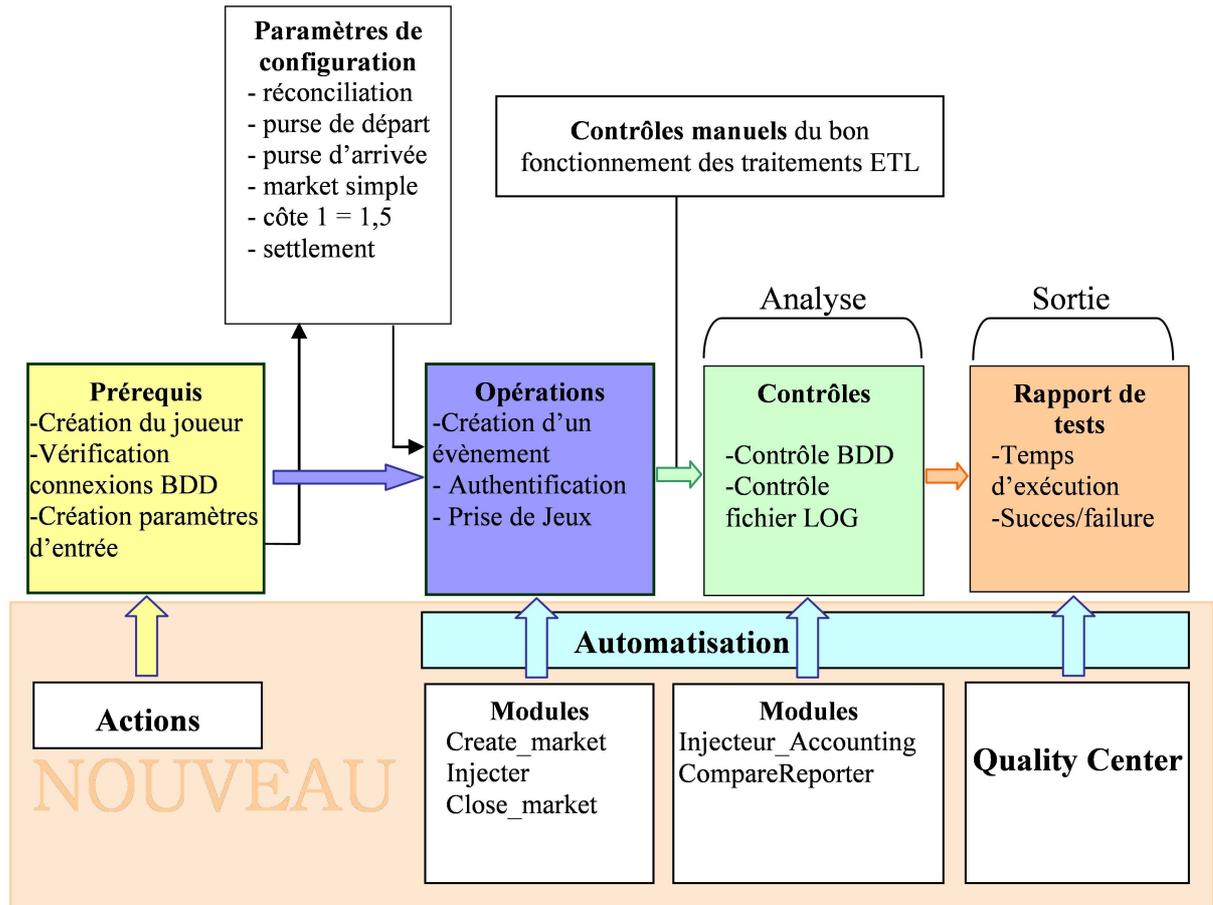
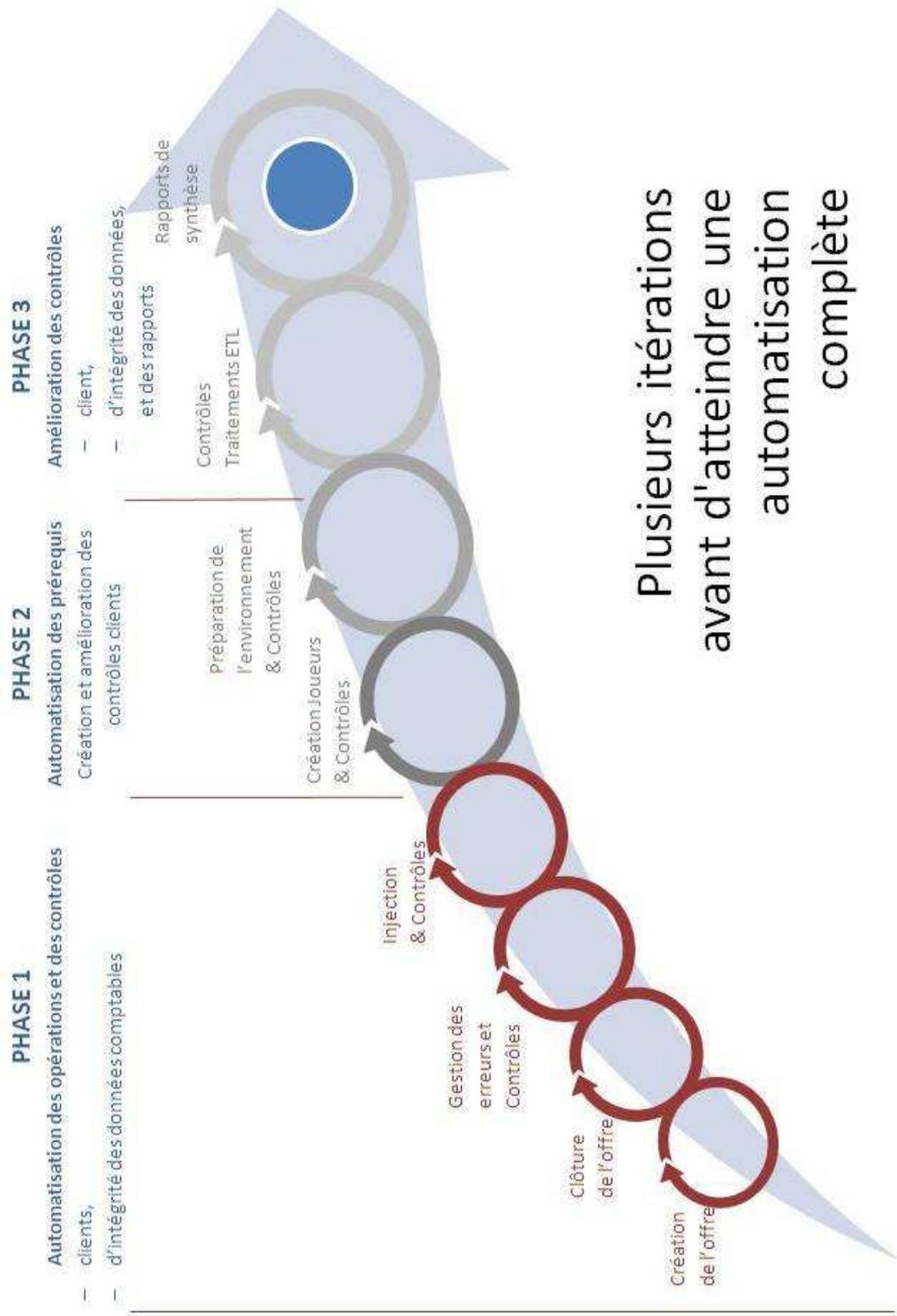


