

Test et recette dans un projet logiciel : évolution des méthodes et outils

Drifa Kesraoui

► To cite this version:

Drifa Kesraoui. Test et recette dans un projet logiciel : évolution des méthodes et outils. domain_shs.info.docu. 2014. mem_01128924

HAL Id: mem_01128924

https://memsic.ccsd.cnrs.fr/mem_01128924

Submitted on 10 Mar 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.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives| 4.0 International License



CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Ecole Management et Société-Département CITS

INTD

MEMOIRE pour obtenir le
Titre professionnel "Chef de projet en ingénierie documentaire" INTD
RNCP niveau I

Présenté et soutenu par

Drifa KESRAOUI

Le 17 décembre 2014

Test et Recette dans un Projet Logiciel, Evolution des Méthodes et Outils

Jury :

SCOPSI Claire Maitre de conférences, Laboratoire Dicen-IDF, CNAM-INTD
SIDHOUMI Hakima, Présidente chez Salihanne-Consulting

Promotion 43



Dédicaces

Je dédie ce mémoire à ceux que j'aime et qui vivent loin de moi. J'aurais bien aimé passer ce moment important et solennel de ma vie en leur présence, je regrette leur absence :

- Ma mère « Yema », qui a œuvré pour ma réussite, de par son amour, son soutien, ses précieux conseils, tous ses sacrifices et sa présence dans ma vie. Elle a su faire de moi une personne véritablement indépendante. Reçois à travers ce travail, l'expression de mes sentiments d'amour et d'affection et de mon éternelle gratitude. Je t'aime maman.

- Mon père « Vava », qui peut être fier de trouver ici le résultat de mes longues années d'absence, je te dis pardon d'être partie « Elghourva ». Merci pour les valeurs nobles ; l'éducation et le soutien permanent venus de toi. Tu n'as jamais cessé de me prodiguer de sages conseils à suivre à l'étranger, tu m'as appris à forger la personne que je suis.

Que Dieu le tout puissant soit à vos côtés et vous accorde une bonne santé «thughzi laamar».

- Mes sœurs : Nora, Dhrifanagh, Kheloudja, Dalila, Fatiha, Souad, qui m'ont toujours entourée, motivée et soutenue durant toutes ces années d'études en France. Vous êtes toujours présentes dans des moments difficiles pour me consoler. Pour toute la complicité et l'entente qui nous unissent, ce travail est un témoignage de mon attachement pour vous.

- Mon frère Ali « Adjinagh » et sa femme Kenza qui m'ont donné le maximum de ce qu'ils pouvaient. Vos conseils, vos encouragements et vos prières ont toujours guidé mes pas vers la réussite. Que ce travail soit l'expression de mon attachement et de mon amour pour vous.

- Mes petits neveux et nièces adorés : Amira, Daoud, Hmidouche, Chahinez, Sarah, Melissa, Menad, Yassine, Tiziri, Sabrine, Meriem. Ce modeste travail doit vous servir d'exemple pour réussir et faire mieux que votre tante. Je vous aime tous.

- A la mémoire de ma grand-mère « Djida Chaveha » qui m'a beaucoup aidée par tous les moyens possibles pour étudier. Tu as toujours été dans mon esprit et dans mon cœur. Je te dédie aujourd'hui ma réussite. Que Dieu t'accueille dans son éternel paradis.

- Mes beaux frères Hamitouch, Nadim, Djamel, Ahcene ; Je ne pourrais jamais exprimer à sa juste valeur le respect que j'ai pour vous.

"N'allez pas là où le chemin peut mener. Allez là où il n'y a pas de chemin et laissez une trace." Ralph Waldo Emerson.



Remerciements :

Je tiens à remercier très chaleureusement:

- Claire SCOPSI, Maître de conférences au CNAM-INTD, ma directrice de mémoire, de m'avoir encadrée et aidé à structurer ma réflexion. Je suis particulièrement reconnaissante pour ses remarques pertinentes, ses conseils et ses encouragements tout au long de ce travail.
- Mes professeurs et toute l'équipe pédagogique de l'INTD. Merci pour votre compréhension, vos conseils et surtout pour votre soutien dans mes démarches administratives compliquées. Sans votre aide je n'aurais pas pu achever cette formation.
- Hakima SIDHOUMI, qui m'a chaleureusement accueillie au sein de son cabinet de conseil en informatique « Salihaane-Consulting », pour un stage de trois mois. Je la remercie pour m'avoir intégrée pleinement aux activités du service.
- Mes chers amis: Saliha, Hayet, Karim, Zahra, Fariza, Afaf, Jugurtha, Nassima, Ouardia Malika. Vous m'avez tous soutenue, protégée et encouragée durant toutes ces années.
- GILLES SAINT-GAL, Vice-président du Conseil général, Melle DAOUST et Mme DESBROSSE de la Mairie de Fontenay-sous bois, pour leur aide dans mes démarches administratives. Que ce travail soit le témoignage sincère de ma profonde reconnaissance pour leur soutien et leur encouragement.
- Tous mes collègues de formation qui m'ont aidée et soutenue. En particulier Laëtitia CAMMAS pour sa disponibilité, sa générosité. Sans oublier Francesca et Inès, Amélie, Clara, Pierre, Emilie, Jean-Paul, Franck, Gallia pour nos échanges constructifs pendant la période du travail en équipe.
- Mohand Thar MOHAMMEDI et Jean-Pierre NAINA, consultants en informatique MOA/ MOE, pour toutes les connaissances en informatique qu'ils m'ont transmises, leurs disponibilité et précieux conseils. Recevez ce travail en signe de ma reconnaissance et ma profonde estime.
- Denis DEMONTOUX, chef du bureau des libertés publiques et de la citoyenneté (Sous-préfecture d'Argenteuil), pour m'avoir accordé sa confiance et donné une possibilité de continuer ce projet professionnel que j'ai failli abandonner à cause des complications administratives. Je vous prie de trouver l'expression de ma sincère reconnaissance et profonde gratitude.
- CHMITLIN Anne, son mari Jean-Louis et leurs enfants qui m'ont toujours reçue chaleureusement chez eux et m'ont beaucoup appris et transmis de la culture Française.

Notice

KESRAOUI Drifa. "Test et Recette dans un Projet Logiciel, Evolution des méthodes et Outils". 2014. 100 p. Mémoire pour le titre professionnel de niveau 1, Chef de projet en ingénierie documentaire, CNAM-INTD, 2014.

La phase Test, représente une partie importante de tout projet logiciel. Les entreprises doivent se doter des outils complets et modernes de gestion des tests, dans le but d'assurer le bon fonctionnement des applications avant qu'elles soient mises en production.

Ce projet de mémoire, porte essentiellement, sur la valorisation de la phase « test » et les enjeux majeurs d'une recette fonctionnelle. Pour cela nous présentons toutes les nouvelles tendances du test logiciel sur le marché, permettant d'améliorer l'activité de test, obtenir un RIO (le retour sur investissement) positif et une forte rentabilité. Nous nous intéressons aussi aux grands acteurs du secteur consulting en informatique qui proposent des services d'assistance à la maîtrise d'ouvrage, et des solutions TIC.

Nous terminons par une présentation d'un modèle de démarche de pilotage d'un projet de recette, en s'appuyant sur le stage effectué au sein du cabinet de conseil « Salihanne-Consulting ». Nous expliquons pourquoi « Quality Center » est l'outil le plus utilisé par les consultants fonctionnels, après avoir fait une analyse comparative avec les autres outils.

RECETTE FONCTIONNELLE, AUTOMATISATION DES TESTS, REFERENTIEL DE TEST, BENCHMARKING, METHODES AGILES, VEILLE CONCURRENTIELLE, CABINETS DE CONSEIL, TESTS MANUELS, BUGS, CYCLES DE DEVELOPPEMENT, SOLUTIONS LOGICIELLES.

The "Test" phase, is an important part of any software project. Companies must provide complete and modern tools for test management, in order to ensure the proper operation of applications before they are commissioned.

This project thesis, focuses mainly on developing the 'test' phase and on the major issues of functional acceptance. For this, we introduce all the new market trends in software testing, improving test activity, improve RIO (Return On Investment) and strong profitability. We are also interested in the big players in the IT consulting industry offering ICT solutions and support services for project managers.

To conclude, we introduce a model of a process of a functional acceptance, based on a working experience in the consulting firm "Salihanne-Consulting".

After a comparative analysis of other tools, we explain why "Quality Center" is the most used tool by functional consultants.

Table des matières

CONSERVATOIRE NATIONAL DES ARTS ET METIERS	1
Ecole Management et Société-Département CITS.....	1
INTD.....	1
Remerciements :	3
Notice.....	4
Table des matières.....	5
Liste des tableaux.....	8
Liste des figures.....	9
Introduction	10
Première partie : Prestation du métier d' MOA DE Recette Fonctionnelle.....	14
1 Présentation de la société « Salihanne-Consulting »	16
1.1 Les services du cabinet « Salihanne-Consulting »	17
1.1.1 Intégration, Paramétrage ERP	17
1.1.2 Développement sur mesure d'applications web	17
1.1.3 Mobilité et Telecom.....	17
1.1.4 Assistance à la maîtrise d'ouvrage	17
1.1.5 Maitrise d'ouvrage	17
1.1.6 Réseaux	18
1.1.7 Maitrise d'œuvre.....	18
1.2 La mission de stage	18
2 Les acteurs du projet informatique: MOA et MOE	19
2.1 Définition d'un projet	19
2.2 Equipe MOA (la maîtrise d'ouvrage)	19
2.3 Equipe MOE (la maîtrise d'œuvre) :	20
3 Introduction à la Recette Fonctionnelle :	22
3.1 Définition de la recette :	22
3.2 Les grands axes de la recette fonctionnelle :	23
4 Introduction au test logiciel	24
4.1 Qu'est-ce que tester ?.....	24
4.2 Les principes de base de l'activité « test »	24
5 Les Bugs et les conséquences	26
5.1 Qu'est-ce qu'un Bug ?	26
5.2 Qualification de la gravité des anomalies :	27

5.3	Les sources des Bugs logiciels	27
5.4	Exemples des Bugs aux conséquences désastreuses	27
Deuxième partie : Méthodes et Outils de Test, Historique et Evolution		30
1	Les tests manuels et les tests automatisés	32
1.1	Les tests manuels : (Méthodes traditionnelles)	32
1.2	Les tests automatisés : (Méthodes nouvelles) :	32
1.3	Avantages et inconvénients des tests manuels et les tests automatisés :	33
1.4	L'apport de l'automatisation.....	34
1.5	Les facteurs clés des tests fonctionnels automatisés :	35
2	Modèles de cycle de développement des logiciels (l'évolution)	36
2.1	Petit historique	36
2.2	Le cycle en « V » ou Cascade	37
2.2.1	Les inconvénients du cycle en V	38
2.3	La méthode Agile.....	39
2.3.1	Qu'est ce que la méthode Agile ?	39
2.3.2	D'où vient la méthode Agile ?	40
2.3.3	Les 4 valeurs de la méthode Agile :	41
2.3.4	Les 12 principes de la méthode Agile	42
2.3.5	Exemple des méthodes Agiles « SCRUM »	43
2.3.6	Avantages et inconvénients de la méthode Agile	45
2.4	Du cycle traditionnel à l'agilité : Les incontournables du test logiciel	47
3	La veille concurrentielle : les grands axes du projet.....	49
3.1	Présentation du projet effectué au stage	49
3.1.1	Audit de situation	49
3.1.2	Contexte et Objectif.....	50
3.2	Le processus de veille	50
3.2.1	La recherche d'informations.....	50
3.2.2	Le choix des sources	50
3.2.3	Le choix des outils	50
3.2.4	Etude de marché, analyse sectorielle	52
3.2.5	Les facteurs clés du succès de la veille	52
3.3	Le BENCHMARKING	53
3.3.1	Les objectifs du benchmarking.....	53
3.3.2	Panorama des référentiels de tests.....	54
3.3.3	Panorama des acteurs spécialisés dans la qualité logicielle	57
3.4	Actualité de l'activité de test	60
3.4.1	Les organisations professionnelles de l'activité de test.....	60
3.4.2	Agenda d'événements	62

3.4.3	Les publications sur les médias sociaux	62
3.4.4	La presse en parle :	64
3.4.5	Retour d'expérience : Témoignage/Interviews	65
Troisième partie: Organisation et planification de la Recette Fonctionnelle.....		67
1	Contexte et objectif de la recette.....	69
2	Les principes généraux de la recette	70
3	Les grandes étapes de la Recette fonctionnelle	71
3.1	Etape de la préparation :	71
3.2	Etape de la réalisation : pilotage de la recette	72
3.2.1	Définition des exigences.....	72
3.2.2	La préparation des tests :	73
3.2.3	L'exécution des tests :	73
4	Les points clés de la recette fonctionnelle.....	76
5	Les outils de test utilisés pour la mise en œuvre de la stratégie de recette	77
5.1	Présentation de Quality Center.....	77
5.2	Les modules du Quality Center :	78
5.3	Les modules de Quality Center	80
5.3.1	Module de Gestion	80
5.3.2	Module Exigences	80
5.3.3	Module Plan de test	80
5.3.4	Module Exécutions des tests.....	80
5.3.5	Module Anomalies	80
5.3.6	Module Tableau de bord	81
6	Lexique spécifique relatif au test	82
Conclusion.....		86
Bibliographie		89
Annexes		97
Annexe 1 « Pour aller plus loin ».....		98
Annexe 2 Glossaire des tests des logiciels		100
Annexe 3 Des solutions logicielles open sources		103

Liste des tableaux

Tableau 1: les différents niveaux d'anomalies	27
Tableau 2: tests manuels et tests automatisés: Avantages/Inconvénients	33
Tableau 3: Avantages et Inconvénients de la méthode Agile	45
Tableau 4: Test sur les projets Agiles	48
Tableau 5: Outils de test logiciel	54
Tableau 6: Panorama des acteurs de l'activité consulting	58

Liste des figures

Figure 1: Cabinet de Conseil Salihanne-Consulting.....	16
Figure 2: Processus du Model-Based Testing.....	23
Figure 3: Le BUG Informatique	26
Figure 4 : Méthode RAD	37
Figure 5: le Cycle en V.....	38
Figure 6: le cycle en W.....	Erreur ! Signet non défini.
Figure 7:Itératif-Incrémental-Adaptatif.....	39
Figure 8: le « Manifeste Agile »	40
Figure 9: les 4 valeurs de la méthode Agile	41
Figure 10: Les 12 principes de la méthode Agile	42
Figure 11: "La méthode SCRUM"	43
Figure 12: les étapes de la recette fonctionnelle	71
Figure 13: Quality Center	79

Introduction

La complexité des nouveaux développements ainsi que pour la maintenance informatiques rend impossible la réalisation des applications sans aucune anomalie. Partant de ce constat, il est donc primordial de ne pas négliger les activités de la phase « Test et recette » dans les projets. Une phase qui nécessite le plus d'attention.

Les enjeux pour l'entreprise sont nombreux : Lorsqu'un projet n'est pas défini correctement depuis le début, cela peut conduire à une grande perte pour l'entreprise; de très lourdes conséquences sur le plan financier ou plus grave sur le plan humain. Selon une étude américaine, c'est près de soixante milliards de dollars qui sont perdus chaque année aux Etats-Unis suite à des problèmes de qualité dans les applications logicielles¹.

A l'heure actuelle, les organisations de développement informatiques, commencent à prendre conscience de l'importance de cette phase dans le management du projet, désormais elle est prévue dans les plannings : définir une stratégie de recette fonctionnelle et la mettre en œuvre, peut assurer que le projet reste dans la bonne direction. Cependant ces organisations de développement sont obligées de faire des choix des méthodes et des outils de tests, et faire appel à des professionnels pour mettre en œuvre cette stratégie de recette. Il est indispensable d'associer un outil pour chaque méthode et chaque activité.

Ces méthodes et ces outils de test évoluent à grands pas ces dernières années, de nombreux outils ont été conçus pour aider les Consultants en informatique où Prestataires de services à effectuer leurs démarches.

Après le cycle V, les méthodes agiles qui changent les étapes de la conception, de la programmation et de la vérification. En effet une offre considérable des tendances de test logiciel sur le marché, une grande concurrence qui a une seule et même finalité : celle de « garantir la qualité des livrables ».

Ce mémoire, donc, présentera cette thématique «Test et recette fonctionnelle dans les système d'informations». Il fournit une étude comparative et d'analyse, dans le but de sensibiliser aux enjeux de la phase recette, aux méthodes et les outils à mettre en œuvre, en mettant l'accent sur les tendances et les nouveaux outils de la gestion du patrimoine de tests. Se pose alors la problématique suivante :

Qu'est-ce que la recette fonctionnelle ? Quelle méthode et quel outil choisir ? Faut-il abandonner les méthodes classiques et passer à l'agilité ? Les nouvelles pratiques répondent-elles aux besoins des testeurs? Quel apport et quelles sont les difficultés ?

¹ [NIST2002] TASSEY G., mai 2002, The Economic Impacts of Inadequate Infrastructure for Software Testing, Rapport du NIST (National Institute of Standards & Technology).

Ce mémoire s'adresse aux consultants en informatique et à tous ceux qui participent à la conception des logiciels, qu'ils soient gestionnaires, développeurs, chefs de projet ou qu'ils soient « côté métier » initiateurs et futurs utilisateurs de ces logiciels, dans le but d'attirer leurs attention sur l'importance de la phase test et sur les bonnes pratiques visant à améliorer les processus de test et de validation des logiciels.

Sur la base de mon expérience au stage, ce mémoire fournit un guide stratégique qui livre une série d'informations sur :

- L'organisation et la planification de la recette fonctionnelle,
- Des outils de gestion permettant d'organiser, structurer toutes les phases de test,
- Des outils d'automatisation de test permettant d'effectuer les tests qui sont impossible à effectuer manuellement (tests unitaires, fonctionnels, techniques)
- Des méthodes par exemple Agile qui ont tendance à prendre la place des méthodes classiques,
- De nouveaux acteurs spécialisés dans la qualité logicielle,
- Des professionnels d'Assistance à la Maîtrise d'Ouvrages -(Cabinets de conseil)- qui proposent des services en la matière,
- Toute l'actualité sur l'activité de test : les organisations professionnelles, les nouvelles tendances tests logiciels disponibles sur le marché, leurs apports, ainsi que leurs difficultés.

Ce sont les grands axes et les éléments essentiels construisant ce mémoire en trois parties :

Dans un premier temps, nous allons commencer par présenter l'organisation du Cabinet de conseil et de formation en informatique « SALIHANNE-CONSULTING² », où s'est déroulé le stage ainsi que la mission effectuée durant ces trois mois.

Ensuite nous donnerons des généralités sur un projet informatique : les acteurs MOA (la maîtrise d'ouvrage) et MOE (la maîtrise d'œuvre), une introduction à la recette fonctionnelle, aux tests logiciel et nous terminerons par présenter les bugs et les conséquences désastreuses.

La deuxième partie est consacrée aux méthodes et outils permettant d'améliorer l'activité de test. Cette partie est composée de deux grands points :

-L' historique et l'évolution des méthodes avec une étude d'analyse et de comparaison entre : (les tests manuels et les tests automatisés), et (le cycle en « V » et la méthode

² Cabinet de conseil et de formation en systèmes et logiciels informatiques
<https://www.salihanne-consulting.com/>

agile), nous ferons un point sur les avantages et des inconvénients de chaque méthode et chaque outil.

-Ensuite « la veille concurrentielle et technologique » : un projet qui a été mis en place pendant la mission de stage au sein du Cabinet de conseil et de formation en informatique « SALIHANNE-CONSULTING ». Nous présenterons aussi le processus complet de la veille qui a été déployé dans le but d'avoir toute l'actualité sur les nouvelles pratiques de test.