Figure 4-1 Schéma du processus d'automatisation comptable et ses actions automatisées

Pour automatiser l'ensemble du processus comptable, nous avons besoin, après le mois de juin, d'autres itérations ayant pour objectif l'automatisation des actions encore manuelles du processus. Le schéma ci-dessous montre les itérations nécessaires à l'automatisation complète du processus comptable.



Plusieurs itérations
avant d'atteindre une
automatisation
complète

Figure 4-2 Schéma présentant les itérations nécessaires à l'automatisation comptable

L'impossibilité de pouvoir automatiser de bout en bout le processus comptable a fait qu'à chaque campagne de tests, la personne en charge des tests a été obligée d'intervenir pour effectuer a minima les contrôles des traitements ETL. De juin 2011 jusqu'à aujourd'hui, le jeu de données comptables a été modifié à plusieurs reprises et, par manque de temps, les utilisateurs n'ont pas reporté ces modifications afin de lancer de nouveaux tests automatisés comptables. L'exécution manuelle de ce test est apparue aussi rapide que l'exécution semi-automatisée. On comprend, de ce fait, que la première condition pour que l'on utilise l'automatisation comptable réside dans la stabilité du jeu de données en entrée afin d'éviter que les utilisateurs aient à le modifier souvent. La seconde condition est l'automatisation complète du processus afin d'éviter que les utilisateurs n'aient à intervenir manuellement et que le gain de temps entre les actions manuelles et les actions automatisées soit réel.

4.1.1 Inconvénients de l'automatisation comptable

Il faut se poser la question du réel intérêt d'automatiser un processus très spécifique qui est complexe, qui demande beaucoup de temps de développement, nécessite la maintenance de beaucoup de modules pour un gain de temps minime. La comptabilité prend en moyenne quatre demi-journées tous les trois mois. Donc huit jours dans l'année.

4.1.2 Avantage dans notre projet

L'avantage est que lorsque ce test sera complètement automatisé, il pourra être régulièrement effectué. Ce qui n'est pas le cas aujourd'hui

D'autre part, l'autre avantage d'avoir développé les modules d'automatisation comptable a toujours été de pouvoir réutiliser les modules créés pour d'autres tests. Ainsi, de nouveaux tests ont pu être automatisés et possèdent un gain de productivité notable. Le test Arjel en est un exemple.

4.2 Qualification de l'Arjel

4.2.1 Description

La personne chargée de faire les tests de non-régression Arjel, doit effectuer plusieurs actions répétitives.

En effet, cette qualification oblige la personne à créer une offre complète dans le back office de ABP ; c'est-à-dire qu'elle doit créer pour tous les sports, tous les « Markets » qui y sont associés. Elle doit ensuite coter chaque pronostic de chaque « Market » avant de pouvoir ouvrir ces « Markets », c'est-à-dire les rendre disponibles sur le front office du site ParionsWeb.

Ensuite, elle doit effectuer une prise de pari sur chacun des Markets créés.

Et enfin, elle doit clôturer, promulguer, et payer les Markets créés.

4.2.2 Avantage de l'automatisation

Pour les trois principales actions de la qualification Arjel détaillées ci-dessus, l'automatisation facilite grandement leur exécution. Le temps de travail de la personne en charge de cette qualification a fortement diminué.

En effet, l'automatisation évite à la personne en charge de cette qualification d'effectuer des tâches longues et répétitives. Avec plus de 230 Markets à créer, coter, et ouvrir, la création d'une offre manuelle Arjel prend en moyenne 2 journées de travail et est très épuisante pour la personne en charge de cette tâche.

Effectuer des prises paris sur chacun des Markets oblige la personne à s'authentifier, à rechercher le Market sur lequel elle souhaite faire une prise de pari, mettre le montant de la mise, valider la prise de pari, noter le numéro de la transaction et enfin passer à l'autre prise de pari. Cette action prend en moyenne 1 journée et est également fatiguant pour la personne car très répétitive.

Et finalement, la personne doit clôturer, promulguer et payer tous les Markets qu'elle vient de créer. Cette action prend en moyenne une demi-journée et comme pour les autres actions, est très répétitive.

En tenant compte du fait que l'offre sportive évolue sans cesse et qu'il y a donc de plus en plus de sports et de Markets, le temps de travail augmente donc d'une livraison à l'autre. Avec l'outil de création de l'offre sportive, le temps de travail manuel concerne seulement la mise à jour du nombre de Markets et des prises de paris dans les fichiers de configuration.

Les outils de création et de clôture de l'offre sportive ainsi que d'injection de prises de paris évitent toutes ces actions manuelles, en fournissant le moyen de créer rapidement des Markets, de les coter, de les ouvrir, d'effectuer des prises de paris dessus, de clôturer les Markets, de les promulguer et de les payer de manière automatique. **En quelques heures, le module d'automatisation effectue le travail d'une personne à temps plein pendant trois jours et demi.** Cette personne peut pendant ce temps contrôler des cas qu'elle n'aurait pu contrôler auparavant comme, par exemple, vérifier des points précis de l'application. On notera également que sans l'automatisation, ce test Arjel n'aurait pu être effectué. Nous avons donc pu augmenter la couverture des tests.

L'autre avantage est de pouvoir facilement élargir le périmètre des tests sans trop d'efforts. Si d'une livraison à l'autre de nouveaux Markets viennent à être ajoutés, la charge de travail concernerait uniquement la mise à jour des fichiers d'entrées des modules concernés, et non pas la création manuelle des anciens et nouveaux Markets.

4.2.3 Mise en œuvre

Si l'on reprend le tableau de correspondance établi lors de l'analyse préliminaire, nous remarquons que toutes les sous-fonctions utilisées pour la qualification Arjel sont aussi utilisées pour la qualification comptable. Le développement de ces sous-fonctions au travers de modules génériques a permis d'automatiser la qualification Arjel.

Fonctions Sous-fonctions	Qualifier la Comptabilité	Qualifier la bascule	Qualifier l'Arjel	Qualifier le paiement	Qualifier la réconciliation	Qualifier Betradar
S'authentifier	X	X	X	X	X	
Faire une prise de paris	X	X	X	X	X	
Créer un « Market »	X	X	X	X	X	
Coter un « Market »	X	X	X	X	X	
Ouvrir un « Market »	X	X	X	X	X	
Clôturer un « Market »	X	X	X	X	X	
Promulguer un « Market »	X	X	X	X	X	
Annuler un « Market »	X	X	X	X		
Contrôler en base de	X		X	X	X	
Lire un fichier CSV	X			X	X	
Générer un fichier CSV	X	X	X	X	X	
Exécuter un script					X	
Lire un fichier XML						X

Si nous reprenons le tableau de correspondances entre les sous-fonctions et les modules, créé lors de l'analyse détaillée, nous en déduisons que certains modules nécessaires à la qualification Arjel ont déjà été développés et peuvent être de nouveau utilisés. Le tableau ci-dessous liste les modules utiles pour automatiser la qualification Arjel.

Modules	Fonction	Qualifier l'Arjel
	Injecter	X
	Create_Market	X
	Cloture_Market	X
	CompareReporter	X

Les trois premiers modules sont actuellement utilisés pour automatiser en partie la qualification de l'Arjel. Les fichiers et les paramètres d'entrées ne sont pas les mêmes que pour la qualification comptable, ils ont été modifiés pour élargir le nombre de « Markets » gérés et de prise de paris effectuées. Seul le contrôle en base de données se fait toujours manuellement. L'adaptation des fichiers de paramétrages et d'entrées pour ce module n'a pas encore été faite pour cette qualification.

CONCLUSION

L'objectif de départ du projet était l'automatisation de l'ensemble des tests de non-régression techniques effectués par l'équipe HMOA Multimédia Sport. L'identification des besoins et l'analyse de l'existant ont montré que l'objectif initial n'était pas atteignable dans le temps imparti et anticipait sur les futurs besoins de la Direction. Cette analyse a donc permis de préciser les attentes, de définir les priorités et les bases d'une architecture de type modulaire. Dans le même temps, une analyse des risques a permis d'émettre des réserves quant à l'utilisation d'outils imposés, non maîtrisés. C'est seulement lorsque l'objectif d'automatisation du test de non-régression comptable a été clairement défini que j'ai commencé à réfléchir sur la méthodologie à utiliser pour ce projet. L'anticipation des futurs besoins, la proximité avec les personnes concernées et l'incertitude sur l'évolution du projet m'a conforté dans le choix d'une méthodologie de gestion de projet de type agile.

Par la suite, il fut nécessaire de comprendre le fonctionnement du test comptable avant de pouvoir commencer la réalisation du projet. Plusieurs entretiens ont, dans ce sens, été réalisés et une analyse détaillée a mis en évidence quels modules étaient concernés et comment ils devaient s'ordonner.

Dans une première étape, nous avons vu que l'utilisation et l'adaptation d'un des outils existants n'ont pas été un franc succès. Celui-ci était techniquement limité. Il se basait sur l'interface graphique de l'application et que celle-ci était souvent amenée à évoluer. Cependant, c'est également durant cette étape que nous avons pu apprécier de disposer d'une architecture modulaire et de l'agilité dans notre projet, nous permettant ainsi de réagir rapidement à cette situation, en faisant valider par la Direction, le remplacement des modules inexploitables.

Lors des deuxième et troisième étapes, nous avons tenu compte des erreurs précédentes et avons respecté les deux conditions nécessaires à la réussite de l'automatisation de nos modules : à savoir, ne pas être limités techniquement par l'outil et dépendre le moins possible d'une autre application (et si cela s'avère nécessaire, que celle-ci soit le plus stable possible).