Cette démarche s'appuie sur des éléments nécessaires pour surveiller tout ce qui se dit quotidiennement sur ce le sujet « test et recette dans les projets » : sur la presse spécialisée, des publications sur les blogs des professionnels de la recette fonctionnelle, surveiller également les événements organisés par les experts du domaine et les acteurs des solutions logiciel.

Cette étude révèle ensuite, à travers deux panorama détaillés : le premier concerne des cabinets de conseils en informatique et les outils qu'ils proposent, le deuxième concerne les référentiels du test disponibles sur le marché, suivis de deux tableaux comparatifs pour démontrer la différence des fonctionnalités et des enjeux de chaque outil.

La troisième partie, consiste à présenter une démarche sur l'organisation et la planification de la recette fonctionnelle.

Cette présentation, s'appuie sur mon expérience de stage suite ma participation à l'organisation et la planification de la stratégie de test : depuis l'élaboration de cahier de recette jusqu'à l'établissement du PV de recette, une méthodologie basée sur un l'outil « Quality Center³ » de Hewlett Packard, celui le plus pratiqué ; il permet d'optimiser le processus de recette, et assure le bon fonctionnement des applications avant qu'elles soient mises en production.

Nous terminerons par une présentation des fonctionnalités et les avantages de cet outil illustrés sur des captures d'écrans de l'outil « Quality Center ».

Observation, surveillance, comparaison et analyse, investigation, témoignages, des exemples tirés du terrain : ce sont les pistes de réflexions et la méthodologie choisie pour argumenter notre travail et fournir des informations nécessaire pour couvrir tout le domaine et la problématique abordée.

³ HP, Quality Center : un outil de gestion des tests

Première partie :
Prestation du métier d' MOA de
Recette Fonctionnelle

Le projet informatique est, certes, un projet technique, mais aussi et surtout un projet humain, cependant, personne n'est à l'abri d'une erreur. Pour cela la recette est indispensable avant la mise en production, elle permet de réduire les risques d'apparition d'anomalies et s'assurer que le produit final correspond à ce qui a été demandé. Si la recette est mal réalisée des dysfonctionnements ne seront pas repérés, et les résolutions seront longues et coûteuses.

En effet, dans cette première partie nous vous présenterons des généralités sur un projet informatique: les acteurs principaux ; MOA (maîtrise d'ouvrage), MOE (maîtrise d'œuvre), les responsabilités et les rôles principaux de chacun.

Ensuite une introduction à la « Recette fonctionnelle » et aux « tests logiciels », en mettant l'accent sur les enjeux majeurs de cette phase dans un projet, dans le but de sensibiliser à l'importance des tests, aux risques, aux coûts et les conséquences des bugs.

Nous devons répondre aux questions suivantes :

- Qu'est-ce qu'un projet informatique ?
- Quels sont les acteurs principaux ?
- Qu'est-ce qu'une recette fonctionnelle ?
- Quels sont ses grands axes ?
- Comment définir l'activité de test dans un projet logiciel ?
- Qu'est-ce qu'un Bug informatique?
- Quelles sont les conséquences désastreuses?

Avant de répondre à ces questions, nous allons vous présenter le cabinet de conseil « Salihanne-Consulting », l'organisation où j'ai effectué ma mission de stage d'une durée de douze semaines, le premier point évoqué dans la première partie ci-dessous,

1 Présentation de la société « Salihanne-Consulting »

Salihanne-Consulting, est une entreprise spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques.

-L'entreprise est en activité depuis le 01 Novembre 2013,

-Domiciliée à GARGES LES GONESSE (95140), 29 BOULEVARD DE LA MUETTE,

-Enregistrée sous le numéro de SIREN : 799 376 157 et SIRET : 799 376 157 00019,

-Code APE / NAF : 6202A, activité : Conseil en systèmes et logiciels informatiques⁴



Figure 1: Cabinet de Conseil Salihanne-Consulting⁵

L'entreprise Salihanne Consulting (Cabinet de conseil), s'investit dans la formation de ses consultants, de sorte à perfectionner leurs compétences et à leur apporter des méthodes de travail innovantes pour les aider au mieux à atteindre leurs objectifs et à assurer la réussite de chaque mission.

Les consultants du cabinet interviennent, dans divers secteurs d'activités :

- Banque,
- Assurance,
- Énergie,
- Télécom,
- Industrie.

Ces consultants, ont des compétences aussi bien fonctionnelles que techniques : MOA, AMOA, MOE.

Ils interviennent à toutes les phases de déploiement du projet dans toute la France ; pour conseiller et guider ses clients dans leurs choix stratégiques, en offrant des solutions sur mesure pour répondre à leurs besoins spécifiques ainsi qu'une expertise permettant la réussite de chaque projet.

⁴ <http://www.verif.com/societe/SALIHANNE-CONSULTING-799376157/>

⁵ Source : <https://www.salihanne-consulting.com/>

1.1 Les services du cabinet « Salihanne-Consulting »

Le cabinet de conseil et de formation en informatique, Salihanne consulting est spécialisé dans plusieurs domaines :

1.1.1 Intégration, Paramétrage ERP

Afin d'améliorer leur rendement, les organisations doivent rechercher le meilleur moyen de gérer leurs ressources avec le plus d'efficacité. Salihanne-Consulting offre des services de gestion intégrés (ERP) qui aideront ses clients à rentabiliser au maximum leurs investissements et leurs activités. Le cabinet a pour mission de mettre en œuvre, intégrer et paramétrer des progiciels ERP comme ceux d'Oracle ou SAP. Après une étude de l'entreprise cliente, les consultants du cabinet proposeront la solution la plus adaptée et dans les meilleurs délais.

1.1.2 Développement sur mesure d'applications web

Le groupe Salihanne-Consulting, étudiera avec ses clients leur projet d'application web ainsi que son contexte concurrentiel afin de déterminer au mieux son positionnement stratégique et leur conseiller les solutions techniques les plus adaptées, en offrant les meilleures performances possibles.

1.1.3 Mobilité et Telecom

Les applications mobiles sont un secteur en évolution constante avec des spécifications qui changent au jour le jour. Que ce soit une application visant à augmenter la productivité de l'entreprise ou une application mobile visant les utilisateurs. Les experts du cabinet, sont capables d'étudier avec ses clients le potentiel commercial de leur application et les conseiller efficacement sur le choix de la solution de développement et les plateformes à envisager.

1.1.4 Assistance à la maîtrise d'ouvrage

Les consultants du cabinet Salihanne-Consulting, offrent pour ses clients une aide en tant qu'experts afin de remettre le développement de leur projet sur les rails. Ils assisteront pour finaliser la maîtrise d'ouvrage et les étapes qui suivent afin d'assurer que le projet reste dans la bonne direction en définissant avec les clients une recette fonctionnelle et/ou en participant également à la phase de Testing.

1.1.5 Maitrise d'ouvrage

La maîtrise d'ouvrage est une étape importante dans le cycle de vie de tout projet puisqu'elle définit dès le départ l'objectif du projet aussi le budget qui lui est alloué. Elle maîtrise l'idée de base du projet et représente ce que les utilisateurs finaux à qui l'ouvrage, le projet final est destiné. En gros la maîtrise d'ouvrage est responsable de l'expression

fonctionnelle des besoins concernant le projet. C'est dans cette optique que les experts du cabinet assistent ses clients de la manière la plus efficiente possible.

1.1.6 Réseaux

De nos jours les réseaux sont des outils importants dans la vie d'une entreprise. Qu'il s'agisse d'intranet ou d'extranet, les consultants du cabinet offrent les meilleurs services et suivis techniques à la pointe de la technologie et font une analyse préliminaire sur les installations, afin de permettre aux entreprises de booster leur potentiel.

1.1.7 Maitrise d'œuvre

Les experts du cabinet ont des capacités et des connaissances approfondies en informatique (le développement, réseau, base de données, serveurs), pour étudier le potentiel commercial d'une application et conseiller efficacement sur le choix de la solution de développement et les plateformes à envisager.

1.2 La mission de stage

J'ai intégrée l'équipe des consultants du cabinet de conseil « Salihane-consulting » en début de juillet 2014. J'étais chargée de la mise en place d'un projet de la veille concurrentielle sur les outils et méthodes de la recette fonctionnelle, un processus de veille a été déployé pour avoir toute l'actualité de l'activité de « test logiciel», nous vous présenterons ce projet en détail dans la deuxième partie de ce mémoire.

J'ai également participé à la définition d'une méthodologie de tests basée sur l'outil HP « Quality Center » et sa mise en œuvre sur plusieurs applications. Cela m'a permis d'acquérir des connaissances sur la pratique de cet outil (ses fonctionnalités, ses étapes et ses avantages).

La capacité du cabinet de conseil « Salihane-Consulting » a répondu favorablement aux clients au travers de ses méthodes, de son savoir-faire, les propositions techniques, et de ses compétences. Ses clients ont bénéficié d'une application parfaitement fiable à l'issue de la phase de recette.

2 Les acteurs du projet informatique: MOA et MOE

En générale, la réussite d'un projet informatique repose sur ces deux acteurs : MOA (maîtrise d'ouvrage), MOE (maîtrise d'œuvre) : le premier décrit ce qu'il faut faire, et le second le met en œuvre, cela veut dire que les responsabilités et les missions sont différentes.

2.1 Définition d'un projet

« Un projet est un ensemble unique d'activités, contrôlées et coordonnées, avec des dates de début et de fin, effectuées avec pour objectif de conformité à des exigences spécifiques, incluant des contraintes de temps, de coût et de ressources » [1, GLOSSIARE, CFTL/ISTQB, p 51].

2.2 Equipe MOA (la maîtrise d'ouvrage)

La maîtrise d'ouvrage (MOA) est composée d'un chef de projet MOA et une équipe de collaborateurs chargés de participer au projet jusqu'à sa mise en œuvre ; des experts métier, dans le domaine professionnel pour lequel un projet est monté.

La recette fonctionnelle se situe dans cette partie de l'organisation. Aussi, ce sont tant les équipes qu'il faudra défendre en priorité qu'apporter conseil lorsque divers points des documents semblent ambigus. Le pilote de recette sera amené régulièrement à être en relation avec les équipes de recueil de besoins des utilisateurs, du cahier des charges, des spécifications fonctionnelles que de l'assistance aux utilisateurs. C'est le projet qui compte avant tout.

La MOA est responsable de :

- La définition et description de besoins des utilisateurs: La maîtrise d'ouvrage est l'entité responsable de la description du « QUOI »:
L'équipe MOA doit expliquer clairement le résultat qu'ils souhaitent obtenir sur un plan purement fonctionnel, en rédigeant un document appelé "expression du besoin".
- L'identification du budget et du retour sur investissement «ROI»: C'est donc la maîtrise d'ouvrage qui définit les budgets à utiliser pour réaliser le projet demandé en calculant ROI positif.
- La responsabilité de la définition fonctionnelle: Pour toute décision relative à la définition fonctionnelle du projet (règle de gestion, ...). La MOA, doit valider toutes les descriptions fonctionnelles concernant son outil.

- L'accompagnement au changement: Accompagner le changement auprès des utilisateurs du projet, est une étape pas facile pour la MOA, car la bonne réalisation de cet accompagnement dépend la réussite globale du projet, dans le cas contraire - (logiciel mal fabriqué)-, il sera rejeté par l'utilisateur final.

2.3 Equipe MOE (la maîtrise d'œuvre) :

La maîtrise d'œuvre (MOE) est composée d'un chef de projets MOE qui pilote une équipe d'informaticiens responsables du "COMMENT" :

Les équipes MOE sont dans une autre partie de l'organisation. Elles ont des connaissances approfondies dans le développement, réseau, base de données, serveurs, etc.

Cependant, il est nécessaire non pas de s'opposer, mais au contraire d'être complémentaire, pour qu'elle puisse réaliser les projets dans des objectifs de délais et de qualité.

Le pilote de recette sera amené régulièrement à être en relation avec les équipes d'intégration, d'homologation, de développement etc.

Bien souvent, ceci contribuera à ce que les équipes MOE aient une correcte compréhension du besoin, ce qui est le meilleur garant du fonctionnement correct de l'application. Réciproquement, être à l'écoute de la MOE permet d'informer la MOA des informations de terrain comme la faisabilité ou du cout d'une évolution vis-à-vis de son intérêt réel.

La MOE est responsable de :

- La compréhension du besoin : Comprendre le besoin de la maîtrise d'ouvrage sans aucune ambiguïté est la première responsabilité de la MOE, en rédigeant un cahier des charges détaillé qui reprend chaque besoin exprimé par la maîtrise d'ouvrage.
- Le devoir de conseil : La maîtrise d'œuvre a un devoir d'expliquer les points de risques identifiés dans les besoins énoncés par la maîtrise d'ouvrage : lever les alertes, si possibles au travers de documents formalisés (notes, ...).
- L'adaptation aux métiers : Il arrive souvent que la maîtrise d'œuvre travaille sur plusieurs projets issus de métiers différents, pour cela Les informaticiens doivent avoir des capacités d'adaptation aux nouveaux métiers, par exemple, s'adapter au vocabulaire et au langage d'un nouveau domaine.
- Les choix technologiques: La maîtrise d'œuvre doit faire un bon choix du : langage de programmation, d'architectures, de réseau, de bases de données, etc.
- La mise en œuvre technique : Il s'agit de la mise en œuvre de tous les moyens technologiques nécessaires (code informatique, des bases de données, etc.
- La qualité et la conformité des réalisations : L'équipe de la maîtrise d'œuvre doit réaliser des tests pour que les applications livrées sont conformes aux spécifications validées au début du projet et sans bugs.

Conclusion : Ainsi la réalisation de tout projet informatique, dépend de la complémentarité, la compréhension, et l'entente de ces deux acteurs MOA, MOE ; chacun d'eux a une mission et une responsabilité bien précise et qu'il doit respecter et accomplir dans les règles de l'art.

3 Introduction à la Recette Fonctionnelle :

La recette fonctionnelle est une activité relative au domaine des tests, donc à la qualité logicielle. Elle a la particularité de se situer dans les équipes de la MOA et de se réaliser lors de la réception des livrables de la MOE. Le périmètre doit tendre vers la globalité de l'application et surtout elle doit simuler les actes les plus probables qu'effectueront les utilisateurs.

Comme tout type de test, la recette s'effectue dans un objectif de « juste qualité ». En effet, réaliser la complétude des tests relève d'une cardinalité qui tend vers l'infini, la « sur-qualité » à un coût tant humain que financier.

3.1 Définition de la recette :

« Opération formelle, réalisée par le client, pour vérifier la conformité du produit fourni (composants logiciels spécifiques, progiciels, prestations et documentation) par rapport à la spécification validée (plus, éventuellement, les modifications acceptées). Cette vérification se fait sur la base d'un dossier de recette et se termine par un procès-verbal de recette qui formalise l'acceptation du client ». [2, DRECQ Vincent].

Une étape très importante dans un projet informatique, mais très délicate ; il arrive souvent des tensions entre le maître d'œuvre et le maître d'ouvrage et les testeurs.

Parfois la MOA propose des solutions techniques pour la MOE alors que ce n'est pas son rôle, parfois la MOE intervient sur les choix fonctionnels, et surtout reproche à la MOA son implication superflue en pensant que projet informatique est une affaire des informaticiens.

Des problèmes sont souvent le fait de la méconnaissance de l'objet de la recette et des techniques associées. Le schéma ci-dessous, représente le Model-Based Testing (MBT)⁶ : piloter la production et la maintenance des tests fonctionnels et d'acceptation à partir d'une modélisation de l'expression de besoins : processus et règles métier.

⁶ Source : Magazine IT-expert, L'alignement de l'IT avec les besoins Métier, publié, le 11 juillet 2013, par Aurelie Magniez
<http://www.it-expertise.com/lalignement-de-lit-avec-les-besoins-metier/>

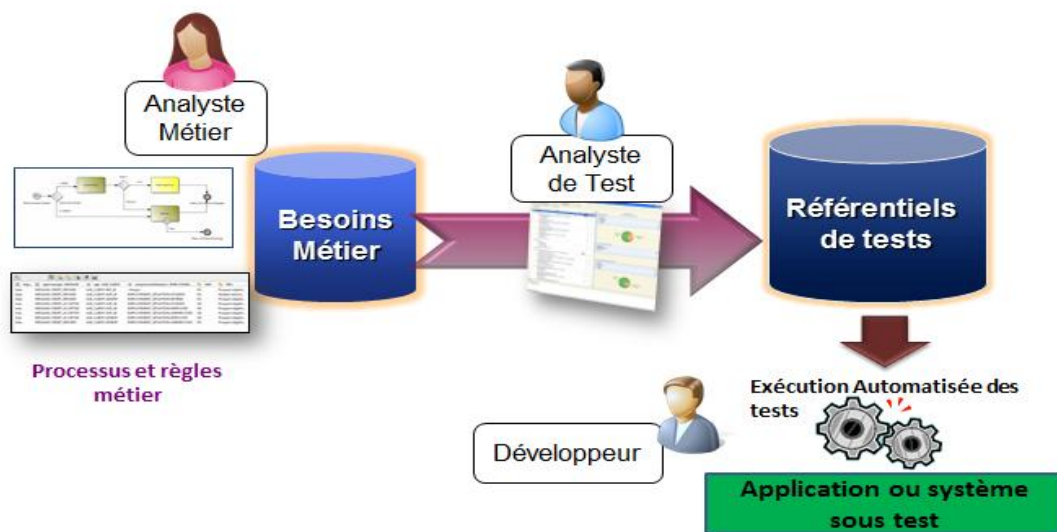


Figure 2: Processus du Model-Based Testing

3.2 Les grands axes de la recette fonctionnelle :

Pour garantir un niveau de qualité optimale de l'application livrée, l'équipe "Salihanne consulting" s'appuie sur : 4 Axes majeurs :

- Le plan de tests
- L'exécution des tests
- Le suivi de la recette
- La communication

Et sur 4 orientations :

- Fonctionnelle : Respect des exigences métiers issues des spécifications
- Technique : respect des exigences de performance et du bon fonctionnement du système.
- Documentaire : cohérence de la documentation produite par rapport aux éléments livrés.
- Normative : respect des normes du projet

4 Introduction au test logiciel

Toute fabrication de produit suit les étapes suivantes :

- Conception
- Réalisation
- Test

« Le test », est un sujet vaste et complexe : les applications sont variées, les outils sont différents, cependant, les tests doivent être organisés et menés selon une démarche.

4.1 Qu'est-ce que tester ?

Il existe plusieurs définitions sur l'activité de test logiciel :

« Le test d'un logiciel est une activité qui fait partie du processus de développement. Il est mené selon les règles de l'assurance de la qualité et débute une fois que l'activité de programmation est terminée. Il s'intéresse aussi bien au code source qu'au comportement du logiciel. Son objectif consiste à minimiser les chances d'apparitions d'une anomalie avec des moyens automatiques ou manuels qui visent à détecter aussi bien les diverses anomalies possibles que les éventuels défauts qui les provoqueraient ». [3, FELIX Patrick, p17]

4.2 Les principes de base de l'activité « test »

Principe N°1 : Impossible de tout tester :

Dans une application informatique, il n'est pas possible de tout tester. Cependant, avant de tester une application, il faut bien s'assurer ce test couvre les exigences fonctionnelles et techniques également. Il a été démontré scientifiquement, que la preuve du zéro défaut est un problème indécidable. « Il est donc impossible de garantir l'absence de défaut dans les logiciels ». [4, SEVRE, p21].

Cependant il est important de définir une stratégie de test et analyser les risques.

Principe N°2 : Mise en évidence de défauts:

Pour tester correctement un logiciel il faut des connaissances du contexte, de l'intelligence et de la créativité, il ne faut jamais croire que c'est simple et qu'il n'y a pas d'erreur. Si aucun défaut n'est découvert, ce n'est pas une preuve qu'il n'en reste pas.