C'est également pendant ces étapes que nous avons découvert les avantages à utiliser le framework Scrum pour gérer un projet. En effet, nous avons vu que la maintenance du planning des développements s'est faite très facilement et directement par l'équipe de développement. L'équipe est devenue autonome et les réunions quotidiennes ont grandement amélioré la communication au sein celle-ci. Les membres, peu enclins à parler de leurs difficultés, l'on fait avec plus de facilité à l'occasion de ces réunions, les problèmes ont pu être posés et du temps a été gagné. L'utilisation de Scrum au quotidien a pour beaucoup contribué à la réussite de ce projet.

Ensuite, c'est l'utilisation au quotidien, des modules automatisés par les utilisateurs qui nous a permis de juger si le projet a constitué une réussite ou non. Nous avons vu que plusieurs parties du processus d'automatisation comptable ont nécessité d'entreprendre des actions manuelles de la part des utilisateurs. Nous avons également vu, que ces derniers ont été peu enclins à utiliser un outil qui n'est pas entièrement automatisé, surtout quand le gain de productivité n'apparaît pas évident. Néanmoins, le choix d'automatiser la comptabilité et de part ce fait, les modules les plus réutilisables, a permis de mettre à disposition des outils qui ont fait gagner beaucoup de temps aux utilisateurs. C'est en cela une réussite du projet.

Finalement, à l'issue de ce projet, il apparaît clair que l'automatisation d'un processus aussi complexe que la comptabilité n'a pas été le meilleur choix ; que la réussite de ce projet tient à la

capacité de l'équipe à s'être adaptée aux changements, facilitée par une gestion de projet Agile et une architecture bien pensée au départ.

Nous avons constaté à quel point il est nécessaire, à l'avenir, que les tests que l'on souhaite automatiser soient simples, aient un fort potentiel de retour sur investissement, soient très souvent effectués et utilisent des applications tierces stables dans le temps.

Sur un plan plus personnel, l'ensemble des travaux réalisés dans le cadre de ce mémoire m'ont permis réflexivement d'éprouver le déroulement d'un projet informatique au regard des contraintes qui peuvent surgir, des réalités économiques et professionnelles pouvant se manifester, mais aussi au regard d'une dimension relationnelle et humaine qui reste inhérente à toute gestion de projet. Répondre aux besoins, s'adapter aux situations et gérer un projet ont été au cœur de cette expérience enrichissante à tout point de vue. Il est notable, d'ailleurs, que les propositions ont été dans l'ensemble bien accueillies par le responsable de l'équipe, et cette dernière, qui me faisaient totalement confiance. Mon autonomie s'en est trouvée renforcée, m'aidant à prendre confiance en moi-même et me permettant d'être force de propositions. J'ai également pu mettre en application, lors de la démarche projet, les connaissances que j'avais acquises tout au long de ma formation au CNAM. Ce projet est aussi, pour moi, l'occasion de finaliser ma formation d'ingénieur CNAM en informatique, option système d'informations et d'obtenir mon diplôme.

ANNEXES

Annexe 1 - Plan de recette comptable

	Date PDJ	Montant PDJ	26-oct	27-oct	28-oct	Statut	Gain	Remboursé
metropole	Jackie10	Main						
		J1	1 Nantes			Gagnant		1,1
		J1	7					
			1 Nantes			Gagnant		1,1
			1	Sampras		Annulé		
			1		Créteil	Gagnant		1,1
			1 Nantes		Créteil	Gagnant		1,21
			1	Sampras	Créteil	Annulé/Gagnant		1,1
			1 Nantes		Créteil	Gagnant		1,1
			1 Nantes		Créteil	Annulé/Gagnant		1,21
metropole	Live							
	J1	1 Lourdes			Gagnant		1,1	
DOM	Jackie20	Main						
	J1	1 Nantes			Créteil	Gagnant		1,21
	J1	1		Toulouse	Perdant		0	
	Live							
	J1	1 Cannes			Perdant		0	
St Pierre et Miquelon	Jackie40							
	J1	1		Sampras	Annulé			1
Monaco	Juliano							
	J1	1 Nantes			Limoges	Annulé/Gagnant		1,1
St Bart	Jackie50							
	J1	1 Verzon			Perdant		0	
	J2	1		Rennes	Perdant		0	
St Martin	Jackie30							
	J1	1 Nantes			Bordeaux	Perdant		0
St Martin	Jackie30							
	J2	1 Bastia			Gagnant		1,1	
			26-oct	27-oct	28-oct			
	CA encaissé		16	2	0			
	CA participant		5	6	7			
	CA remboursé							
	CA participant net		5	4				
	CA encaissé non encore participant		11	7	0			
	Gains		3,3	2,2	6,93			

Figure 0-1 Jeu de données comptables en entrée Plan de Recette V3

3	Résulter Match	Résulté						
	26-10	Eq gagnante			OX1	OX2	OX3	X
Main	Nantes - Vierzon	Nantes	22		nantes	null	Vierzon	1
Live	Cannes-Lourdes	Lourdes	31		lourdes	null	Cannes	2
	27-10							
Main	Sampras- Agassi	Annulé	45		Sampras	null	Agassi	3
Main	Toulouse-Paris	Paris	48		Paris	null	Toulouse	4
Live	Rennes-Bastia	Bastia	55		Rennes	null	Bastia	5
	28-10							
Main	Créteil-Bordeaux	Créteil	101		Créteil	null	Bordeaux	6
Main	Limoges-Tarbes	Annulé	153		Limoges	null	Tarbes	7

RESULTATS COMPTABLES

J1	1	5	2	3	8	9	
	Metropole	Monaco	DOM	St Pierre	St Barth	St Martin	Total
CA encaissé	9	1	3	1	1	1	16
CA participant	3		1			1	5
CA remboursé							0
CA participant net	3		1			1	5
CA encaissé non encore participant	6	1	2	1	0	1	11
Gains	3.3						3.3

RESULTATS COMPTABLES

J2	1	5	2	3	8	9	
	Metropole	Monaco	DOM	St Pierre	St Barth	St Martin	Total
CA encaissé					1	1	2
CA participant	2		1	1	1	1	6
CA remboursé	2			1			3
CA participant net	1		1		1	1	4
CA encaissé non encore participant	4	1	1			1	7
Gains	1.1					1.1	2.2

RESULTATS COMPTABLES

J3	1	5	2	3	8	9	
	Metropole	Monaco	DOM	St Pierre	St Barth	St Martin	Total
CA encaissé							
CA participant	4	1	1			1	7
CA remboursé	2	1					3
CA participant net	4	1	1			1	7
CA encaissé non encore participant							
Gains	4.62	1.1	1.21				6.93

Figure 0-2 Résultats attendus pour tous les jours

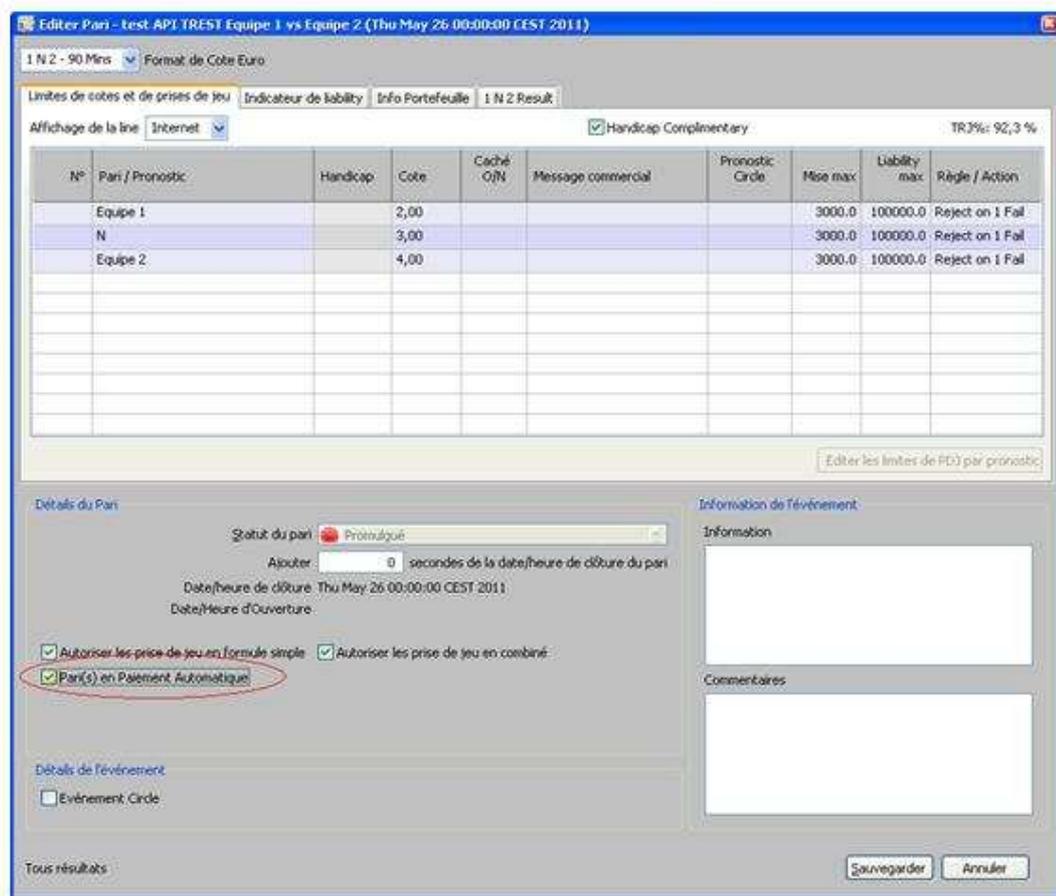
Annexe 2 - Demande d'évolution pour l'ajout de fonctionnalités de paiement

Settlement

Objectifs

Fournir la possibilité depuis l'API REST d'effectuer un paiement à l'identique de ce qui est fait actuellement dans le BackOffice lorsque l'on clique sur le bouton « paiement automatique » après avoir promulgué un market ou un Event.

1^{ère} priorité : Settlement depuis un market



Analyse

Mise à jour du statut du market

Request :

```
{  
  "status": "settled"  
}
```

template.doc

Page 6/8

*Ce document est la propriété de la société Française des Jeux.
Il ne peut être utilisé, reproduit ou divulgué sans son autorisation écrite préalable.*

Figure 0-3 Illustration de la demande d'évolution (part 1).

- Cette action déclenche le processus de paiement du market identique au BackOffice en plaçant dans la table [REDACTED] les demandes de paiement, concernant le market, qui seront par la suite traitées par le process Settlement d'AGP qui :
- Dépile les demandes de la table [REDACTED] s'il n'y a pas de bloquer de réconciliation présent.
- Mettra à jour la table « [REDACTED] »
- Enregistre des logs

2nd priorité : Settlement depuis un Evénement

The screenshot displays a web application interface for event statistics. At the top, there are tabs for '90 Mins', 'Mi-temps', '2ème Mi-temps', 'Match', and 'D'autres'. Below these, the score is shown as 'EQUIPE 1' with a score of '1' and 'EQUIPE 2' with a score of '0'. There are sections for 'Statistiques' and 'Subeur et Disciplinary'. At the bottom, there are checkboxes for 'Période annulé' and 'Paiement automatique pour Evénement', with the latter being checked and circled in red. A JSON snippet at the bottom shows 'atatus': 'settled'.