Principe N° 3 : Tester le plus tôt possible :

La planification sérieuse des tests dans le planning est indispensable à la maîtrise du projet, cela nous permet d'effectuer les tests le plus tôt possible dans le cycle de développement du logiciel. Il a été prouvé que des vérifications et des corrections effectuées au moment des exigences sont moins coûteuses que celles effectuées au moment du déploiement de l'application. « Les études menées sur le coût associé à la détection d'une erreur montre que si une erreur décelée lors de la phase d'élaboration du cahier des charges coûte 1 alors que la même erreur décelée en phase de conception coûte 10 et une erreur décelée en phase d'exploitation coûte 100». [4, SEVRE, p20].

Principe N°4 : Les tests dirigés par les risques :

Analyser les risques, et les points de vigilances de l'activité « test », nous permet d'être attentifs aux enjeux économiques de l'application à tester. Il est indispensable de définir la priorité de chaque fonctionnalité et prendre en compte le dysfonctionnement et son éventuel impact sur l'organisation cible.

Principe N°5 : Assurance qualité (AQ) logiciel

La présence d'un tiers de confiance (AQ logiciel) est également indispensable pour garantir la qualité du logiciel. Le développement n'est pas le seul responsable et juge de l'évaluation honnête de la qualité du logiciel : « L'assurance qualité permet de mettre en œuvre un ensemble de dispositions qui vont être prises tout au long des différentes phases de fabrication d'un logiciel pour accroître les chances d'obtenir un logiciel qui corresponde à ses objectifs (son cahier des charges) », [3, FELIX p14].

Conclusion : cet ensemble de principes de la vérification des produits logiciel permet les résultats suivant :

- La maîtrise des besoins et les risques métiers.
- Garantir la qualité du produit.

5 Les Bugs et les conséquences

Le problème fondamental du développement des logiciels est « le problème de l'erreur ». De nombreux défauts et anomalies pouvant affecter un logiciel.

En effet, dans cette partie, nous porterons notre attention sur les sources des bugs logiciels, ensuite les niveaux de la gravité des anomalies et nous terminerons par quelques exemples de bugs et leurs conséquences.

5.1 Qu'est-ce qu'un Bug ?

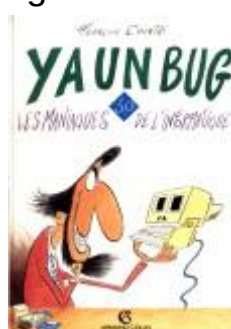


Figure 3: Le BUG Informatique⁷

Est-ce si grave ? D'où vient l'erreur ? Quels sont les conséquences désastreuses ? Y'a-t-il un moyen de chasser les bugs ?

Le terme BUG est utilisé pour désigner: Défaut => Erreur => Anomalie.

Défaut: « Une imperfection dans un composant ou un système qui peut conduire à ce qu'un composant ou un système n'exécute pas les fonctions requises, par exemple une instruction ou une définition de données incorrecte. Un défaut, si rencontré lors de l'exécution, peut causer la défaillance d'un composant ou d'un système ». [1, Glossaire CFTL/ISTQB, p 25]

Erreur: « action humaine produisant un résultat incorrect ». [1, Glossaire CFTL/ISTQB, p29]

Anomalie: « toute condition qui dévie des attentes basées sur les exigences de spécifications, documents de conception, documents utilisateurs, standards etc., ou des perceptions ou expériences de quelqu'un. Les anomalies peuvent être trouvées pendant, mais pas uniquement, les revues, tests, analyses, compilations ou utilisation des produits logiciels ou de la documentation applicable. [1, Glossaire CFTL/ISTQB, p11]

⁷ "Les victimes de l'informatique", Une BD de François Cointe chez Armand Colin-1990
<http://www.bedetheque.com/BD-Y-a-un-bug-Tome-1-Les-victimes-de-l-informatique-488.html>

5.2 Qualification de la gravité des anomalies :

Il existe plusieurs niveaux d'anomalies illustrés sur ce tableau ci-dessous :

Gravité	Description
Bloquante	= L'anomalie provoque un arrêt complet du système ou une fonctionnalité indispensable est inexploitable (pour des raisons fonctionnelles ou de performance, Aucune solution de contournement.  une anomalie bloquante doit faire l'objet d'un correctif rapide
Majeure	= Une fonctionnalité indispensable est partiellement inopérante sans bloquer l'exploitation de l'outil. L'application peut continuer.  une solution de contournement est identifiée, qu'il ne faut pas négliger.
Mineure	= Une fonctionnalité non essentielle présente des dysfonctionnements sans bloquer l'exploitation de l'outil.  La solution peut attendre et les dysfonctionnements seront traités en dernier.

Tableau 1: les différents niveaux d'anomalies

5.3 Les sources des Bugs logiciels

L'origine des bugs informatiques viennent des erreurs humaines tel que :

- Une incompréhension du besoin et des exigences de l'organisation commanditaire,
- Manque du savoir faire et d'expertise,
- Manque des compétences informatiques pour comprendre l'environnement de développement et de l'exécution,
La complexité des plates-formes,
- Une faible puissance des systèmes informatisés.

5.4 Exemples des Bugs aux conséquences désastreuses

Il existe de nombreux exemples de bugs informatiques qui ont empêché le fonctionnement correct d'un programme, en voici quelques exemples et les conséquences désastreuses :

Domaine aéronautique: exemple 1 : « la Mariner 1 » : [5, ZAITOUN, p5].

En juillet 1962, la seconde Mariner1 (NASA) perd l'itinéraire quatre minutes après son lancement.

La cause du problème: Erreur de transcription d'une équation ; une instruction du programme de guidage écrit en Fortran, contenait une virgule à la place d'un point.

Exemple 2 : « Ariane 5 » :

Le 4 juin 1996, Ariane 5, explose en Guyane lors de son premier vol, (Auto-destruction), 37 secondes après le décollage.

Cause du problème: la réutilisation du logiciel de guidage d'Ariane 4 dans Ariane 5 sans analyse des conditions de son fonctionnement ni tests sur simulateur, malheureusement ce le logiciel ne faisait pas certains tests de débordement (conversion flottant 64 bits trop grand vers entier 16 bits), Ariane 5 était beaucoup plus rapide...

Le domaine de la médecine : « Therac 25 » :

En 1985–1987, la machine Therac 25 chargée de traiter des patients atteints du cancer leur a administré une dose mortelle de radiations ce problème a causé 5 morts.

Cause du problème: Conflit d'accès aux ressources entre 2 parties logicielles.

Exemples récents en France :

- " TF1 Danse Avec les stars⁸ " : Après le bug du direct le 13 octobre 2014, TF1 va rembourser.

L'animateur de l'émission, "Vincent Cerrutti" annonce "un souci technique": Un bug informatique qui empêche de comptabiliser le vote du public. TF1 remboursera les téléspectateurs. Écrit par Benjamin Guy.

- Le Bug SFR et chiffres du chômage, en octobre 2013 :

Baisse du chômage, un bug chez SFR a faussé les chiffre: le ministre du travail, a reconnu que la forte baisse du chômage au mois d'août (50 000 demandeurs d'emploi) résultait en partie d'une panne chez l'opérateur SFR.

Conclusion : Mais enfin ya-t-il un moyen de détecter ces anomalies ? Quels sont les différents outils et les méthodes pour réaliser des tests des applications et détecter les défauts ?

Effectivement il existe des moyens pour détecter des défauts et réaliser des tests, ces moyens, peuvent être automatiques ou manuels. Lesquels ?

⁸ <http://www.closermag.fr/tele/dossier-danse-avec-les-stars/saison-4/news/danse-avec-les-stars-4-damien-sargue-destabilise-par-le-bug-technique-215886>

Nous répondrons à cette question dans la deuxième partie de ce mémoire consacrée aux « méthode et outils de test ».

Vous cherchez une solution de test logiciel open source,

Vous cherchez des professionnels dans la qualité des logiciels,

Vous vous intéressez aux tendances test logiciel,

Un projet de la veille sur cette activité a été mis en place dans la partie suivante

Deuxième partie :
Méthodes et Outils de Test,
Historique et Evolution

Cette partie est consacrée aux méthodes et outils existants qui peuvent aider les testeurs et les chefs de projets à améliorer leurs performances dans les projets d'applications informatiques. Une étude d'analyse et de comparaison entre :

- Les tests manuels et les tests automatisés,
- les cycles de développements,

Cette comparaison est illustrée par des schémas et un tableau comparatif.

L'objectif de cette partie est de mettre en lumière l'évolution de l'activité de test lors de la transition des méthodes traditionnelle vers les nouvelles pratiques, cela nous permet de découvrir :

- les outils disponibles sur le marché,
- les grands acteurs et éditeurs des solutions test logiciel,
- l'apport de l'automatisation des tests,
- la forte valeur ajoutée des méthodes agiles,

Nous nous sommes intéressés aussi aux défis organisationnels qui apparaissent lors des transitions vers ces nouvelles méthodes de développement logiciel, tels que : les nouveaux rôles, les nouvelles responsabilités introduites? Quel impact sur les testeurs et ces organisations ?

Et pour avoir l'actualité de tous les acteurs de ces nouvelles pratiques, nous avons aussi effectuée une veille concurrentielle et technologique, en mettant en place un dispositif, un processus de veille et tous les éléments nécessaires pour surveiller tout ce qui se dit quotidiennement sur ce le sujet du « test logiciel » ; consulter la presse spécialisée et les publications sur les blogs des professionnels de la recette fonctionnelle, surveiller aussi les événements organisés par les experts du domaine et les acteurs des solutions logiciel.

1 Les tests manuels et les tests automatisés

« Le test est un processus manuel ou automatique, qui vise à établir qu'un système vérifie les propriétés exigées par sa spécification, ou à détecter des différences entre les résultats engendrés par le système et ceux qui sont attendus par la spécification». [3, FELIX Patrick, p17]

Cette définition souligne deux catégories de test : ceux effectués manuellement (tests manuels) par le testeur et ceux rendus automatisés avec différents outils d'automatisation de tests (tests automatisés).

1.1 Les tests manuels : (Méthodes traditionnelles)

Cette phase, opérationnelle à proprement dit, est située dès la réception des livrables. Il est souhaitable d'insister sur le fait que plus la phase de préparation n'a été correctement réalisée, plus les tests manuels de la recette ne seront optimaux.

La méthode usuellement utilisée lors de cette phase contient les aspects qui suivent :

- L'exécution des tests
- La gestion des anomalies
- La rédaction de documents reflétant l'état des tests réalisés

1.2 Les tests automatisés : (Méthodes nouvelles) :

L'automatisation des tests est « l'utilisation de logiciels pour exécuter ou supporter des activités de tests, par exemple ; gestion des tests, conception des tests, exécution des tests ou vérification des résultats.». [1, Glossaire CFTL/ISTQB, p 67].

Ces objectifs principaux permettent théoriquement de jouer à volonté des tests de non régression suite à la livraison d'une nouvelle version d'une application.

L'automatisation d'un test n'a de sens que si le test répond à un certain nombre de critères :

- le test est systématique : il doit être exécuté à chaque nouvelle version de l'application
- le test est répétitif : il est présent dans de nombreux scénarios de test
- le test est automatisable : il est possible techniquement de faire jouer le test par un robot

1.3 Avantages et inconvénients des tests manuels et les tests automatisés

Le principe de ces deux catégories de tests (manuels ou automatisés) n'est pas le même, des différences dans méthodes de réalisation, des difficultés dans l'utilisation, les deux représentent des avantages et des inconvénients. Le tableau comparatif ci-dessous illustre toutes ces différences :

Type de test	Avantages	Inconvénients
<u>Tests manuels :</u> Les tests manuels sont exécutés manuellement par un testeur d'après un plan de test donné : le testeur entre les données de test via une interface; lance les tests; observe les résultats et les compare avec les résultats attendus. En cas d'erreurs, le testeur réalise une analyse qui fera l'objet d'une fiche d'anomalie si nécessaire	-Pouvoir commencer avant que l'application soit parfaitement stable. -Tester en parallèle des opérations de corrections. -Créer des scripts de test simples rapidement. -Implication souvent plus facile d'utilisateurs totalement néophyte dans la démarche de test.	-L'impossibilité de respecter les délais -Une activité laborieuse, longue, coûteuse et peu reproductible. -La répétition des mêmes tests devient très pénible pour le testeur. -Le temps consacré aux tests de non-régression est significatif -lorsque des applications doivent fonctionner sur plusieurs plates-formes, la charge des tests manuels croît proportionnellement en multipliant les risques d'erreurs humaines et d'incohérences. « livre blanc, Borland »
Test automatisé : contrairement aux tests manuels, le support d'outils qui décharge le testeur. Dans ce cas de test, le lancement des tests; l'enregistrement des résultats; se fait de façon automatique	-L'automatisation des tests permet : -d'accélérer les cycles de validation et optimisent la qualité logicielle. -se libérer des tâches répétitives -en cas des tests de régression -La fiabilité puisque ces tests peuvent vérifier toujours aussi consciencieusement les résultats, même après une énième exécution, ce qui est loin d'être le cas avec un humain.	-L'automatisation n'est pas adaptée pour tous les tests. -L'entreprise doit donc prévoir un projet « pilote » afin de se rendre compte si les outils d'automatisation sont en adéquation avec ses attentes et ses logiciels. - l'automatisation est coûteuse par rapport aux tests manuels et le retour sur investissement n'est pas immédiat. - les cas de tests automatisés devront alors être maintenus. Ceci engendre forcément une charge, un coût de test supplémentaire

Tableau 2: tests manuels et tests automatisés: Avantages/Inconvénients

Conclusion

Après cette étude comparative des deux catégories de test nous constatons que :

Les inconvénients des tests manuels empêchent de réaliser des tests minutieux et laissent passer les bogues-pouvant parfois s'avérer critiques à l'utilisation.

Un apport considérable de l'automatisation : un niveau de maturité technologique, organisationnelle, budgétaire, et bien évidemment la réduction des risques et des incidents en production. Pour cela nous sommes intéressés plus profondément à ce sujet d'automatisation des tests ; nous vous montrerons quelques témoignages sur l'apport et multiples bénéfices de l'automatisation résumés comme suit :

1.4 L'apport de l'automatisation

Aujourd'hui les organisations de développement logiciel sont face à un grand challenge:

[6, BORLAND]

- Améliorer la qualité des applications complexes,
- Accélérer les délais et avec des budgets serrés,
- Fournir des solutions informatiques « gagnantes ».

Pour faire face à ce grand challenge, ces organisations de développement, ont choisi l'automatisation du processus de test pour optimiser la qualité des logiciels

Y a-t-il un apport considérable/une forte valeur ajoutée par rapport au test manuels? Quelle sont les meilleures pratiques ? Est-ce que l'automatisation répond-elle généralement ?

L'automatisation des tests fonctionnels permet de multiples avantages :

Constituer une bibliothèque de ressources : l'expérience démontre que l'automatisation permet de réemployer des tests déjà existants sans en recréer à chaque nouveau projet. Et les tests réemployables sont davantage utilisés et permettent aux équipes de corriger un plus grand nombre d'erreurs et de constituer une bibliothèque de ressources.

La cohérence: Lors de la réutilisation des tests, les équipes d'assurance qualité accèdent à de nouveaux seuils de cohérence. L'automatisation permet la cohérence au processus de test, grâce à une procédure reproductible de documentation des résultats permettant de recréer et vérifier les erreurs pour accélérer leur résolution

Des gains de productivité: L'automatisation permet au département d'assurance qualité, de lancer des tests sans surveillance, et valider simultanément le bon fonctionnement d'une application sur plusieurs plates-formes, navigateurs et environnements. Cette opération libère des ressources pour se focaliser sur d'autres aspects de la qualité. Ces gains de productivité permettent de raccourcir les cycles de test et améliorer la qualité logicielle.

1.5 Les facteurs clés des tests fonctionnels automatisés :

Dans son blog du test logiciel industrialisé, Dalisys⁹, souligne 5 facteurs clés de succès d'une démarche d'automatisation des tests fonctionnels :

Bien prendre le temps de formaliser un plan de test. C'est notamment l'une des clés pour s'assurer de la bonne implémentation des exigences de test.

Décider des périmètres éligibles à l'automatisation sur des critères valables. Cette démarche permettra de maximiser la couverture des tests à automatiser.

Automatiser les tests sans avoir à scripter.

Maîtriser les outils et techniques permettant d'exploiter un maximum de données d'entrée de tests de façon automatisées.

Automatiser un maximum de contrôle des résultats attendus.

Certes aujourd'hui un grand nombre d'entreprises modernes ont réussi à accélérer les délais de développement et réduire les coûts de livraison d'applications de haute qualité grâce à l'automatisation des tests fonctionnels. Mais cela n'empêche que ces entreprises rencontrent quelques difficultés. En effet, le choix d'un d'outil, ne suffit pas : il faut aussi aborder tous les aspects de cette démarche d'automatisation, d'après des études réalisées, 85 % des entreprises échouent dans cette démarche d'automatisation pour des raisons suivantes :

- Manque d'investissements initiaux pour planifier un processus de développement logiciel,
- Le personnel n'est pas formé,
- L'environnement n'est pas adapté. [11, DALISYS].

⁹ Dalisys : Une entreprise spécialisée en Tests logiciels
<http://www.dalisys.com/fr/une-equipe-d-experts-en-tests-logiciels>

2 Modèles de cycle de développement des logiciels (l'évolution)

« Le Cycle de vie logiciel : Une période temporelle qui commence lorsque un produit logiciel est conçu et se termine lorsque le logiciel n'est plus disponible à l'usage. Le cycle de vie logiciel inclut typiquement une phase de mûrissement, une phase d'exigences, une phase de conception, une phase d'implémentation, une phase de test, une phase d'installation et livraison, une phase d'opération et de maintenance, et parfois une phase de retrait. Note : ces phases peuvent se recouper ou être exécutées de façon itérative ». [1, Glossaire CFTL/ISTQB, p 61].

2.1 Petit historique

Certains pensent que les origines des méthodes remontent aux années 60. A l'époque l'informatique existait déjà sur les premiers ordinateurs dits « scientifiques » pour l'armée et certaines branches de la recherche. Par contre les entreprises se contentaient de matériel dit « classique » qu'il fallait câbler physiquement. "J'ai vécu professionnellement cette époque et je peux assurer qu'aucun chef de projet en informatique de gestion ne s'occupait de méthode". [7, Vickoff, p4]

Ensuite, dans les années 70, l'arrivée de l'approche « cascade » caractérisée par un cycle en V qui a marqué le début des grands développements de systèmes d'informations.

Vers la fin des années 80, les communications de James Martin (Enseignant à l'Université d'Oxford), annonce les fondements du développement Itératif-Incrémental-Adaptatif, la base des approches Agiles actuelles.

Et en 1991, publication de la première méthode agile, RAD « Rapid Application Développement » sous la forme d'un guide complet de mise en œuvre. (Figure 4 ci-dessous).

Cette méthode est largement diffusée à l'échelle mondiale. Le passage à l'agilité gagne plusieurs directions informatiques : Quelle différence entre le cycle en « V » et le cycle itératif ? Quels sont les facteurs de réussite des projets agiles ?

Avant de présenter la méthode agile, ses valeurs et ses principes -cités dans le Manifeste agile-, nous allons commencer par définir le cycle en « V », car il s'agit du modèle le plus couramment utilisé, et particulièrement chez « Salihanne-Consulting » lieu de déroulement de ma mission de stage.

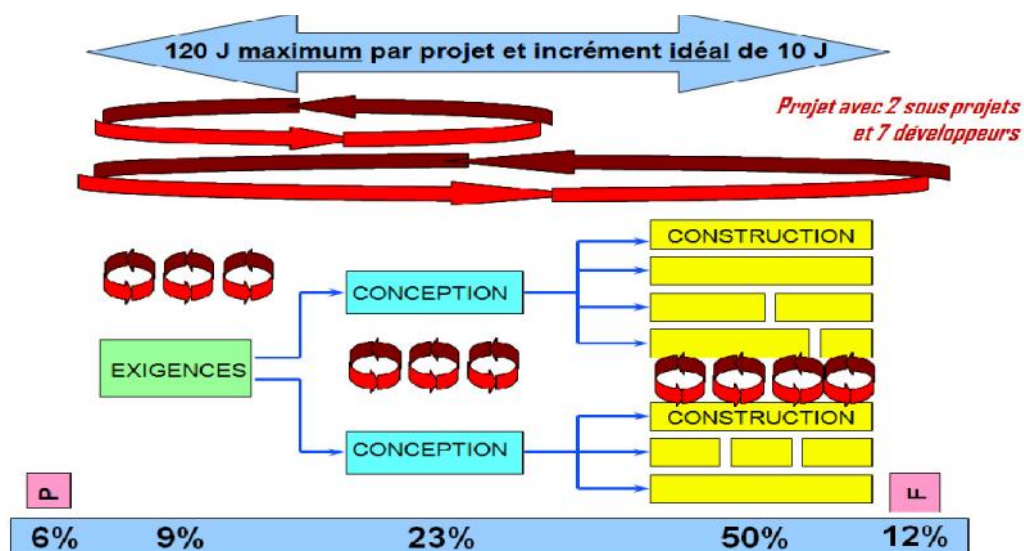


Figure 4 : Méthode RAD¹⁰

2.2 Le cycle en « V » ou Cascade

Le cycle en « V », est une « Structure décrivant les activités du cycle de développement logiciel, depuis la spécification des exigences jusqu'à la maintenance. Le modèle en V illustre comment les activités de test peuvent être intégrées dans chaque phase du cycle de développement ». [1, Glossaire CFTL/ISTQB, p78]

Il est représenté par un V dont la branche descendante contient toutes les étapes de la réalisation du logiciel, et la branche montante toutes les étapes de tests du logiciel, la pointe du V, quant à elle, représente la réalisation concrète du logiciel, le codage :

Chaque étape de conception correspond une étape de test,

Les tests peuvent intervenir tout au long du processus de développement du logiciel.

Nous pouvons voir ci-dessous le cycle en V (figure 5 ci-dessous) :

¹⁰ Vickoff Jean-Pierre, Méthode Agile – Historique et évolution, p 8

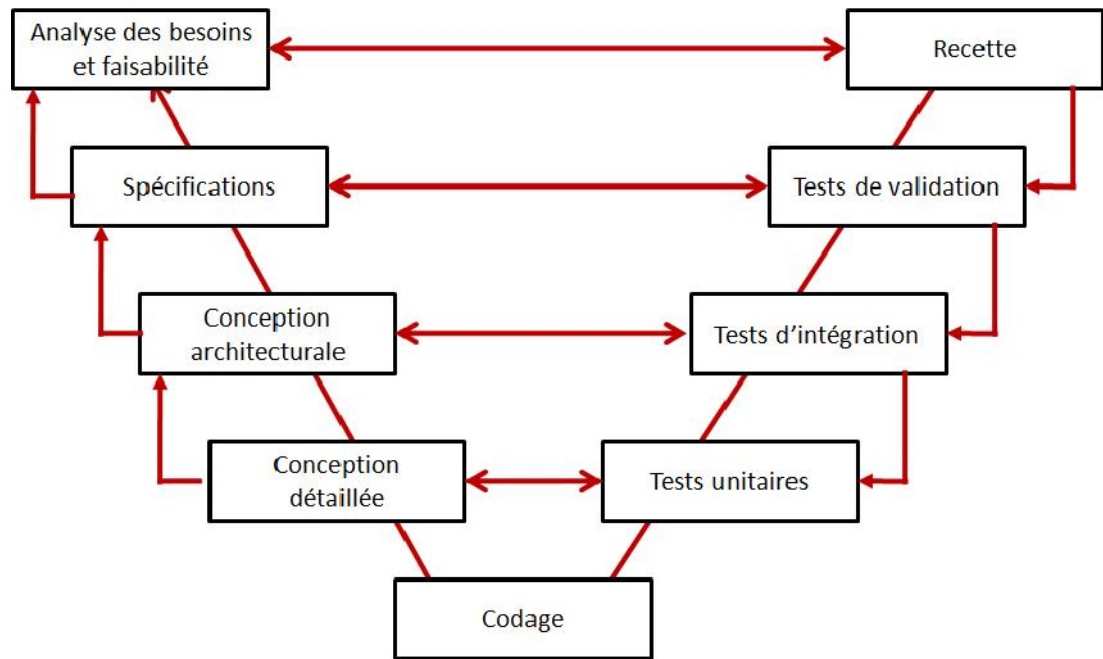


Figure 5: le Cycle en V¹¹

2.2.1 Les inconvénients du cycle en V

- Le cycle en V n'est pas adapté à tous les projets informatiques,
- Il est difficile à l'appliquer rigoureusement,
- il attend du client une expression détaillée et validée du besoin
- le cycle V laisse peu de place aux changements importants dans les spécifications dans une phase avancée du projet.
- La réalisation dure le temps qu'il faut.
- Le rendez vous est repris avec le client pour la recette.
- Certaines fonctionnalités demandées se révèlent finalement inutiles à l'usage.

En 1994 une enquête du Standish Group fait le constat suivant :

D'après le contenu du rapport publié par; « 31 % des projets informatiques sont arrêtés en cours de route, 52 % n'aboutissent qu'au prix d'un important dépassement des délais et du budget tout en offrant moins de fonctionnalités qu'il n'en était demandé ; seuls 16 % des projets peuvent être considérés comme des succès. »

Cette même enquête renouvelée en 2008 indique un taux de réussite de 35%, ce qui est plutôt positif mais demeure très faible. Le problème reste entier. Parmi les motifs d'échecs, arrivent en tête : Manque d'implication des utilisateurs finaux : 12,8 %.

Changement de spécifications en cours de projet : 11,8 %. [9, STANDISH GROUP]

¹¹ Vickoff Jean-Pierre, Méthode Agile, LES MEILLEURES PRATIQUES COMPREHENSION ET MISE EN ŒUVRE, p 5.
www.Entreprise-Agile.com

2.3 La méthode Agile

Toutes les méthodes de la gestion de projets comportent une phase d'analyse et conception approfondies. Lors de cette phase, il faut se poser trois grandes questions : QUI ? QUOI ? COMMENT ?

Qui ? Qui vont être les acteurs du système ? Les intervenants

Quoi ? Que doit faire le système ? La phase de spécifications fonctionnelles, c'est dans cette partie que l'on va retrouver toutes les règles de gestion métiers qui définissent ces spécifications fonctionnelles.

Comment ? Comment ce système va satisfaire les besoins des utilisateurs : c'est à ce moment que l'on va définir l'architecture, la conception détaillée (diagrammes de classes, scénarios et diagrammes de séquences, modèles conceptuels de données...).

Dans le cas de projets développés à l'aide des méthodes agiles, ces trois questions 'ci-dessus' se posent à chaque itération qui embarque un lot important d'évolutions. Ces itérations passent systématiquement par une mise à jour des différentes documentations (règles de gestion...) qui doit être faite avant le codage.

2.3.1 Qu'est ce que la méthode Agile ?

Le développement agile est un « Ensemble de méthodologies de développement basées sur le développement incrémental itératif. Le principe étant que les exigences et les solutions évoluent grâce à la collaboration entre des équipes auto-organisées et travaillant sur plusieurs domaines fonctionnels » [1, Glossaire CFTL/ISTQB, p10].

La méthode agile est une nouvelle pratique de développement de projets informatiques. Cette méthode repose sur trois critères : Itératif-Incrémental-Adaptatif.

L'aboutissement : un cycle adopté par l'ensembles des méthodes Agiles actuelles

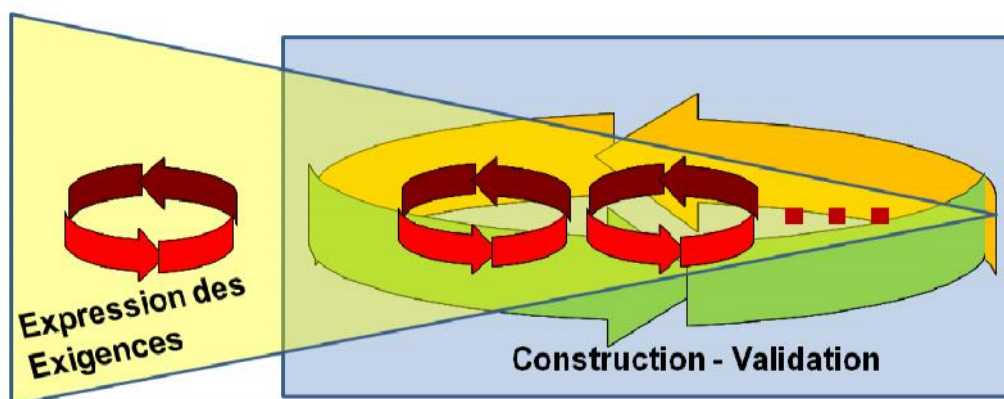


Figure 6: Itératif-Incrémental-Adaptatif¹²

¹² Vickoff Jean-Pierre, Méthode Agile, LES MEILLEURES PRATIQUES COMPREHENSION ET MISE EN ŒUVRE, p 6.

Une méthode itérative : la base de cette méthode itérative est la capacité de planifier une itération de production en termes de fonctionnalités et d'interdépendances.

Une méthode incrémentale : la production des fonctionnalités s'effectue en plusieurs incréments.

Une méthode adaptative : ce qui caractérise cette méthode, c'est sa capacité d'accepter les modifications, des changements et l'évolution du livrable, avant, après et pendant la production. Cela permet au client de modifier ses exigences en cours de projet.

2.3.2 D'où vient la méthode Agile ?

"Et c'est en 2001 qu'un groupe de 17 personnalités de l'industrie logicielle -(les noms figurent sur la figure 8 (ci-dessous) a défini une nouvelle manière de conduire les projets". La mise au point du manifeste "Agiles" signe la naissance des méthodes agiles dans le développement d'une application informatique."

La figure 8 représente : les quatre valeurs de la méthode Manifeste agile ainsi que les noms des 17 personnalités de l'industrie logicielle qui contribués à la naissance du Manifeste Agile.



Figure 7: le « Manifeste Agile »¹³

¹³ Le Manifeste Agile ou Agile Manifesto, est un texte apparu en 2001, rédigé par 17 experts du développement logiciel. Ce texte reprend les 4 valeurs communes des méthodes agiles et les dérive en 12 principes : <http://agilemanifesto.org/iso/fr/>

Le développement Agile est un terme issu du « Manifeste Agile », développé par un groupe d'experts du développement d'applications informatiques. Ce groupe estime que le traditionnel cycle de développement en cascade/cycle en V, ne correspondait plus aux contraintes et aux exigences des organisations en évolution rapide. Ce qui les conduit à développer une nouvelle méthodologie « le développement Agile ».

Toutes les méthodes dites "Agiles" s'articulent autour de 4 grandes valeurs développées par le manifeste agile et 12 principes, sont défendus par « l'Agile Alliance »¹⁴.

2.3.3 Les 4 valeurs de la méthode Agile :

Pour réussir le développement d'un logiciel il est Mieux de valoriser :

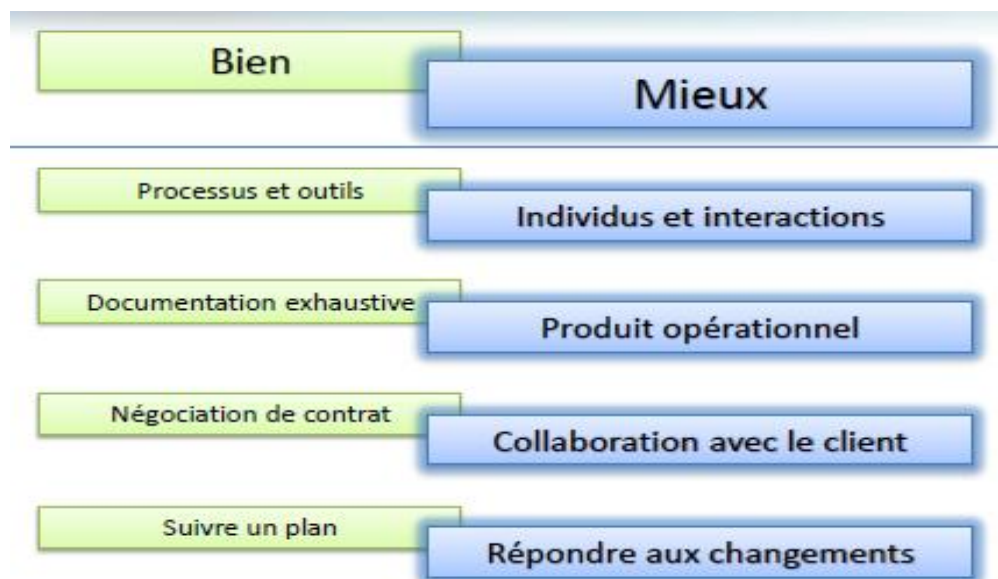


Figure 8: les 4 valeurs de la méthode Agile¹⁵

✓ **Les Individus et leurs interactions** mieux que les **Processus et outils** :

Pour créer des équipes ultra-performantes, les méthodologies agiles donnent la priorité aux personnes et aux interactions par rapport aux processus et aux outils. Toutefois, ces cycles fonctionnent uniquement lorsque l'équipe adopte un comportement essentiel : la transparence, approbation et le respect)

Des études sur la « saturation de la communication » au cours d'un projet révèlent qu'en l'absence de tout problème de communication, les équipes obtiennent des résultats 50 fois supérieurs à la moyenne industrielle. Pour faciliter la communication, les méthodologies agiles cherchent à augmenter la communication et la collaboration au moyen de cycles examen-et-adaptation fréquents

¹⁴ Association Agile Française pour promouvoir les méthodes agiles,
<http://agile-france.org/>

¹⁵ Le retour des spécifications dans un projet décisionnel lean, le 10 juin 2013,
<http://fleid.net/tag/projet-decisionnel/>

- ✓ Un Produit opérationnel mieux qu'une Documentation exhaustive :

Du logiciel qui fonctionne : Le Manifeste Agile recommande que les équipes distribuent les composants du logiciel éprouvé à des intervalles définis.

- ✓ La Collaboration avec le client mieux qu' une Négociation de contrat:

Les rédacteurs du manifeste ont souligné que la participation du client dans le processus de développement du logiciel est essentielle à la réussite étant donné que les méthodologies agiles identifient la valeur de l'engagement de client.

- ✓ Répondre au changements mieux que de Suivre d'un plan :

Les méthodologies agiles reposent sur le fait que, pour réussir, elles doivent planifier le changement. C'est pourquoi elles ont des processus établis, tels que des examens et des rétrospectives, qui sont conçus spécifiquement pour modifier régulièrement les priorités en fonction des commentaires du client et de la valeur commerciale.

2.3.4 Les 12 principes de la méthode Agile

Ce tableau ci-dessous, illustre les 12 principes de la méthode agile, développés par le « Manifeste Agile »

- Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
- Un logiciel opérationnel est la principale mesure d'avancement.
- Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Figure 9: Les 12 principes de la méthode Agile¹⁶

Analyses des deux schémas : Il existe plusieurs méthodes agiles individuelles, mais elles ont toutes les mêmes valeurs et même principes. Cependant nous avons résumé de cet

16

ensemble de principes et valeurs quelques facteurs permettent aux développeurs la réussite des méthodes agiles :

- la possibilité de livrer de manière continue des fonctionnalités complètes utiles et utilisables.
- le développement se fait par itérations courtes, cela, permet un gain de temps
- l'adaptation aux changements permet des solutions simples et adaptables
- l'agilité permet un rapprochement utilisateurs et développeurs et s'améliorer continuellement.

2.3.5 Exemple des méthodes Agiles « SCRUM »

Parmi les méthodes Agile, « Scrum » est la plus utilisée: elle est donc la plus éprouvée, documentée et supportée et très simple à comprendre. Sa maîtrise est en revanche difficile. Cette méthode à été créée en 2002, dont le nom est un terme emprunté au rugby qui signifie « la mêlée ». Elle s'appuie sur le découpage des projets en itérations encore nommées « sprints ».

Un sprint peut avoir une durée qui varie généralement entre deux semaines et un mois.

Scrum est considéré comme un cadre ou « framework » de gestion de projet.

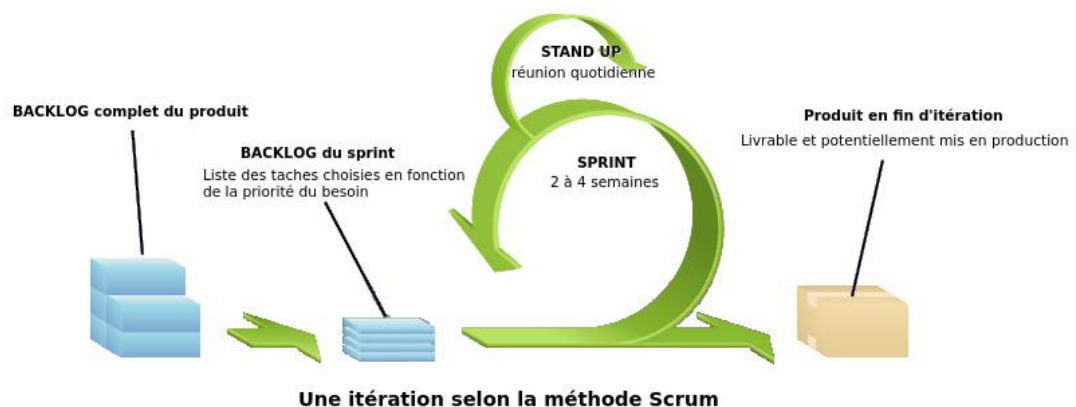


Figure 10: "La méthode SCRUM"¹⁷

Les trois rôles de La méthode « Scrum » : La méthode « Scrum » définit trois rôles pour un projet, [8, NEUMANN]

Le rôle du « Product Owner » :

- Il s'agit de celui qui porte la vision du produit à réaliser, représentant officiel du client,

¹⁷ http://ineumann.developpez.com/tutoriels/alm/agile_scrum/#LI-B

- Il est l'interlocuteur principal du Scrum Master et des membres de l'équipe,
- Il définit les besoins du produit et rédige les spécifications,
- Il peut se faire aider de responsables fonctionnels pour la rédaction des spécifications,
- Il est également chargé de définir et prioriser les users stories pour chaque sprints.

Le rôle « Scrum Master » :

- Il est le garant de l'application de la méthodologie Scrum,
- Il s'agit d'une personne chargée de veiller à la mise en application de la méthode,
- Il n'est pas un chef de projet, une personne chargée de lever les obstacles éventuels qui empêcherait l'avancement de l'équipe et du projet pendant les différents sprints.

Le rôle de l'équipe de développement qui réalise le produit « team members » :

- Des personnes chargées de la réalisation du sprint et d'un produit utilisable en fin de sprint.
- Il peut s'agir de développeurs, architectes, personnes chargées de faire des tests fonctionnels...

Les réunions d'un projet « Scrum » : La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps (timeboxing):

Planification du Sprint (Sprint = itération) : au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « Product Owner »).

Revue de Sprint : au cours de cette réunion qui a lieu à la fin du sprint, l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner et des utilisateurs finaux. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).

Rétrospective de Sprint : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du « vécu » sur le sprint écoulé (principe d'amélioration continue).