*Ce document est la propriété de la société Française des Jeux.
Il ne peut être utilisé, reproduit ou divulgué sans son autorisation écrite préalable.*

Figure 0-4 Illustration de la demande d'évolution (part 2).

}

- Cette action déclenche le processus de paiement du market identique au BackOffice en plaçant dans la table settlementqueue les demandes de paiement, des markets associés à l'évènement, qui seront par la suite traitées par le process Settlement d'AGP qui :
- Dépile les demandes de la table settlementqueue s'il n'y a pas de bloquer de réconciliation présent.
- Mettra à jour la table « betslipupdatequeue »
- Enregistre des logs

1^{ère} priorité

Mise en place du Settlement depuis le market

2nd priorité

Mise en place du Settlement depuis l'évènement

Tableau Récapitulatif demande d'évolution

Fonctionnalité	Element concerné	Priorité	Livraison
FeedCode	eventPath (sport, catégorie, league), Event (GameEvent, RankEvent)	1	11.1.3
FeedCode	Market Outcome Price Opponent Player	2	11.1.3
Settlement	Market	1	11.1.3
Settlement	Event	2	11.1.3

Figure 0-5 Illustration de la demande d'évolution (part 3).

Annexe 3 - Illustration de l'ordonnancement des modules *via* QC

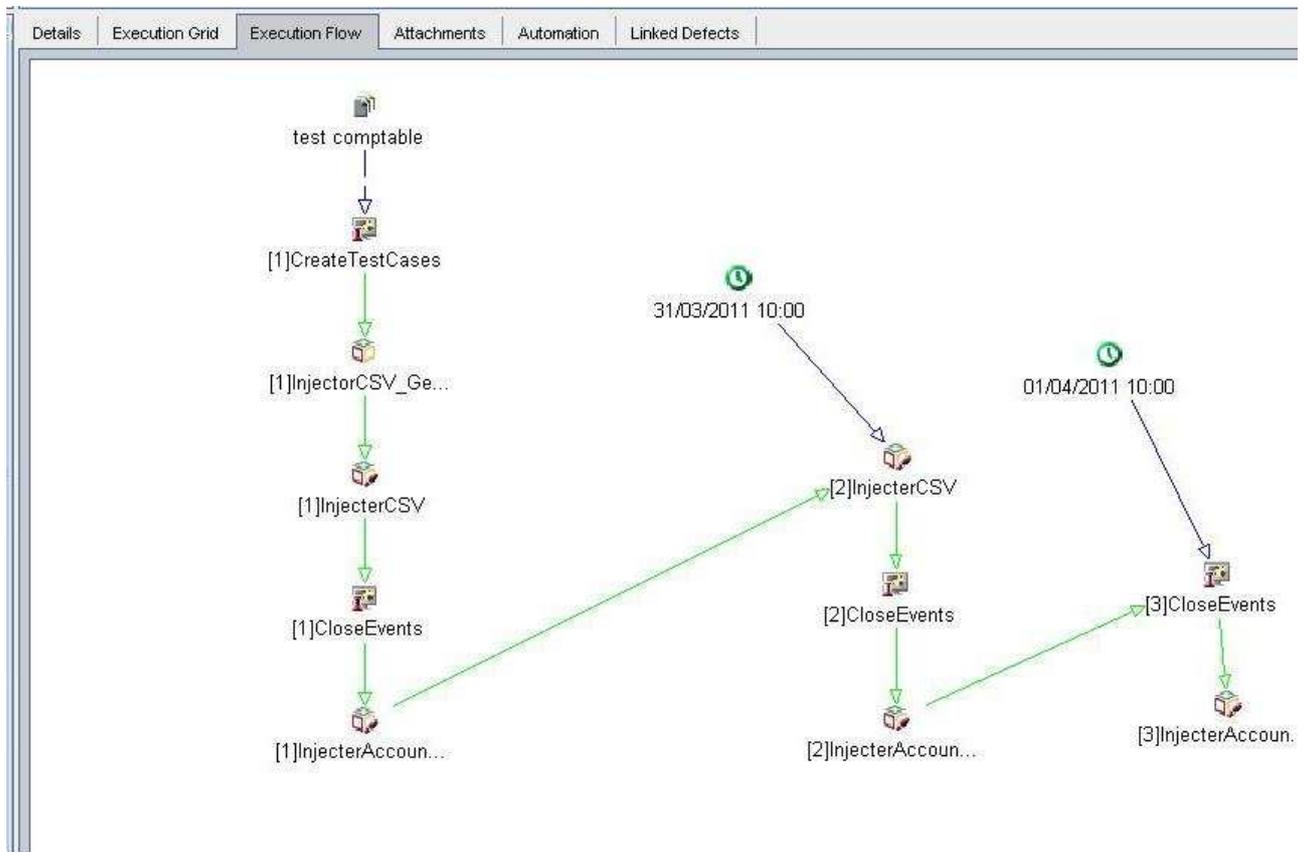


Figure 0-6 Illustration du processus comptable automatisé depuis un « TestSet » QC