Mêlée quotidienne : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée « stand up meeting ») en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est ce que j'ai terminé depuis la dernière mêlée ? Qu'est ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? »

2.3.6 Avantages et inconvénients de la méthode Agile

Nous avons établi un tableau pour illustrer les multiples bénéfices des méthodes agiles et également les difficultés aux quels les équipes de développement font face aujourd'hui :

METHODES AGILES	
Avantages	Inconvénients
La méthode agile présente de multiples avantages : -Grace à la participation du client -commanditaire de l'application-, le produit est livré rapidement et répond parfaitement aux attentes. -Un gain en matière de coût : Une fois que le système possède l'essentiel des fonctionnalités nécessaires ; le client a les mains libres, il n'est plus obligé de continuer les itérations suivantes. -Une souplesse dans les spécifications permettant - un gain du temps : dans les méthodes agile, lorsqu'on s'aperçoit que certaines fonctionnalités ne répondent pas aux besoins, une fois les tests de validation ou de recettes réalisés, il suffira de prévoir de nouvelles tâches lors de la prochaine itération. -Contrairement à la méthode du cycle en V : il est souvent nécessaire de repartir au niveau des spécifications et de suivre à nouveau pas à pas toutes les étapes, ce qui peut prendre un certain temps et surtout un coût important.	les tests agiles, souffrent aujourd'hui d'une : -Pénurie de compétences : inclure des notions d'agilité dans les tests n'est pas facile, pour cela il faut prendre en compte des frais et délais considérables pour former les équipes de développement aux outils et bonnes pratiques de test logiciel. -la complexité des modèles et méthodes : l'agilité dans les procédures de tests logiciels nécessite des modèles et des méthodes plus sophistiqués; -La démarche de test devra être menée par un collaborateur capable de travailler en mode AGILE.

Tableau 3: Avantages et Inconvénients de la méthode Agile

Analyse : Effectuer des tests dans un projet agile n'est pas une démarche simple, vu la complexité des modèles des méthodes agiles. Les organisations et les testeurs en particuliers devraient faire face aux quelques difficultés qui apparaissent lors de la transition vers les méthodes agiles, de nouveaux rôles et responsabilités introduites. Mais malgré ces quelques difficultés, des études récentes démontrent que les méthodes agiles sont les plus répondues.

2.4 Du cycle traditionnel à l'agilité : Les incontournables du test logiciel

Le test dans un projet agile ? Pourquoi le développement d'un logiciel passe d'un cycle de vie traditionnel à l'agilité?

Le test doit s'appuyer sur des incontournables : Dans son article "Eric Riou Du Cosquer"¹⁸ l'auteur souligne les principes fondamentaux incontournables à mettre en œuvre, en les adaptant au mode de développement. Le test doit s'appuyer sur des principes seront déployés de différentes façons pour différents modes de développement y compris l'Agile. Ces principes sont :

- La nécessité de documenter les tests en s'appuyant sur la description des caractéristiques fonctionnelles ou non-fonctionnelles à tester,
- la nécessité de documenter leurs résultats,
- Possibilité de vérifier la traçabilité entre l'expression d'un besoin, les exigences, tests associés,
- le recours à un outillage adapté qui permettra de faire mieux et plus vite,
- La mise en œuvre d'un processus, consistant à identifier différents niveaux de test et à concevoir, implémenter, puis exécuter les tests à chaque niveau.

Tous les incontournables du test logiciel, sans exception, doivent être mis en œuvre dans un projet Agile.

Tous les principes fondamentaux et les incontournables du test logiciel, tels ceux recensés par les différents Syllabi de l'ISTQB¹⁹, se retrouvent sur un projet Agile. Le schéma des 4 quadrants, conçu par Brian Marick il y a près de 10 ans pour illustrer le test sur les projets Agiles le montre bien :

¹⁸ Eric Riou Du Cosquer, Trésorier, CFTL (Comité Français du Test Logiciel)

¹⁹ L'ISTQB (International Software Testing Qualifications Board »), le Comité international de qualification du test logiciel.
<http://www.istqb.org/>

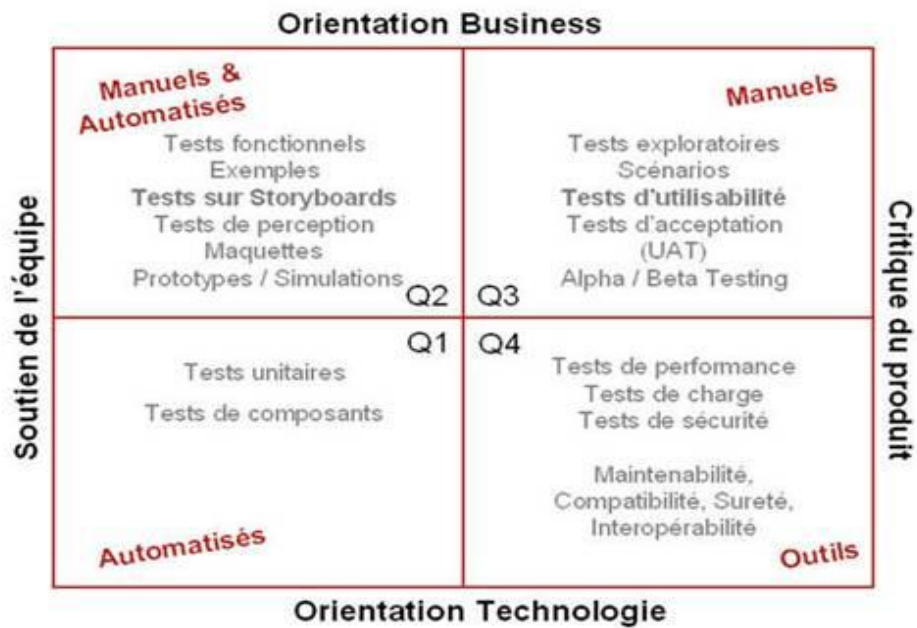


Tableau 4: Test sur les projets Agiles²⁰

Pour terminer, « Projet Agile ou pas, il n'y a qu'un test valable, celui qui permet d'augmenter la satisfaction des utilisateurs et la rentabilité des développements » [10, Eric RIOU].

²⁰ <http://dsisionnel.com/2012/12/20/test-agile-ou-test-dans-un-projet-agile/>

3 La veille concurrentielle : les grands axes du projet

Comment effectuer la veille sur les méthodes et les outils de test ?

La veille (surveillance de l'environnement) est « la mise en place formalisée et organisée d'un système d'information visant la collecte, le traitement et la diffusion de l'information concernant l'environnement de l'organisation, ceci de façon continue et dynamique. Elle est ainsi un outil de pilotage de l'entreprise pour réduire les incertitudes et accroître les opportunités ». [12, CHARTRON]

AFNOR propose la définition suivante : « Activité continue et en grande partie itérative visant à une surveillance active de l'environnement technologique, commercial, sociétal, réglementaire etc. pour en anticiper les évolutions. » [13, AFNOR]

Humbert LESCA complète cette approche en affirmant que « l'on peut comparer la veille stratégique de l'entreprise au radar d'un navire, puisqu'elle vise à anticiper les événements avant qu'il ne soit trop tard pour agir. Cependant, à la différence du radar, qui est objectif, la veille stratégique est caractérisée par son aspect interprétatif ». [14, LESCA]

En général l'action veille est un dispositif de surveillance continue des éléments suivants :

- l'actualité d'un secteur ou d'une activité,
- les nouveaux produits sur le marché,
- les prix,
- les acteurs principaux,
- les événements de la concurrence, et tout ce qui se dit quotidiennement du secteur (sur les réseaux sociaux, les blogs, la presse spécialisée, etc.).

Ce sont les grands points que nous allons développer dans cette partie, en s'appuyant sur le projet de « la veille concurrentielle » réalisé au sein du cabinet « Salihanne-Consulting » lieu du déroulement de mon stage.

3.1 Présentation du projet effectué au stage

L'intitulé du projet : « la veille concurrentielle sur les méthodes et outils de test et recette fonctionnelle »

3.1.1 Audit de situation

Avant de mettre en œuvre un dispositif de la veille, nous avons commencé par faire un audit de situation, et l'analyse des besoins du commanditaire. Cette étape nous a permis de répondre aux questions suivantes :

- Quel est le contexte et l'objectif de la demande?

- Quels sont les axes de surveillance ?
- Quels sont les acteurs ressources ?

L'environnement concurrentiel de l'activité consulting est en perpétuelle évolution : de nouveaux acteurs de la prestation MOA (la maîtrise d'ouvrage), AMOA (assistance à la maîtrise d'ouvrage), des outils modernes de la gestion des tests sur le marché, des méthodes agiles ; une offre considérable de solutions logiciels sur le marché.

En effet, le commanditaire « Salihane-Consulting », un groupe de consultants s'intéressent à toutes ces évolutions et souhaitent mettre place un projet de la veille concurrentielle sur l'activité consulting en informatique.

3.1.2 Contexte et Objectif

Ce projet a pour objectif de surveiller constamment et légalement l'environnement technologique, commercial, du marché de « test logiciel ». Pour être, informés, alertés sur les nouvelles technologies utilisables pour la réalisation des tests.

Pour obtenir toutes les informations actuelles, essentielles sur cette activité, nous avons mis en place un processus de veille, un moyen incontournable pour atteindre cet objectif ; faire un état des lieux, observer, enquêter et identifier toutes les informations nouvelles, utiles, relatives au secteur consulting, ensuite les diffuser au sein de l'entreprise afin d'en saisir des opportunités.

3.2 Le processus de veille

Quels sont les étapes essentielles du processus de la veille ? Quel dispositif à déployer ?

Compte tenu des besoins du commanditaire, le domaine de l'activité, les moyens, les axes de surveillance, nous avons suivi les étapes suivantes:

3.2.1 La recherche d'informations

A ce stade du processus se pose la question de la nature, de la diversité, de la quantité et de la qualité des sources d'information et des informations collectées.

Dans cette étape, il est nécessaire de définir le type d'informations jugées utiles, traiter et analyser les données collectées, et enfin les diffuser.

3.2.2 Le choix des sources

Une fois que les sources spécialisées sont identifiées, les exploiter et ensuite mettre en place un Sourcing : utiliser (web, serveurs de base de données professionnels, agrégateurs).

3.2.3 Le choix des outils

Il existe plusieurs outils pour réaliser une veille concurrentielle:

- Outils de recherche et de représentation des résultats :
- outils de la fouilles des données et Datamining : Calliope, Outils statistiques décisionnel Datawarehouse, Outils de data visualisation, Outils infographie.

- Les alertes Google: Les alertes Google, permettent de sélectionner des sujets que l'on trouve intéressants et de s'abonner. Cela permet de se tenir informé sur toutes les actualités relatives aux sujets sélectionnés et cela à une fréquence qui aura été paramétrée.
- Moteurs de recherche d'entreprises: ils répertorient toutes les entreprises enregistrées au registre du commerce ; ils permettent d'obtenir des données administratives et financières sur les entreprises.
- L'INPI: L'Institut national de la propriété industrielle propose de nombreux services et donne un accès aux bases de données concernant les marques, les brevets, les dessins, les modèles et les sociétés.
- AFNIC : (Association Française pour le Nommage en Coopération):Le site de l'AFNIC permet de connaître le détenteur d'un nom de domaine.
- Annuaire : réaliser un annuaire des sociétés de service nationales et internationales : (Adresses postales, les numéros de téléphones)
- Les newsletters: un moyen d'obtenir les dernières informations sur ses concurrents, il est nécessaire de s'abonner à leurs newsletters qui fourniront divers renseignements tels que des offres, des services, des nouveautés, etc.
- Réseaux sociaux d'entreprise :
- Les réseaux sociaux sont devenus une source incontournable d'informations ils permettent de suivre les acteurs du domaine en temps réel sur (Linkedin, Facebook, Tweeter, les forums pour recueillir des avis des internautes (une ressource à ne pas négliger), les blogs des concurrents aussi pouvant fournir des informations supplémentaires.
- Voici une publication du Groupement Français de l'Industrie de l'Information (GFII), qui démontre l'importance de la surveillance des réseaux sociaux notamment (réseaux sociaux professionnels) :
- « On assiste [...] à une décentralisation des processus de collecte et d'automatisation des tâches de surveillance. D'une approche initiale client-serveur où la donnée informative était collectée au sein d'une fonction documentation puis redistribuée vers les utilisateurs finaux, on passe progressivement à une organisation déconcentrée dans laquelle l'interface utilisateur final est devenue le point d'accès aux données. Le développement des services de veille en mode accès à distance (dans le cloud) vient renforcer cette tendance." » [15, GFII].
- Les fils RSS  : S'abonner aux flux RSS de la presse spécialisée, permet d'avoir un flux d'information et aussi d'être tenu informé à tout moment des événements organisés par les experts du secteur.

De plus en plus de sites internet proposent un abonnement à leurs flux RSS (Really Simple Syndication), offrant à l'utilisateur, sans avoir besoin de se connecter sur la page web, d'être

informé des nouveautés de la page. Il est possible de s'abonner à un ensemble de flux via une plate-forme qui rapatrie sur une page web unique l'ensemble des nouveaux flux publiés sur l'ensemble des pages auxquelles l'utilisateur s'est abonné. [16, MIAUX].

3.2.4 Etude de marché, analyse sectorielle

Une analyse sectorielle et une étude de marché à partir d'un code NAF (Nomenclature des Activités Françaises). Des données pouvant être précisées par l'INSEE (Institut National de la Statistique et des Etudes Economiques) et par d'autres sources.

D'après l'INSEE, l'activité conseil consulting, est classée dans le :

Code APE / NAF 6202A (Activité : Conseil en systèmes et logiciels informatiques)²¹

En effet l'utilisation des divers outils de veille cités ci-dessus nous permettent de répertorier les acteurs de la controverse, constituer un corpus (tous les débats sur le secteur et leurs arguments), ces résultats de la veille et l'analyse sectorielle nous permettent de connaître :

LES CHIFFRES CLÉS

PRINCIPAUX ACTEURS DE LA PROFESSION

ACTUALITE DU DOMAINE

CONJONCTURE DU SECTEUR

LE CADRE REGLEMENTAIRE DU SECTEUR

PLACE DU SECTEUR DANS L'INDUSTRIE

La grille SWOT : Application de la grille SWOT à la veille, permet de prévoir, anticiper et surveiller le marché et évaluer l'image de l'entreprise. C'est un moyen de mettre en évidence les forces (Strengths) et faiblesses (Weaknesses) internes à l'entreprise, ainsi que les opportunités (Opportunities) et menaces (Threats) externes. Ces deux derniers éléments sont particulièrement intéressants dans le cadre d'une veille concurrentielle.

La synthèse documentaire : faire une synthèse documentaire est nécessaire afin d'exploiter l'information obtenue pour l'entreprise.

3.2.5 Les facteurs clés du succès de la veille

« Le succès de la mise en place et du fonctionnement d'une cellule de veille repose sur une série de facteurs relevés dans la norme expérimentale » [13, AFNOR]

Cinq facteurs clés y sont identifiés :

- La réalisation de l'audit de l'existant
- La définition d'objectifs clairs associés à la stratégie de l'entreprise
- L'appel à des spécialistes

²¹ <http://www.verif.com/societe/SALIHANNE-CONSULTING-799376157/>

- La mise au point d'un système de contrôle et de pilotage de la veille
- La mise en place d'un plan de promotion interne des objectifs de la veille, de sensibilisation et de formation [[16, MIAUX, p 65]

3.3 Le BENCHMARKING

Comment mener un BENCHMARKING ? Quel objectif ?

« Qui veut s'améliorer doit se mesurer, qui veut être le meilleur doit se comparer »

(Robert C. Camp, initiateur de la démarche) [17, WALLON]

Le BENCHMARKING, est un processus continu de recherche, une technique d'observation, d'analyse comparative, des performances, des pratiques, des modes d'organisation des autres entreprises afin de s'en inspirer et d'en tirer le meilleur.

Dans un contexte technique, le benchmark est surtout orienté vers la comparaison de performances, comme par exemple dans l'informatique.

Dans un contexte marketing, le Benchmark est une démarche d'observation et d'analyse des performances atteintes et des pratiques utilisées par la concurrence ou par des secteurs d'activité pouvant avoir des modes de fonctionnement réutilisables par l'entreprise commanditaire du benchmark. Il peut être considéré de manière simpliste comme "une pêche aux bonnes idées". [18, Glossaire marketing].

Dans le cadre de notre projet de « la vielle concurrentielle les méthodes et outils de test logiciel », nous avons effectué un BENCHMARKING, pour faire une analyse comparative vu l'offre considérable, régulière des outils existants pour la réalisation des recettes applicatives, afin d'identifier les meilleurs de ces solutions logiciel.

3.3.1 Les objectifs du benchmarking

La méthode BENCHMARKING a été développée au début des années 1980, elle est devenue un standard du management en raison de sa capacité à améliorer l'efficacité et la qualité des processus d'une entreprise.

Les objectifs que produit cette technique sont :

- -une diffusion efficace et rapide des connaissances et bonnes pratiques au sein de l'entreprise, cela permet de gagner du temps.
- Une étude d'analyse et de comparaison des pratiques, des performances et processus de l'organisation avec celles de la concurrence, ensuite exploiter le résultat afin d'améliorer les processus existants
- Récupérer des informations sur la façon dont sont exécutées des activités similaires chez des concurrents,

Afin de répertorier toutes les nouvelles pratiques et nouveaux outils d'une recette applicative, recherchés par le commanditaire et pouvoir les comparer nous avons réalisé deux panoramas détaillés : le premier concerne les référentiels du test disponibles sur le marché, le deuxième concerne des cabinets de conseils en informatique et les outils qu'ils proposent :

3.3.2 Panorama des référentiels de tests

Après avoir effectué une recherche et une analyse sur les outils de test et les nouvelles pratiques, nous avons sélectionné les plus utilisés, ensuite nous avons effectué une analyse comparative : la plate forme de ces outils, les fonctionnalités attendues, leurs performances dans la gestion des tests ; les différences, les détails sont illustrés sur le tableau ci-dessous :

Tableau 5:Outils de test logiciel

Outils	Editeur	Fonctionnalités
 (4) Borland *SilkTest* ²²	 Micro Focus	-un outil d'automatisation des tests, il permet : -l'ouverture sur tous les NET, Web, Java™, -la réduction des coûts des activités de test, -une intégration et personnalisation de reprise, - l'exécution des tests sans surveillance, -la facilité de la maintenance,
 (3) Qmetry ²³	 « Hewlett Packard »	Un outil complet de gestion de test, il fournit : - une interface très simple et intuitive pour écrire les cas de test et les relier à des exigences. - un module de gestion des défauts flexible - une traçabilité automatique - des rapports - il s'adapte à différents workflows
 (2) QuickTest Professional (QTP) ²⁴	 « Hewlett Packard »	-Un outil d'automatisation des cas de test de non régression. -Il peut être couplé à l'outil de gestion (Quality Center). -Permet de vérifier des éléments de texte, les

²² <http://www.borland.com/Products>

²³ http://www.verifysoft.com/fr_qmetry_test_management.html

²⁴ <http://www.altea-conseil.com/les-outils-de-test/quality-center.html>

		propriétés des objets, des captures d'écran. -Des capacités de modification puissantes
 (1)Quality Center ²⁵	 « Hewlett Packard »	-un outil complet de gestion de recette, il permet de : -Référencer les tests et les exigences à contrôler - Gérer les campagnes de test -Lancer l'exécution des tests manuels et automatisés -Gérer les anomalies - Suivre l'avancement des campagnes de test à toutes les étapes du processus, grâce à la mise à disposition de fonctions d'analyse et de reporting.
 (6)Mantis ²⁶	 « Hewlett Packard »	Un outil de gestion des anomalies, il permet : -créer des tests automatisés sur des périphériques mobiles, -Il est équipé d'un système de reportings, - Il peut parfaitement être couplé à TestLink, -il possède une interface web en PHP, MySQL. - rend la gestion des anomalies simple et organisée.
 (5) Test Link ²⁷	sous licence GPL	- interopérabilité avec : Mantis, Bugzilla, Jira. -Un outil permettant de centraliser les cas de tests -Compatible avec la plupart des navigateurs web, - il utilise une architecture classique PHP/MYSQL

➤ D'autres solutions open sources ont été identifiées :

R.T.M.R. : « Requirements and Tests Management Repository » (Référentiel de gestion d'exigences et de tests). C'est un outil de test logiciel open source qui permet de gérer les exigences d'un logiciel tout au long de son cycle de vie, de décrire les scénarios et cas de

²⁵ <http://www.altea-conseil.com/les-outils-de-test/quality-center.html>

²⁶ <http://www.altea-conseil.com/mantis.html>

²⁷ <http://www.altea-conseil.com/les-outils-de-test/testlink.html>

tests assurant la validation de ces exigences, d'exécuter des campagnes de tests ciblées, de suivre l'ensemble des anomalies rencontrées lors des tests. La solution intègre un système de gestion de version par projet, par exigence, par scénario et par cas de test qui assure le suivi des évolutions logicielles et permet de couvrir aisément l'ensemble des tests de non régression.²⁸

SonarQube : (anciennement connu sous le nom « Sonar »), c est une plate forme open source pour gérer la qualité du code. Il a un moyen très efficace de la navigation, un équilibre entre la vue de haut niveau, tableau de bord, Time Machine et outil de chasse de défaut. Cela permet de découvrir rapidement des projets et / ou des composants qui sont dans la dette technique d'établir des plans d'action.²⁹

Software Diagnostics Flight Recorder : Un logiciel qui recueille automatiquement toutes les informations pertinentes sur l'exécution du logiciel. L'outil vous permet de gagner du temps dans la communication entre les testeurs et les développeurs ces derniers voient directement le comportement erroné du logiciel et trouvent ainsi l'erreur.³⁰

Testwell CMT++ /CMTJava : Testwell CMT++ et CMT Java, fournissent des estimations sur les probabilités d'erreurs dans l'application, sur la durée requise pour la compréhension du code, sur le volume logique du code, etc. Comme l'équipe travaillant sur un projet n'a généralement pas le temps de vérifier tout le code, CMT++ et CMT Java vous permettent de localiser les modules risquant de poser des problèmes³¹.

Squash TA : Un nouvel outil pour la gestion du patrimoine de tests. Une solution libre, sous licence LGPL v3. Cet outil permet de gérer l'ensemble des étapes d'une recette, généralement fonctionnelle, de la gestion des exigences à l'exécution des campagnes de test, en passant par les cas de tests, les scénarios, la gestion des anomalies, les comptes-rendus d'exécution, le reporting, etc. Par rapport à d'autres solutions, Squash TM il est aussi nativement multi-projets, chaque utilisateur pouvant accéder à l'ensemble des projets sur lesquels il a les droits³²

²⁸ <http://rtmr.net/>

²⁹ <http://www.sonarqube.org/>

³⁰ http://www.verifysoft.com/fr_software_diagnostics_flight_recorder.html

³¹ http://www.verifysoft.com/fr_cmtx.html

³² <http://linuxfr.org/news/squash-tm-nouvel-outil-pour-la-gestion-du-patrimoine-de-tests>.

3.3.3 Panorama des acteurs spécialisés dans la qualité logicielle

Améliorer la qualité des systèmes d'information en entreprise, la qualité des logiciels, c'est se poser sans cesse la double question :

Comment fabriquer un logiciel sans anomalies?

Le SI (Système d'information/le produit est-il bien celui dont les utilisateurs ont besoin?

Ces questions préoccupent les acteurs de l'activité consulting (les cabinets de conseil, sociétés de services spécialisés dans la qualité logiciel) et les amènent à évaluer et améliorer toute l'organisation et la fabrication d'un système d'informations (méthodes et outils des maîtrises d'œuvre et d'ouvrage), en effet, Ils deviennent fournisseurs d'idées et de solutions logiciels. Ces acteurs ont été identifiés lors de notre recherche : quelques cabinets de conseil et les équipes d'experts consultants, prestataires en (A) MOA (assistance/la maîtrise d'ouvrage) et MOE (la maîtrise d'œuvre). Sur le tableau suivant nous avons cité toutes les stratégies de la gestion des projets avec des solutions TIC qu'ils proposent :

Acteur	Services proposés	Outil Edité	Année création
 Acial	-cabinet de conseil a pour vocation -d'aider les entreprises à intégrer la dimension qualification sur l'ensemble du cycle de vie du Système d'Information.	- MMI : un outil pour l'amélioration de l'activité de test : un "pure player" du Test et de la Qualité Logicielle	1998
 Verifysoft Technology	-La société a pour objectif de fournir des solutions et des conseils dans le domaine des tests de logiciel, pour ses technologies éditées.	-l'outil d'exécution de tests Elvior TestCast d'Elvior, Tallin (Estonie) -outils de diagnostic de logiciels de Software Diagnostics, Potsdam -l'outil de gestion de tests QMetry™ de QMetry, Santa Clara (Etats-Unis). Testwell CTC++ Test Coverage Analyser Testwell CMT++ et CMTJava Code Complexity Measurement Tools Testwell CTA++ C++ Test Aider	2003

 EUROGROUP CONSULTING	-Pilotage de projets, -Alignement stratégique des SI avec les enjeux métiers et techniques -la mise en place d'une gouvernance de programme SI est indispensable. Elle permet de réduire les coûts unitaires par des logiques de pilotage du ROI, de rationalisation et de mutualisation des SI	SDSL : est un véritable outil de management et de pilotage de la filière SI.	près de 30 ans
 OSAXIS	-la gestion de projet, conseil, audit, et réalisation. -Elaboration des systèmes d'informations et/ou l'ingénierie en architectures Web et Mobile. -rédaction de cahier des charges -élaboration d'expressions de besoins -diverses missions d'AMOA/AMOE -recette fonctionnelle	/	/
 Pentalog	- prestations de réalisation projets -contrôle des applications -la réalisation des plateformes et outils des tests. -La réalisation de tests fonctionnels -La réalisation de tests techniques : développement de tests unitaires, tests de performance, tests de robustesse, tests de vulnérabilité	/	/

Tableau 6: Panorama des acteurs de l'activité consulting

- D'autres sociétés spécialisées dans la qualité logicielle :

ConseilOrga³³ : « accompagnement et direction de projets et programme » : Créée par Vincent Drecq, ConseilOrga est un cabinet de conseil en système d'information et direction

³³ <http://www.altea-conseil.com/consulter-nos-offres.html>

de projets et programmes et aide à la décision, dans l'objectif de promouvoir les bonnes pratiques en gestion de projet.

ConseilOrga intervient sur toutes les phases amont d'un projet, et s'engage à aider et accompagner ses clients dans le pilotage de leurs projets et programmes.

Akal-consulting³⁴: AKAL Consulting est une société de conseil en informatique spécialisée dans le domaine des infrastructures réseaux et systèmes : optimisation des réseaux, conseil en télécoms, conseil en systèmes des organisations multi-sites.

Le travail de l'équipe AKAL-Consulting permet de réduire le temps consacré au projet côté Client, tout en assurant un cadre méthodologique (AKAL Smart), la qualité et la traçabilité des livrables. Des consultants intervenants en (A) MOA, l'équipe Akal Consulting, accompagne la conception et la mise en œuvre des infrastructures SI.

Virage groupe³⁵ : un éditeur de logiciel de gestion de portefeuilles de projets (PPM). Des intervenants experts de logiciels et du pilotage, capables de travailler sur les processus de pilotage, l'intégration du logiciel dans le système d'information, d'assurer des formations et d'accompagner le changement. VIRAGE s'engage à accompagner ses clients dans l'utilisation de l'application, dans le paramétrage du logiciel, dans l'exploitation technique et les mises à jour.

Hsc³⁶ : un cabinet de consultants en sécurité informatique depuis 1989 - Spécialisé sur Unix, Windows, TCP/IP et Internet. Le cabinet HSC accompagne et oriente ses clients dans toutes les phases amont d'un projet: (Assistance à maîtrise d'ouvrage, prestations de conseil et de conception, rédaction et dépouillement d'appels d'offres. L'équipe des consultants HSC offre des solutions de sécurité et des fonctionnalités fournies par rapport aux besoins exprimés, et intervient également sur des architectures existantes afin d'intégrer de nouveaux besoins de l'entreprise ou d'améliorer la sécurité d'une plate-forme.

Pentalog³⁷ : Une société de services informatiques, propose une gamme de prestations IT qui couvre l'ensemble des tests et intervient pour dans la qualification et le contrôle de votre application et met en place toute l'infrastructure nécessaire pour la réalisation du projet (outils et les plateformes). Pentalog intervient sur :

La réalisation de tests fonctionnels : TRA, Tests d'intégration, Tests de non régression

³⁴ <http://www.akal-consulting.com/#!/resolution-incident/cd0d>

³⁵ <http://www.viragegroup.com/fr/services/nos-services>

³⁶ <http://www.hsc.fr/services/conseil.html.fr#ouvrage>

³⁷ http://www.pentalog.fr/systemes_d_information/testing_systemes_d_information.htm

La réalisation de tests techniques : développement de tests unitaires, tests de performance, tests de robustesse, tests de vulnérabilité

Mise à disposition de plateforme de tests : mise en place de plateforme virtuelle pour qualifier un logiciel sur plusieurs environnements.

ValidIT³⁸: l'équipe ValidIT commercialise et déploie des solutions technologiques, des services professionnels pour fiabiliser, tester les applications informatiques avant leur déploiement et surveiller les niveaux de service après déploiement.

Les consultants interviennent aux problématiques de validation des processus métier automatisés :

- Mise en place d'indicateurs clés de l'activité métier
- Contrôle des flux de données inter-applicatifs
- Déploiement de surveillances applicatives
- Tests de charge et diagnostics de performance
- Industrialisation des activités de recette et l'automatisation des tests.

3.4 Actualité de l'activité de test

Dans le cadre du projet veille et la surveillance du secteur consulting en informatique, nous vous proposons également une liste des organisations professionnelles (associations mondiales sans but lucratif) qui investissent pour promouvoir, et défendre la profession de chef de projet, la fabrication du logiciel libre, les valeurs et principes des méthodes agiles, afin d'aider les professionnels de cette carrière à progresser et améliorer leur réussite. Nous nous sommes aussi intéressées aux différentes publications sur la presse et les réseaux sociaux et quelques événements organisés au sujet de l'activité de test logiciel

3.4.1 Les organisations professionnelles de l'activité de test

Ci-dessous une présentation de chaque organisation, sa date de création, et ses finalités. Pour aller plus loin sur le sujet de ces organisations des liens internet indiqués en bas de cette page :

- CFTL : « le Comité Français des Tests Logiciels (CFTL) »³⁹

Une association à but non lucratif fondée en 2004 pour faire la promotion du métier de testeur de logiciels. Le bureau du CFTL détermine le contenu des formations et les questions d'examen associées.

Les Membres du bureau du CFTL : Le bureau du CFTL a été renouvelé lors de l'Assemblée Générale de décembre 2013 et comprend :

³⁸ <http://www.validit.com/Societe.html>

³⁹ <http://www.cftl.fr/>

Président : Eric Riou du Cosquer, Consultant Manager Certilog

ISTQB : Secretary

Vice - Président : Olivier Denoo, vice-président testware

Trésorier : Bertrand Cornanguer, Consultant Senior en Qualité des Systèmes Informatiques

Secrétaire : Bruno Legeard, Université de Franche-Comté et Smartesting

Le Comité Technique : Afin de garantir l'adéquation entre la certification CFTL-ISTQB et le monde de l'entreprise, le CFTL s'est doté d'un Comité Technique dont l'avis est consultatif. Ce comité est constitué de personnalités académiques, du monde industriel et des services :

➤ PMI⁴⁰ « Projet Management Institute » :

Une association sans but lucratif au monde pour la profession de chef de projet.

Nos ressources professionnelles et de recherche de cette organisation, permettent à plus de 700.000 membres, détenteurs de la certification et bénévoles dans presque tous les pays du monde de faire progresser leur carrière, d'améliorer la réussite de leurs organisations et contribue à accroître la maturité de la profession.

- Cet investissement et ce soutien mondial du PMI pour la gestion de projet est renforcé par les activités suivantes:
- Normes mondialement reconnues et programme de certification
- Vastes programmes de recherche académiques et du marché
- Chapitres et communautés de pratique
- Possibilités de perfectionnement professionnel

Le Chapitre France est la représentation du PMI à l'échelle de notre pays et des personnes qui s'y rattachent.

➤ APRIL⁴¹ : « Une association promouvoir et défendre le logiciel libre ».

Pionnière du logiciel libre en France, l'April, constituée de 4114 adhérents (3721 individus, 393 entreprises, associations et organisations), est depuis 1996 un acteur majeur de la démocratisation et de la diffusion du logiciel libre et des standards ouverts auprès du grand public, des professionnels et des institutions dans l'espace francophone :

➤ AGILE ALLIANCE⁴² :

L'Agile Alliance est une organisation sans but lucratif chargée de promouvoir à l'échelle mondiale les valeurs et principes du Manifeste agile.

⁴⁰ <http://www.pmi-france.org/>

⁴¹ <http://www.april.org/association>

⁴² <http://agile-france.org/>

3.4.2 Agenda d'événements

Nous nous sommes intéressés également aux événements concernant l'activité test logiciel (Les conférences, les colloques, les journées nationales et internationales), pour avoir plus d'informations nous vous proposons une visite virtuelle sur les liens internet cités dans l'annexe de ce mémoire, cela vous permettra de répertorier tous les experts du domaine ainsi que leurs offres :

➤ Congrès :

Le 6^{ème} Congrès Européen de science des systèmes, organisé le 19-22 septembre 2005, animé par le professeur associé à l'Ecole des Mines de Nantes : " Alain Moscowitz »

Ce texte est inspiré des travaux d'Henri Egéa, Jean-François Vautier et Bernard Séjourné sur "la conduite de projets et les nouvelles représentations du Monde ainsi que les publications de l'AFITEP, AFAI et PMI". ⁴³

➤ Journées⁴⁴ :

« Les journées françaises des tests logiciels » : 6^{ème} édition

Le Comité Français des Tests Logiciels (CFTL) met le South African Software Testing Qualifications Board (SASTQB) à l'honneur de sa 6^e édition de la Journée Française des Tests Logiciels.

« Les journées Française des tests logiciels » : 7^{ème} édition ⁴⁵

C'est l'évènement de l'année 2005 dans le domaine des tests logiciels. Il s'agit de la 7^{ème} édition de la Journée Française des Tests Logiciels s'est déroulée au Beffroi de Montrouge le 14 avril 2015, un lieu réservé pour recevoir les 700 participants attendus.

Cette journée est précédée d'une journée de tutoriaux le 13 avril dans lesquels des experts ont apporté leur expérience pratique dans des ateliers. Un évènement entièrement dédié aux tests logiciels et à la qualité des systèmes d'information, une occasion pour les entreprises de présenter ses offres, de participer et d'intervenir dans le cadre des différentes conférences proposées.

3.4.3 Les publications sur les médias sociaux

Les réseaux sociaux est une source incontournable pour surveiller tout ce qui se dit sur cette activité : (Sur les réseaux sociaux (Linkedin, Facebook, Tweeter, Forums, Blogs). Suite à notre recherche effectuée, nous avons sélectionné quelques blogues des professionnels de

⁴³ <http://www.res-systemica.org/afscet/resSystemica/Paris05/moscowitz.pdf>

⁴⁴ <http://dsisionnel.com/2014/03/10/lafrique-du-sud-a-lhonneur-de-la-6e-edition-de-la-jftl/>

⁴⁵ <http://www.jftl.org/>

test logiciel et quelques publications (des articles intéressants) que nous vous proposons de consulter :

LES BLOGS/CLUBS DES CONSULTANTS FONCTIONNELS

- Blog du secteur technologie «Ulrich's blog »⁴⁶ :

L'article : "Intégration des tests fonctionnels CasperJS avec Grails", publié le 7 avril 2013 par ULRICH'S BLOG.

Le but de cet article est de vous présenter un plugin Grails permettant d'instrumentaliser vos tests fonctionnels via CasperJS. la méthode d'utilisation et ses avantages

- Blog des consultants indépendants :

Développer votre Activité de Conseil et Service

L'article : "LES 7 GALÈRES DU CONSULTANT INDÉPENDANT"

Dans cet article, l'auteur souhaite revenir sur les préoccupations et difficultés majeures du métier de consultants indépendants.

<http://blog-du-consultant.fr/les-septs-galeres-du-consultant-independant/>

- Le blog de « OSAXIS »,⁴⁷

un cabinet de conseil et d'ingénierie en système d'informations.

Créé en 2003 par trois associés issus de différentes expériences de recherche et développement, la société : Des articles sont régulièrement mis en ligne sur des sujets divers et variés afin de vous faire part des actualités du moment tout.

- Un blog sur la Gestion de Projets Agile ⁴⁸:

Des articles divers sur la gestion de projets informatiques, voici les plus consultés :

- Comment élaborer une stratégie de tests
- Comprendre la théorie des contraintes, par le jeu.
- Gestion de projet - étape #4: La recette fonctionnelle (ou utilisateur)
- Le processus d'élaboration des Spécifications en méthodes Agiles
- Les jalons d'un projet informatique
- Modèle de procès-verbal de réception (ou livraison)
- Gestion des risques projets: l'analyse des risques 1/3

⁴⁶ <http://ulrichinaction.blogspot.fr/2013/04/integration-des-tests-fonctionnels.html>

⁴⁷ <http://www.osaxis.fr/blog/automatiser-ses-tests-fonctionnels-partie-2-2/>

⁴⁸ <http://gestiondeprojets.wordpress.com/sabonner-au-blog/>

➤ Blog de « DALISYS »⁴⁹ :

Une société de service spécialisée en test logiciel des systèmes d'information, reconnue à ce jour auprès de plus de 100 grands comptes et clients en France et en Europe. La société est l'un des acteurs pionniers de la professionnalisation des activités de test en France. un article publié en octobre 2013 sur « les tests fonctionnels automatisés : les 5 Facteurs Clés du succès » : La presse en parle :

Quelques revues et magazines spécialisés :

➤ DUNO : « Editeur de savoir »⁵⁰

Paru aux éditions DUNOD : Nous l'avons rencontré pour qu'il nous présente l'ouvrage qu'il vient de publier : "Pratiques du management de projet, 40 outils et techniques pour prendre la bonne décision". Le groupe Hachette Livre, a rencontré l'auteur : « Vincent Drecq » pour entretenir avec lui au sujet de cet ouvrage

➤ DSISIONNE⁵¹L :

Inscrit en mars 2010, DSISIONNEL est le magazine dédié aux enjeux des Direction des systèmes d'informations, face aux challenges des nouvelles technologies et au pilotage des activités du système d'informations.



:<http://dsisionnel.com/>

➤ LE JDN :⁵²

« le journal du net », le premier magazine en ligne du grand public Français de l'Internet.

Concernant les sujets de l'activité test, nous proposons quelques articles :

- Processus de tests fondamentaux
- Test logiciel : processus fondamentaux
- Les tests logiciels : des garants de la qualité - Gestion de projets...
- Tests psychotechniques recrutement - Test psychotechnique
- Comment réussir ? - Les tests de personnalités décryptés
- Salaire des ingénieurs logiciel en 2013 - Salaire d'un ingénieur logiciel

⁴⁹ <http://www.dalisys.com/blog/category/bonnes-pratiques-test-logiciel/>

⁵⁰ <http://www.dunod.com/collection/strategies-et-management>

⁵¹ <http://dsisionnel.com/2014/03/10/lafrique-du-sud-a-lhonneur-de-la-6e-edition-de-la-jftl/> .

⁵² <http://www.journaldunet.com/developpeur/algo-methodes/test-logiciel-processus-fondamentaux/>

➤ Linux magazine⁵³ :

Linux Magazine France est un magazine francophone mensuel spécialisé dans les logiciels libres : Article publié dans Linux Magazine 78, décembre 2005 : « Introduction à la production informatique ».