Annexe 4 - Extrait d'un code VBScript mis en place dans QC

```
=====
'
' Main Test Function
' Debug - Boolean. Equals to false if running in [Test Mode] : reporting to Quality Center
' CurrentTestSet - [OTA COM Library].TestSet.
' CurrentTSTest - [OTA COM Library].TSTest.
' CurrentRun - [OTA COM Library].Run.
=====
Sub Test_Main(Debug, CurrentTestSet, CurrentTSTest, CurrentRun)
' *** VBScript Limitation ! ***
' "On Error Resume Next" statement suppresses run-time script errors.
' To handle run-time error in a right way, you need to put "If Err.Number <> 0 Then"
' after each line of code that can cause such a run-time error.
On Error Resume Next

' clear output window
TDOutput.Clear
CONST FAILURE = "Failed"
CONST PASSED = "Passed"
Dim TEXTPARAM
Dim VALUEPARAM
Dim startDate
Dim expectedFileNumber
Dim Env
' ***** Get a parameter *****
If CurrentTSTest.HasSteps Then
Set StepParams = CurrentTSTest.Params ' is a TSTest object
With StepParams
Configfile = Trim(.ParamValue("configfile"))
End With
end If
' ***** Define Step One *****
Set ObjRun = CurrentRun
Set StepF = ObjRun.StepFactory
StepF.AddItem("Création de l'offre sportive")
Set StepList = StepF.NewList("")
nStepKey = StepList.Count
StepList.Item(nStepKey).Field("ST_DESCRIPTION") = "This Step permit to create offer"
StepList.Item(nStepKey).Field("ST_EXPECTED") = PASSED
StepList.Item(nStepKey).Field("ST_STATUS") = PASSED
' ***** EXECUTION *****
sCmd = "..\Automatisation\bin\OfferManager\_mainLoader.exe " & Configfile
' Xtools.run sCmd, "", -1
Set oExec = WshShell.Exec(sCmd)
' *****
' call COM method : String Status
' The test execution status.
Dim result ' String
result = oExec.Status
TDOutput.Print CStr(result)
' handle run-time errors
If Err.Number <> 0 Then
TDOutput.Print "Run-time error [" & Err.Number & "] : " & Err.Description
' update execution status in "Test" mode
If Not Debug Then
CurrentRun.Status = FAILURE
CurrentTSTest.Status = FAILURE
End If
End If
```

Figure 0-7 Extrait du code source du VBScript dans QC appelant un exécutable

Annexe 5 - Extrait du code d'un rakefile

```
s.platform = Gem::Platform::RUBY
s.required_ruby_version = '>=1.9'
#s.extra_rdoc_files << 'README'
#s.rdoc_options <<
# '--title' << 'JSON implementation for ruby' << '--main' << 'README'
# s.test_files.concat Dir['./tests/test_*.rb']
s.author = "A. Al Oumami & S. Huerta"
s.add_dependency('log4r', '>=1.1.9')
s.add_dependency('activemodel', '>=3.0.7')
s.add_dependency('activesupport', '>=3.0.7')
s.add_dependency('inifile', '>=0.4.1')
s.add_dependency('json', '>=1.5.1')
s.add_dependency('rest-client', '>=1.6.1')
s.add_dependency('utf8_validator', '>=1.0.0')
s.add_dependency('dbi', '>=0.4.5')
s.add_dependency('ruby-oci8', '>=2.0.6')
s.add_dependency('roxml', '>=3.1.6')
s.add_dependency('ocra', '>=1.2.0')
end

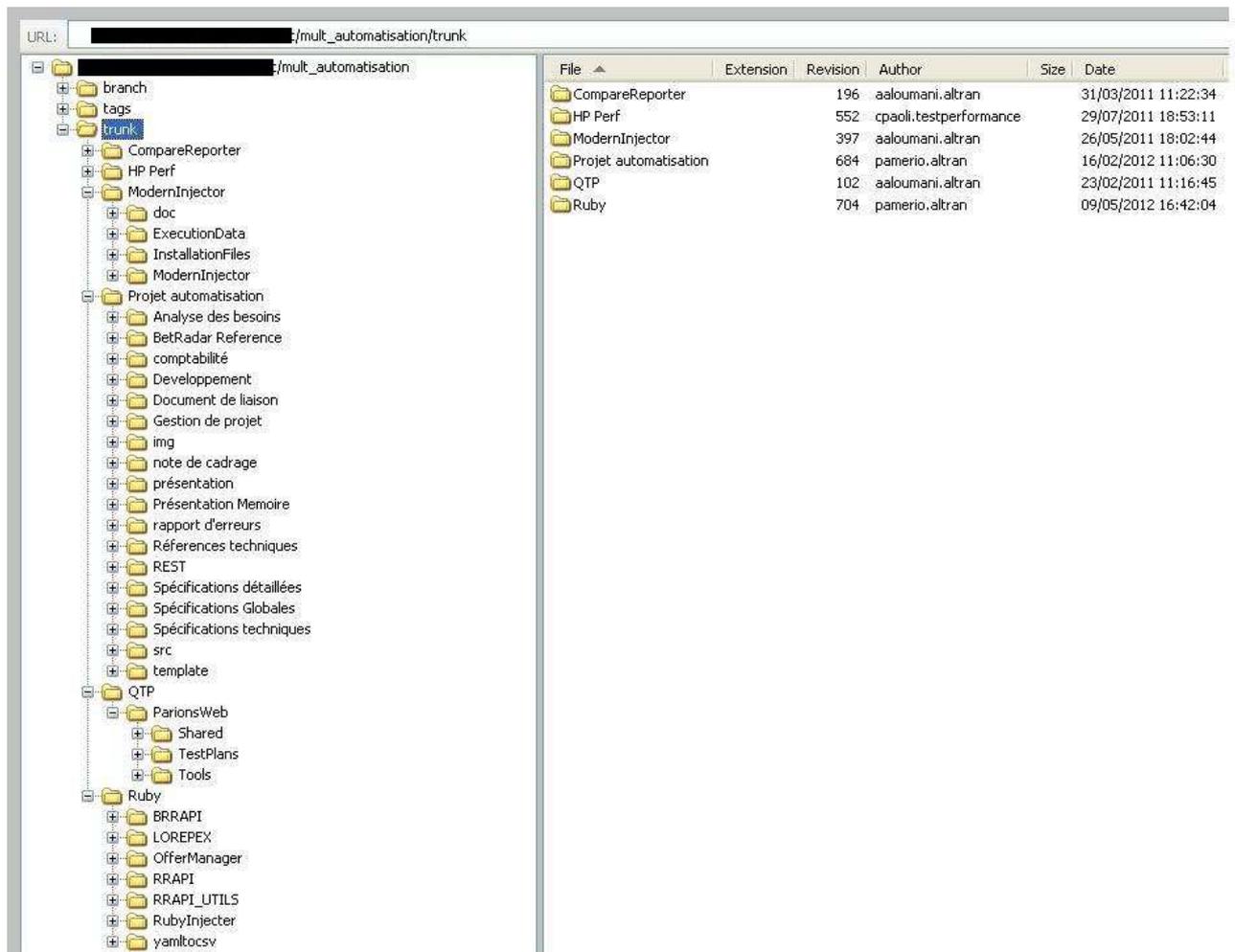
task :default => [:build_bin]
task :clean => [:clean]

task :build_bin do
  puts "===== Building _mainLoader.rb ====="
  begin
    system("ocra #{File.dirname(__FILE__)}/lib/_mainLoader.rb --no-autoload")
    puts "===== Done building _mainLoader.exe ====="
    require 'fileutils'
    FileUtils.mv(File.dirname(__FILE__) + "/_mainLoader.exe", File.dirname(__FILE__) + "/pkg/_mainLoader_#{PKG_VERSION}.exe")
    puts "===== Check '"+File.dirname(__FILE__) + "/pkg/' Directory! "
  rescue SystemCallError
    puts "Could not build _mainLoader.exe"
  end

  begin
    system("ocra #{File.dirname(__FILE__)}/lib/_mainCloser.rb --no-autoload")
    puts "===== Done building _mainCloser.exe ====="
    require 'fileutils'
    FileUtils.mv(File.dirname(__FILE__) + "/_mainCloser.exe", File.dirname(__FILE__) + "/pkg/_mainCloser_#{PKG_VERSION}.exe")
    puts "===== Check '"+File.dirname(__FILE__) + "/pkg/' Directory! "
  rescue SystemCallError
    puts "Could not build _mainCloser.exe"
  end
end
```