➤ La revue Génie logiciel : ⁵⁴

La revue « Génie logiciel », est une publication francophone sur le génie logiciel et l'ingénierie de système :

➤ LeMagIT⁵⁵ :

Un site internet dédié à l'information IT qui répond aux besoins spécifiques des CIO et des Directeurs des Systèmes d'Information (DSI) ainsi qu'aux besoins techniques des équipes informatiques.

➤ Revue APRIL : « Promouvoir et Défendre les Logiciels Libres »⁵⁶ :

Cette revue de presse sur Internet fait partie du travail de veille mené par l'April dans le cadre de son action de défense et de promotion du logiciel libre :

3.4.4 Retour d'expérience : Témoignage/ Interviews

Ce travail de recherche s'appuie sur de nombreux retours d'expériences vécues par les acteurs de l'activité test au cours de leur vie professionnelle.

o Témoignage sur le métier des consultants :

Dans un enregistrement, l'auteur « Didier Durandy », revient sur son histoire et sur son parcours de consultant en France, et à l'international pour des hommes clés qui font appel à lui, souvent dans l'urgence. Son ouvrage est un guide de conseils pour les consultants dont les missions sont parfois périlleuses, comme le laisse sous-entendre son sous-titre : « Kit de survie pour réaliser des missions en terrain miné ». ⁵⁷

⁵³ <http://articles.mongueurs.net/magazines/linuxmag78.html>

⁵⁴ [http://ticri.univ-rraine.fr/informatique.fr/index.php/G%C3%A9nie_logiciel_\(revue\)](http://ticri.univ-rraine.fr/informatique.fr/index.php/G%C3%A9nie_logiciel_(revue))

⁵⁵ <http://www.lemagit.fr/>:

⁵⁶ <http://www.april.org/revue-de-presse>

⁵⁷ http://www.widoobiz.com/?powerpress_pinw=38425-podcast

o Témoignage sur les méthodes de projet :

Témoignage client: auteur "Thierry Roche, DSI de l'Apec". Quels ont été les avantages de ces méthodes pour votre projet ? :

« Déjà, on voit concrètement l'évolution du projet car après chaque itération, les utilisateurs peuvent visualiser un « bout » de projet qui fonctionne, ça brise l'effet tunnel des méthodes classiques. Cette évolution nous permet de prioriser nos réels besoins, l'application s'enrichit selon nos demandes. Le surplus n'existe pas, on ne développe pas des fonctionnalités qui ne nous serviront jamais comme c'est régulièrement le cas en adoptant des méthodes classiques. On atteint un bénéfice utilisateur plus rapidement mais aussi un gain financier non négligeable. Nous n'imaginons même pas un retour avec des méthodes classiques. »

La source : Le Monde Informatique | Dossier méthodes agiles : Le renouveau des relations client/fournisseurs.⁵⁸

⁵⁸ <http://www.lemondeinformatique.fr/les-dossiers/lire-methodes-agiles-le-renouveau-des-relations-client-fournisseurs-en-ingenierie-429.html>

Troisième partie:
Organisation et planification de la
Recette Fonctionnelle

Cette partie consiste à présenter les méthodes essentielles pour la planification et l'organisation d'une prestation d'AMOA de recette fonctionnelle.

La présentation s'appuie sur mon expérience au cabinet de conseil « Salihanne-Consulting » suite à ma participation à l'organisation et la planification de la stratégie de test.

Les grands points développés dans cette partie sont :

- Le contexte et objectif de la recette fonctionnelle,
- Les principes généraux de la recette,
- Les grandes étapes de la recette,
- Les points clés de la recette fonctionnelle

Ensuite nous terminerons par présenter l'outil utilisé pendant la réalisation de notre projet de recette « Quality Center » : des détails sur les modules de cet outil, ses avantages, et toutes ses fonctionnalités attendues.

1 Contexte et objectif de la recette

La « recette », c'est l'opération par laquelle le client reconnaît que le produit livré par le fournisseur est conforme à la « commande » passée, qu'il est exploitable dans le système d'information de l'entreprise, et enfin qu'il est opportun de le mettre à disposition des utilisateurs. [2, DRECQ Vincent]

Ce projet de recette fonctionnelle, que nous allons vous présenter, a été effectué par l'équipe du Cabinet de conseil « Salihanne-Consulting », suite à l'envoi de « Crédit agricole » d'un dossier relatif à une prestation d'Assistance à la maîtrise d'ouvrage pour la vérification d'aptitude fonctionnelle d'une application "Assurance vie", le client souhaite faire une migration des contrats d'épargne et retraite vers un nouveau système d'information ; créer un nouveau programme permettant de gérer ces contrats.

Dans le cadre de ce programme, des développements informatiques ont été réalisés, ensuite la recette fonctionnelle, indispensables avant la mise en production. Il s'agit, donc, de réaliser, la recette « utilisateurs » de ces développements informatiques.

Nous avons pris en charge ce projet avec la collaboration de l'équipe prestataire « Salihanne-Consulting » depuis l'élaboration de la stratégie de recette jusqu'à l'établissement du PV (procès verbal) de la recette.

Ce projet de recette a pour but d'assurer que :

- L'application répond aux besoins fonctionnels des utilisateurs.
- L'application livrée correspond à la conception,
- l'application forme un ensemble cohérent,
- la solution technique peut être correctement exploitée,.

2 Les principes généraux de la recette

L'activité de tests mise en œuvre durant le cycle de développement du projet « Assurance vie » doit assurer une couverture de tests prenant en compte :

- La couverture des exigences de « Crédit agricole » énoncées dans les documents référencés dans le Plan de Management de Projet « Assurance vie ».
- La couverture fonctionnelle entre les spécifications fonctionnelles et les tests.
- La couverture d'intégration entre les Spécifications Techniques et les spécifications d'essais d'intégration
- La couverture du logiciel (parcours de l'arborescence du code).

Les différents types de tests sont décrits avec, en particulier, les tests des progiciels intégrés, les tests de défense et de performances. Les principes de constitution et d'exécution des tests de non régression sont également énoncés ainsi que les méthodes de tests et l'utilisation d'outils.

Les différentes phases de tests sont détaillées. L'activité de tests mise en œuvre durant le cycle de développement du projet « Assurance vie » est déclinée en 3 phases:

1. Phase de tests unitaires (comportant éventuellement les tests usines des plates-formes mises en place),
 2. Phase de tests d'intégration comportant les tests d'intégration unitaires, tests fonctionnels et de validation interne ; ces tests intègrent les essais d'intégration avec les systèmes externes ou tests d'interopérabilité.
 3. Phase de validation du client « Crédit agricole » comprend deux étapes: vérification d'aptitude au bon fonctionnement (VABF) puis vérification en service régulier (VSR).
- Chaque phase est découpée en trois étapes :
 - La préparation des tests,
 - L'exécution des tests,
 - Le bilan des tests.

La prise en compte des fiches de modification dans le cycle de tests du projet est ensuite décrite ainsi que la gestion des anomalies et des corrections.

Le suivi, les revues et les indicateurs sont définis. Puis sont indiquées les spécifications des besoins de tests, les fonctionnalités non testées et les conséquences des limitations de configuration des environnements de tests.

Les éléments de planification ainsi que les responsabilités associées sont précisés.

3 Les grandes étapes de la Recette fonctionnelle

L'équipe "Salihanne consulting" garantit un niveau de qualité optimale des applications livrées en s'appuyant sur deux étapes essentielles de la stratégie de recette ; la PREPARATION et la REALISATION : comme indiqué sur la figure 12⁵⁹ ci-dessous :



Figure 11: les étapes de la recette fonctionnelle

3.1 Etape de la préparation :

Cette première étape consiste à organiser une réunion de lancement du projet, dans le but de définir de définir :

- Le planning
- Les acteurs : Il s'agit de citer nommément les différents intervenants sur le projet de recette ainsi que leur rôle précis et leurs tâches principales au niveau de la recette. Ainsi que les différentes interactions entre ceux-ci.
- La documentation nécessaire : Une lecture approfondie de la documentation de l'application : CFG, CFD (spécifications générales et spécifications détaillées, la documentation projet (présentation, compte-rendu...).
- L'organisation des ateliers : dans le but d'échange d'idées sous la forme d'interview des différents acteurs déjà en place (MOA, chef de projet, interlocuteurs fonctionnels et techniques, etc.).

⁵⁹<http://www.conseilorga.com/Documents/Juillet%202010%20-%20Les%20Tests%20et%20l'informatique.pdf>

- Rédaction du PAQ : Le Plan Assurance Qualité qui doit être produit est relatif à l'activité de la recette
- Les outils et environnements : En amont de l'exécution des tests, il faut se renseigner sur l'environnement de test. Une description sera détaillée, afin d'avoir une photo précise de la plateforme de test, ceci doit permettre de répondre à toutes les questions techniques qui peuvent se poser durant la recette, mais également plus ou moins longtemps après la recette.

3.2 Etape de la réalisation : pilotage de la recette

Description détaillée des phases de tests, des types de tests ; réalisation des tests, la gestion des anomalies, de suivi de déroulement des tests, un PV de recette et un bilan permettant d'améliorer la prochaine série de tests.

3.2.1 Définition des exigences

Cette étape consiste à identifier les exigences de test qui constituent les objectifs de test. Elle s'appuie principalement sur les exigences du projet qui vont permettre à tout moment de s'assurer de la Couverture des tests (Tests de performances, de défense, de robustesse), de la Constitution et exécution des tests de non régression et des Méthodes de tests et utilisation d'outils.

3.2.1.1 Exigences fonctionnelles

- Fonctionnalités
- Processus métier

3.2.1.2 Exigences de sécurité et d'accessibilité

- Identification/ authentification,
- Droits d'accès et habilitations gérés à la fois sur des fonctions et sur des données,
- Traçabilité des opérations (dont identification de l'utilisateur)
- Prise en compte le cas échéant des règles de sécurité d'une charte

3.2.1.3 Exigence d'ergonomie

- Ecrans de consultation : lisibilité des écrans, choix des couleurs, clarté des menus,
- Fenêtres : Facilité d'accès, nombre de fonctionnalités à l'intérieur d'une fenêtre, navigation horizontale et verticale,
- Messages : Clarté des messages, lisibilité des messages, durée d'apparition, facilité d'effacement,
- Editions : facilité du paramétrage de l'imprimante, qualité d'impression...
- Aide en ligne
- Respect de la charte graphique

3.2.1.4 Exigences de performance : par exemple,

Utilisation de fonctionnalités plutôt simples, mais très utilisées (comme la création d'une commande client ...).

Finalement, il s'agira de s'assurer que toutes les exigences sont bien couvertes et qu'à chaque exigence correspond au moins un test.

3.2.2 La préparation des tests :

C'est la phase de la conception et la préparation des tests. Il s'agit de la :

Rédaction du cahier de test, (une traduction fine des règles de gestion).

La préparation des fiches de test et le choix des données de tests qui les accompagnent qui vont permettre de valider l'adéquation de l'application par rapport aux spécifications fonctionnelles, mais également par rapport aux besoins métiers.

La préparation des jeux de données qui doivent être complets et effectifs pour permettre de valider les résultats obtenus.

3.2.3 L'exécution des tests :

L'étape d'exécution de la recette doit assurer la maîtrise d'ouvrage que les anomalies critiques et majeures sont identifiées et traitées avant la mise en service de l'application en environnement de production.

Cette phase d'exécution de la recette s'articule autour de :

- La campagne de test,
- la gestion des anomalies,
- suivies de déroulement des tests,
- PV de recette

➤ La campagne de test :

Concevoir les scénarii : les intervenants sur cette phase auront à charge de « dérouler » le scénario dans l'ordre défini, puis, de vérifier les résultats.

Tout résultat obtenu est consigné au niveau de la campagne de tests.

Lorsque le résultat obtenu correspond au résultat attendu, seule une information relative à l'exactitude du test est suffisante comme preuve de bonne exécution du test. Si le résultat obtenu ne correspond pas au résultat attendu, ce résultat est précisé au niveau de la campagne de test et une fiche d'anomalie est ouverte pour décrire l'incident.

➤ La gestion des anomalies :

Toute anomalie constatée est consignée sur une fiche : « Fiche d'anomalie ».

La fiche d'anomalie doit être accompagnée de tous les documents pouvant aider à l'identification et à la résolution du problème :

- image de l'écran au moment où s'est produite l'anomalie, par exemple,
- jeu d'essai utilisé,
- succession des actions effectuées auparavant (l'origine d'une anomalie est parfois dans un traitement ou un écran antérieur).

Toutes les fiches d'anomalies sont centralisées pour être étudiées par le responsable de l'équipe de recette.

➤ Suivi de déroulement des tests :

Cette activité se déroulera tout au long de l'étape d'exécution de la recette fonctionnelle. Ce suivi est réalisé par les moyens suivants :

- Report des résultats des tests sur un document spécifique,
- Point hebdomadaire en présence des responsables d'intégration et/ou chefs de projet,
- Synthèse du passage des tests constituée à partir de la saisie des résultats de test.

La synthèse du passage des tests de la phase concernée permet de fournir pour chaque document du plan de test les éléments suivants :

- Nombre de tests passés
- Nombre de tests concluants
- Nombre d'anomalies
- Nombre de tests non applicables

Une synthèse globale des documents du plan de test présente les informations suivantes :

- Identification du document
- Nombre de tests par document
- Résultats des tests par document
- Nombre de tests passés
- Nombre de tests concluants
- Nombre d'anomalies
- Nombre de tests non applicables
- Pourcentage de passage des tests dans un document
 - Pourcentage de tests passés par rapport à ceux prévus
 - Pourcentage de tests réussis par rapport à ceux passés
- Pourcentage de passage des tests dans un document
- Nombre d'anomalies identifiés par document
- Synthèse globale
 - Nombre total de test

- Pourcentage globaux
- Pourcentage de tests passés par rapport à ceux prévus
- Pourcentage de tests réussis par rapport à ceux passés
- Nombre total d'anomalie identifiés

Ces tableaux sont mis à jour par le Chef de projet ou le responsable d'intégration. Un exemple de tableau de suivi de résultats est donné en annexe. Le tableau de suivi et de bilan est réalisé sous un format Excel

➤ PV de recette :

Un PV de la phase de test doit être établi. C'est le document clôturant la recette, il fait un état précis de la qualité de l'application ou du module livré. Il comporte usuellement les informations suivantes :

- L'intitulé de la recette
- La référence de la recette afin de l'identifier de façon unique
- Les participants à la recette
- Le but et travaux de la recette
- La date de l'établissement du Procès Verbal de recette
- L'avis de conformité
- Les réserves

4 Les points clés de la recette fonctionnelle

- Beaucoup de rigueur dans le déroulement des tests, accompagnée d'échanges structurés avec les équipes de développement,
- Une extrême vigilance dans la gestion des environnements de tests à la fois au niveau des données utilisées et des versions du logiciel installées
- Le respect d'une forte structuration dans les échanges avec les maîtres d'œuvre et/ou les équipes de développement,
- Une très bonne compréhension des enjeux métiers au moment de la caractérisation des anomalies,
- Mise en place d'une communication simple avec la MOE concernée afin de qualifier une anomalie de livraisons, les solutions proposées et dans suivre leur achèvement en fonction des contraintes de planning et de priorité du projet.
- Le respect strict d'une totale impartialité dans le jugement de la gravité des anomalies
- Une planification très détaillée de la phase d'exécution,
- Un suivi rapproché de l'exécution des tests comme du stock d'anomalies afin d'éviter les dérives.

5 Les outils de test utilisés pour la mise en œuvre de la stratégie de recette

Dans le dernier point de cette partie, de ce mémoire, nous nous intéressons à l'outil utilisé pendant la réalisation de notre projet de recette.

Une méthodologie de test, est un ensemble de règles et de démarches adoptées à travers un logiciel permettant d'effectuer certaines tâches dans le but de détecter les erreurs d'un programme informatique.

Il existe de nombreux logiciels pour faciliter la réalisation des recettes applicatives, chaque méthodologie il faut lui associer un outil de test.

L'équipe des consultants du cabinet « Salihane-Consulting » (intervenant dans le domaine de la recette fonctionnelle), ont choisi de s'outiller avec Quality Center, ils disposent de connaissances approfondies dans l'utilisation de l'outil, Il est principalement mis en œuvre au cours des missions effectuées chez leurs clients, pour plusieurs applications.

5.1 Présentation de Quality Center

L'outil Quality Center, est le plus pratiqué par les consultants parce qu'il permet de répondre au besoin du client grâce à ses différents modules, ses avantages, et toutes ses fonctionnalités attendues.

L'outil s'inscrit en phase opérationnelle des tests pour :

- Structurer la formalisation des tests,
- Faciliter la communication entre les acteurs,
- Gérer les anomalies,
- Produire un suivi de l'avancement de la préparation et de l'exécution des tests.

Il permet :

- La capitalisation,
- L'archivage des documents de tests,
- La traçabilité de la couverture de tests,
- La traçabilité de l'exécution des tests.

Quality Center est un outil qui sert de support pour la mise en œuvre des activités de tests et recette dans le cycle de vie des projets et de maintenance.

- o Utilisation en mode projet:

Dans la phase conception, l'outil permet de :

- De référencer les exigences à tester dans le module Requirements
- De définir les cas de test dans le module Test Plan
- D'établir une traçabilité entre les exigences et les cas de test
- D'organiser les campagnes de test dans Test Lab

Dans la phase exécution, l'outil permet:

- D'enregistrer les résultats d'exécution dans le module Test Lab
- De déclarer les anomalies et suivre leur correction dans le module Defects

Dans la phase clôture, l'outil permet:

D'extraire des mesures sur les résultats des tests pour réaliser le PV de recette et le bilan de test.

Dans l'ensemble des phases, l'outil permet:

- De générer des reporting qui peuvent inclure :
- Des éditions de rapports
- Des graphiques statistiques pour le suivi des activités

L'utilisation de Quality Center permet aussi de réaliser les parties suivantes des livrables de test directement avec l'outil :

- Les cahiers de conception et d'exécution des tests
- Les fiches d'anomalies
- Les éléments sur l'avancement des tests à intégrer dans le reporting
- Les résultats des tests à insérer dans les PV de tests/recette
- Les éléments quantitatifs sur les tests à intégrer dans le bilan de test

- Utilisation en mode maintenance :

Lors de la maintenance des applications, Quality Center est utilisé pour gérer les versions successives des plans de test pour les nouvelles versions de l'application.

Dans la phase conception des tests, le cahier de conception des tests de la version précédente est adaptée en fonction des évolutions de l'application. Les adaptations peuvent comprendre :

- La mise à jour des exigences (ajout, modification, suppression de fonctionnalités)
- La sélection des exigences à couvrir dans le cadre de l'évolution :
 - > Nouvelles exigences
 - > Exigences existantes impactées par l'évolution
 - > Exigences non impactées mais dont il faut vérifier la non régression
- La mise à jour des cas de test
- La définition des campagnes de test et des scénarios en fonction des exigences couvertes, et des particularités de la nouvelle version.

5.2 Les modules du Quality Center :

Avant de citer l'ensemble de ces modules et les leurs fonctionnalités, nous vous présentons une capture d'écran (une fenêtre du logiciel Quality center, indiquant les éléments clés de cet outil).

Sur cette fenêtre il y'a quatre éléments essentiels :

- La barre latérale de Quality Center : elle permet l'accès aux modules de l'outil.

- La barre d'outils commune: elle permet l'accès aux outils communs, la documentation et des ressources supplémentaires. Elle est localisée en haut à gauche de l'écran.
- Le Menu des modules : affichage des noms de menus dans lesquels vous choisissez des commandes du module de l'outil Quality Center.
- La barre d'outils de module : elle fournit des boutons pour les commandes d'utilisation les plus fréquentes dans le module de l'outil Quality Center.