Figure 0-8 Extrait du code source du rakefile utilisé pour la création des exécutable.

Annexe 6 - Répertoire SVN utilisé tout au long du projet



The screenshot displays an SVN repository browser interface. The left pane shows a tree view of the repository structure, with the 'trunk' folder selected. The right pane shows a table of files with their metadata.

File	Extension	Revision	Author	Size	Date
CompareReporter		196	aaloumani.altran		31/03/2011 11:22:34
HP Perf		552	cpaoli.testperformance		29/07/2011 18:53:11
ModernInjector		397	aaloumani.altran		26/05/2011 18:02:44
Projet automatisé		684	pamerio.altran		16/02/2012 11:06:30
QTP		102	aaloumani.altran		23/02/2011 11:16:45
Ruby		704	pamerio.altran		09/05/2012 16:42:04

Figure 0-9 Illustration du répertoire SVN utilisé pour le suivi de version des fichiers

BIBLIOGRAPHIE

Réseau Anact (*Organisation et management par projet*, 2006). Disponible sur : <http://www.anact.fr/portal/pls/portal/docs/1/298337.PDF>.

Mercury Interactive Corporation (*Mercury QuickTest Professional Tutorial, Version 8.0*, 2004). Disponible sur : <http://think1808.files.wordpress.com/2008/03/qtp-tutorial.pdf>.

Mercury Interactive (Israel) Ltd. (*QuickTest Professional Automation Object Model Reference*, 2009).

Adobe Systems Incorporated (*Testing ADOBE FLEX 4 Applications with HP QuickTest Professional*, 2010). Disponible sur : http://help.adobe.com/en_US/Flex/4.0/TestingQTP/flex_4.0_testingqtp.pdf

Mercury Interactive (Israel) Ltd (*HP QuickTest Professional, Software Version: 10.00, Add-ins Guide*, Janvier 2009). Disponible sur : <http://www.asi-test.com/Docs/QTAddinsGuide.pdf>

Reinhardt Moller (*Integrating QTP with Terminal Emulators*, 21 Avril 2009). Disponible sur <http://www.jds.net.au/tech-tips/qtp-with-terminal-emulators/>

Lars Christensen (*One-Click Ruby Application Builder*, 2009-2010). Disponible sur : <https://github.com/larsch/ocra>

Anshoo Arora (*Scheduling QTP Tests in Quality Center*, 21 juillet 2010). Disponible sur : <http://support.idhasoft.com/entries/219807-scheduling-qtp-tests-in-quality-center>

K. Schwaber (*Agile project management with Scrum*. Redmond, Wash.: Microsoft Press, 2004).

ScrumAlliance (*transforming the world at work. Suggested Reading*). Disponible sur: http://www.scrumalliance.org/pages/scrum_student_resources

Yip, Jason (*It's Not Just Standing Up: Patterns of Daily Stand-up Meetings*, 2006). Disponible sur: <http://martinfowler.com/articles/itsNotJustStandingUp.html>

Documentation Ruby. Disponible sur : <http://apidock.com/ruby>, <https://github.com/>, <http://www.ruby-doc.org/>

David Thomas with Chad Fowler and Andy Hunt, (*The Pragmatic Programme's Guide*, 2010).