Ci-dessous l'écran de Quality Center :

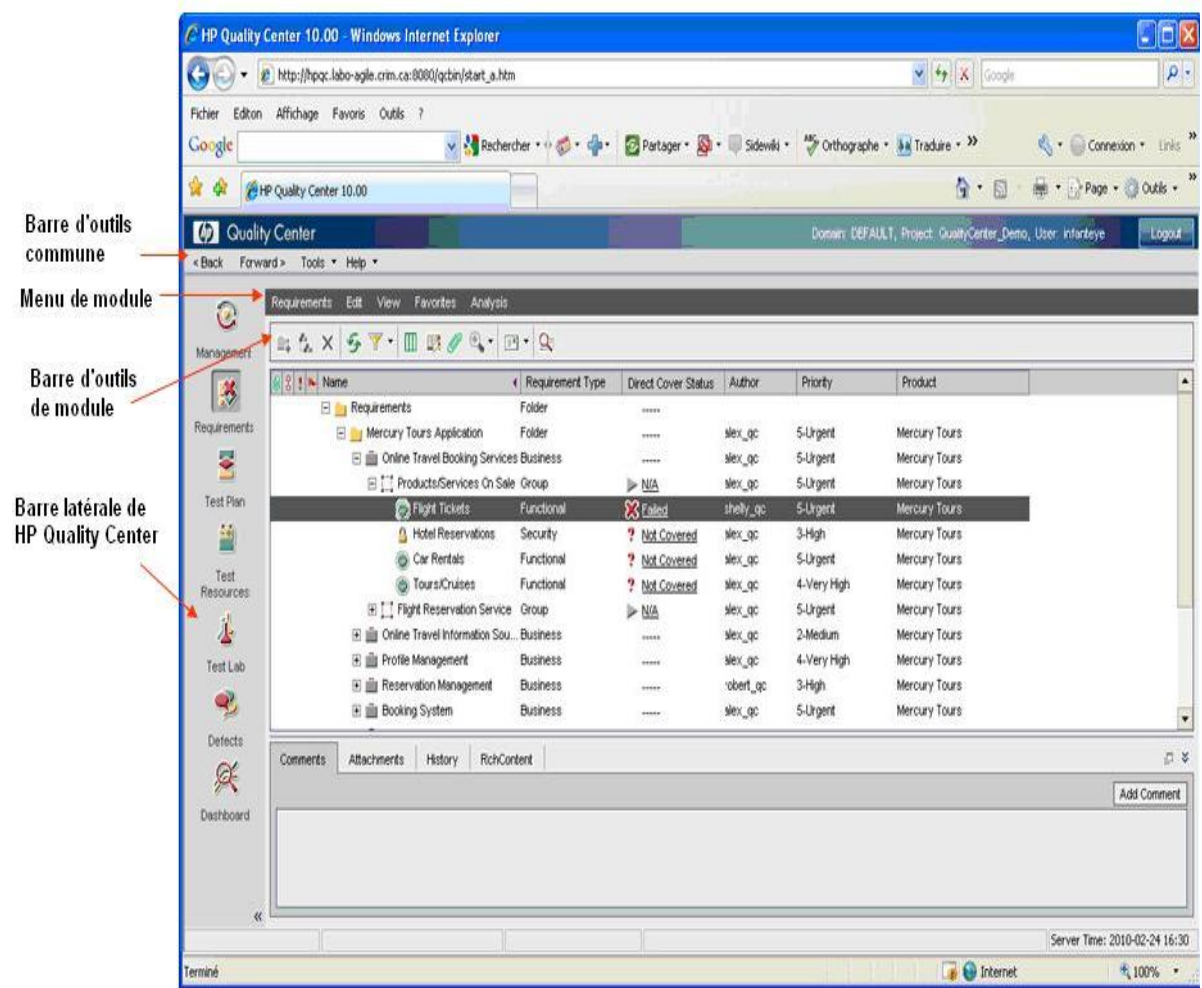


Figure 12: Quality Center⁶⁰

60

http://laboagile.org/web/guest/wiki;jsessionid=B8AFB09ED6C45951AF221D1E0DE52A98?p_p_id=54_INSTANCE_m2M9&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-2&p_p_col_count=1&_54_INSTANCE_m2M9_struts_action=%2Fwiki_display%2Fview&_54

5.3 Les modules de Quality Center

Voici les modules complets de cet outil, leurs avantages et fonctionnalités attendues:-(cette présentation s'appuie sur ma mission de stage) :

5.3.1 Module de Gestion

Ce module permet d'effectuer l'étape de la planification et l'organisation de la recette, identifier les mises à jour d'une application ou d'un projet, suivre également l'avancement et la qualité de la recette, ce qui permet de révéler rapidement tout dépassement de planning.

5.3.2 Module Exigences

Il permet de gérer l'étape de la définition des exigences de test ; identifier les différentes exigences de test :

- l'Intégration de l'application dans le SI,
- l'Installation,
- la documentation Conformité fonctionnelle (règles de gestion),
- les tests de charge et de performance,
- la sécurité,

Une exigence doit être : Documentée, bien précise et interprétée, rattachée à des mises à jour et aux spécifications correspondantes.

5.3.3 Module Plan de test

Il permet d'effectuer l'étape de la conception des tests : un test doit être bien précis et relié à une exigence de test, et renseigner précisément le champ « Description

5.3.4 Module Exécutions des tests

Il permet la préparation de l'étape de l'exécution de la recette ; créer un scénario pour chaque processus métier)

5.3.5 Module Anomalies

La gestion des anomalies est effectuée avec ce module qui permet de créer et suivre les anomalies détectées lors de la recette du projet. Une anomalie détectée doit être saisie. Il faut déterminer le type de l'anomalie. Au moment de la création d'une anomalie, il nécessaire de renseigner le champ « Description » avec des informations sur le contexte du problème, le scénario de reproduction du problème, le résultat observé, le résultat attendu.

[INSTANCE_m2M9_nodeName=Accueil&_54_INSTANCE_m2M9_title=4.2.02+La+Fen%C3%AAtre+de+HPQC](#)

5.3.6 Module Tableau de bord

Le bilan de la recette est géré par ce module, il permet d'analyser les anomalies et les résultats obtenus après l'exécution des campagnes de test, restituer une vue synthétique des résultats des tests et ensuite formaliser les conclusions associées.

6 Lexique spécifique relatif au test

Lexique spécifique relatif aux tests de logiciels : [1, Glossaire CFTL/ISTQ]

Assurance qualité : partie de la gestion de la qualité visant à fournir l'assurance que les exigences qualité seront atteintes [ISO 9000]

Automatisation de l'exécution des tests : Utilisation d'un logiciel (ex. : outil de capture /relecture) pour maîtriser l'exécution des tests, comparer les résultats obtenus aux résultats attendus, mettre en place les pré-conditions de tests, et d'autres commandes de test et de reporting sur les tests

Cas de test bloqué : cas de test ne pouvant être exécuté parce que les pré-conditions pour son exécution ne sont pas réalisées.

Cycle de vie logiciel : Une période temporelle qui commence lorsque un produit logiciel est conçu et se termine lorsque le logiciel n'est plus disponible à l'usage. Le cycle de vie logiciel inclut typiquement une phase de mûrissement, une phase d'exigences, une phase de conception, une phase d'implémentation, une phase de test, une phase d'installation et livraison, une phase d'opération et de maintenance, et parfois une phase de retrait. Note : ces phases peuvent se recouper ou être exécutées de façon itérative.

Déboguer : le processus de trouver, analyser et éliminer les causes de défaillance dans les logiciels.

Défaut : une imperfection dans un composant ou un système qui peut conduire à ce qu'un composant ou un système n'exécute pas les fonctions requises, par exemple une instruction ou une définition de données incorrecte. Un défaut, si rencontré lors de l'exécution, peut causer la défaillance d'un composant ou d'un système.

Défaillance : Écart constaté du composant ou système par rapport au livrable, au service ou au résultat attendu [d'après Fenton]; Incapacité d'un système ou d'un composant d'exécuter une fonction requise dans les limites spécifiées. Une défaillance peut être produite quand un défaut est rencontré

Données de tests : Donnée existante (ex. : dans une base de données) avant qu'un test ne soit exécuté et qui affecte ou est affectée par le composant ou système en test.

Exigence : une condition ou capacité requise par un utilisateur pour résoudre un problème ou atteindre un objectif qui doit être tenu ou possédé par un système ou composant pour satisfaire à un contrat, standard, spécification ou autre document imposé formellement [d'après IEEE 610]

Exigence fonctionnelle : une exigence qui spécifie une fonction qu'un composant ou système doit remplir [IEEE 610]

Fonctionnalité : la capacité d'un produit logiciel à fournir des fonctions qui répondent à des besoins explicites ou implicites quand le logiciel est utilisé sous des conditions spécifiées [ISO 9126]

Fuite mémoire : une défaillance d'accès à la mémoire causée par un défaut dans la logique d'allocation dynamique de l'espace de stockage d'un programme. Cette défaillance fait que le programme ne libère pas la mémoire quand il a fini de l'utiliser, causant au bout du compte la défaillance de ce programme et/ou d'autres processus concurrents par manque de mémoire.

Gestion d'incident : le processus de reconnaissance, d'investigation, d'action et de traitement des incidents. Il implique l'enregistrement des incidents, leur classification et l'analyse de leur impact [d'après IEEE 1044].

Jalon : moment particulier dans un projet auquel des livrables définis (intermédiaires ou non) et des résultats doivent être prêts.

Modèle de développement itératif : un modèle de cycle de développement où le projet est séparé en un nombre d'itérations (souvent nombreuses). Une itération est une boucle complète de développement résultant en une livraison (interne ou externe) d'un produit exécutable, un sous-ensemble du produit final en développement, qui grandit d'itération en itération pour devenir le produit fini.

Modèle du cycle de vie : un cloisonnement de la vie d'un produit ou d'un projet en plusieurs phases.

Maintenance : modification du produit logiciel après livraison pour corriger des anomalies ou améliorer les performances ou d'autres attributs, ou adapter le produit à un environnement modifié [IEEE 1219]

Maturité : (1) La capacité d'une organisation par rapport à la rentabilité et l'efficacité de ses processus et pratiques de travail. Voir aussi Capability Maturity Model, Test Maturity Model. (2) La capacité du produit logiciel à éviter des défaillances suite à la présence de défauts dans le logiciel. [ISO 9126] Voir aussi fiabilité.

Modèle de maturité : un ensemble structuré d'éléments qui décrivent certains aspects de la maturité d'une organisation, et aident dans la définition et la compréhension des processus d'une organisation. Un modèle de maturité fournit souvent un langage commun, une vision partagée et un cadre pour prioriser les actions d'amélioration.

Niveau de maturité : Degré d'amélioration des processus. Ce degré est évalué à travers un ensemble prédéfini de domaines de processus dans lequel tous les objectifs de cet ensemble sont atteints.

Non conformité : non réalisation d'une exigence spécifique [ISO 9000]

Outil de débogage : un outil utilisé par les programmeurs pour reproduire les anomalies, examiner l'état des programmes et trouver les défauts correspondants. Les outils de débogage permettent aux programmeurs d'exécuter un programme pas à pas, d'arrêter un

programme à une instruction du programme et de fixer et examiner les variables du programme.

Plan de tests projet : un plan de tests qui couvre typiquement plusieurs niveaux de tests. Voir Plan de test maître.

Qualité logicielle : la totalité des fonctionnalités et caractéristiques d'un produit logiciel qui influent sur sa capacité à satisfaire des besoins déclarés ou implicites [d'après ISO 9126]

Résultat attendu : le comportement prédit par les spécifications, ou par d'autres sources, du composant ou système, dans des conditions spécifiées

Test fonctionnel : test basé sur une analyse des spécifications d'une fonctionnalité d'un composant ou système. Voir test boîte noire.

Tests des fonctionnalités : le processus de test pour évaluer les fonctionnalités d'un produit logiciel

Technique boîte noire : voir technique de conception de tests boîte noire

Test Boîte Noire : tests, fonctionnels ou non fonctionnels, sans référence aux structures internes du composant ou du système.

Tableau de bord : Une représentation de mesures dynamiques des performances opérationnelles d'une organisation ou d'une activité. On utilise des métriques représentées par des métaphores telles que des alarmes visuelles, des compteurs et d'autres dispositifs ressemblant à ceux du tableau de bord d'une voiture. Le but étant que les effets des événements ou activités puissent être facilement compris et reliés à des objectifs opérationnels. Voir aussi Tableau de bord d'entreprise, bilan.

Technique de conception de tests boîte noire : procédure documentée pour élaborer et sélectionner des cas de tests basés sur une analyse des spécifications, soit fonctionnelles soit non-fonctionnelles, d'un composant ou système sans faire référence à ses structures internes.

Tests d'intégration : tests effectués pour montrer des défauts dans les interfaces et interactions de composants ou systèmes intégrés. Voir aussi tests d'intégration de composants, tests d'intégration système

Test de maintenance : test des modifications d'un système opérationnel ou de l'impact d'une modification d'environnement sur un système opérationnel

Test méthodique: test basé sur un ensemble, de tests standard, par exemple, un standard qualité, ou un ensemble de cas de tests généralisés.

Tests négatifs : tests dont l'objectif est de montrer qu'un composant ou système ne fonctionne pas. Les tests négatifs sont liés à l'attitude des testeurs plutôt qu'à une approche spécifique des tests ou une technique de conception des tests spécifique .

Test de performance : le processus de test pour déterminer les performances d'un produit logiciel. Voir test de rendement.

Test aléatoire : technique de conception de tests boîte noire où les cas de tests sont sélectionnés, par exemple avec un algorithme de génération pseudo-aléatoire, pour correspondre à un profil opérationnel. Cette technique peut être utilisée pour tester les attributs non-fonctionnels tels la fiabilité et les performances.

Test de régression : 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 suites des modifications effectuées. Ces tests sont effectués quand le logiciel ou son environnement est modifié.

Test de fiabilité : le processus de tests pour déterminer la fiabilité d'un produit logiciel.

Test basé sur les exigences : une approche des tests où les cas de tests sont conçus sur base des objectifs de tests et conditions de tests déduites des exigences, p.ex. tests qui exercent des fonctions spécifiques ou examinent des attributs non-fonctionnels tels la fiabilité ou l'utilisabilité.

Conclusion

Pour Garantir la qualité des livrables, s'assurer que le projet est dans la bonne direction et qu'il développe les bonnes fonctionnalités, il faut cependant : « Définir une stratégie de recette et la mettre en œuvre ».

Et pour la réalisation de cette recette, il faut des méthodes et des outils capables de gérer la complexité grandissante du système d'informations.

Le choix est porté plutôt sur des méthodes d'automatisation et d'agilité, une révolution relativement récente, l'outillage associé est maintenant disponible sur le marché y compris dans le secteur open source, une offre considérable de solutions logicielles, des référentiels de la gestion des tests, et toutes les nouvelles tendances de la gestion de projets informatiques à forte valeur ajoutée.

Les méthodes traditionnelles sont laborieuses, coûteuses, peu de productivité et laisse passer les BUGS. Prenons l'exemple du cycle « V », il est encore utilisé comme étant un modèle standardisé, mais il manque de souplesse, souvent une stricte application de cette méthode forcerait à reprendre le cycle depuis le début, en intégrant au niveau fonctionnel les remontées d'ordre technique, ce qui implique des délais conséquents.

C'est pour cela que d'autres méthodes sont apparues : Les méthodes Agiles qui changent les étapes de la conception, de la programmation, et de la validation ; contrairement aux méthodes classiques, on voit concrètement l'évolution du projet car après chaque itération, les utilisateurs peuvent visualiser un « bout » de projet qui fonctionne, ça brise l'effet tunnel des méthodes classiques.

Certes aujourd'hui les nouvelles pratiques gagnent plusieurs directions informatiques, elles ont été déployées avec succès, Les équipes du projet adoptent ces outils et ces méthodes, les avantages se multiplient, un apport considérable : une maturité technologique, organisationnelle, budgétaire.

Par ailleurs, l'expérience révèle que lors de la transition vers ces méthodes, des difficultés apparaissent : au delà de la pratique de meilleures méthodes et meilleurs outils du marché, il y a les personnes qui vont devoir les utiliser :

Effectivement, cette évolution représente des difficultés à prendre en compte dans la constitution des équipes de test ; de nouveaux rôles, nouvelles responsabilités et nouveaux défis. Il faudra impliquer dans les équipes de test des spécialistes capables d'appréhender toutes les démarches de ces nouvelles méthodes et outils.

Et ce problème de pénurie de compétences engendre des frais et délais considérables pour former les équipes de développement aux nouveaux outils et les bonnes pratiques vu leurs complexité.

Enfin, « Penser à tester vos logiciels », de manière manuel ou automatique, ou Agilité, le « le Test », une phase très importante dans le projet, il ne s'improvise pas, il faut le planifier

dès la phase de pré-production avec une forte précision des fonctionnalités qui seront offertes aux utilisateurs finaux.

Ainsi le résume Eric Riou Du Cosquer: «Projet Agile ou pas, il n'y a qu'un test valable, celui qui permet d'augmenter la satisfaction des utilisateurs et la rentabilité des développements».

Bibliographie

La présente bibliographie thématique est arrêtée le 30 octobre 2014. Elle contient les documents nécessaires ayant permis d'alimenter et de couvrir tout le domaine abordé dans ce projet de mémoire.

Les références bibliographiques sont précédées d'un chiffre entre crochets qui correspond à l'ordre de classement des références dans la bibliographie. Chaque référence est suivie d'un résumé.

Les références bibliographiques ci-dessous sont numérotées et classées de manière thématique.

Les thèmes sont organisés de la façon suivante :

- Tests et recette fonctionnelle
- Méthodes et Outils de tests des logiciels
- la veille concurrentielle

D'autres ressources consultées (non citées dans le corps du mémoire)

- Industrialisation et automatisation des tests
- Méthodes Agiles
- Le métier des consultants
- Témoignages sur les méthodes agiles

Test et recette fonctionnelle

[1] Glossaire CFTL/ISTQ, International Software Testing Qualification Board,

Des termes utilisés en tests de logiciels, Traduction française de la Version 2.2F, produite par 'Glossary Working Party' en date du 9 Août 2012. Page 1 de 94. Editeur : Erik van Veenendaal

http://www.cftl.fr/fileadmin/User/Syllabi/Glossaire_des_tests_de_logiciel_-_2_2_F_P1.pdf

[2] DRECQ Vincent, "Tests et recettes dans les projets Système d'Informations : une étape délicate". FICHE THÉMATIQUE, JUILLET 2010, ConseilOrga, Direction de projets et programmes THÉMATIQUE.

<http://www.conseilorga.com/Pages/Recette-test-set-qualification-informatique.aspx>

Extrait du livre : « Pratiques de management de projet », 40 outils et techniques pour prendre la bonne décision ». 2014 - 240 pages EAN13 9782100709984.

Dans ce livre il y'a 40 fiches regroupées en sept chapitres, permettent de découvrir des méthodes concrètes et pragmatiques pour diriger des projets. L'auteur parle des méthodes simple et efficaces pour prendre de bonnes décisions, il propose un tableau de différents outils, qui permet de choisir l'outil le plus adapté au besoin ? ».

[3] FELIX Patrick, "Test et Validation du Logiciel", McInfo4_ASRTests, Janvier 2009
IUT – Bordeaux 1.

Un cours sur la Validation, Vérification & Test des logiciels :

Le test de conformité de systèmes réactifs, Le test fonctionnel de logiciel, Le test structurel, Le test dans un projet logiciel, la stratégie de test.

http://dept-info.labri.fr/~felix/Annee2008-09/S4/McInfo4_ASRTests/1.pdf

[4] SEVRE Cathy, Mémoire CNAM, « Réalisation d'une recette fonctionnelle outillée avec Quality Center et Quick Test Professional ». Soutenu le 21/06/2011. LORTHIOIR2005, "Les tests de non-régression en développement orienté objet (mise en œuvre de l'outil Rational Robot en environnement I2EE).

<http://dumas.ccsd.cnrs.fr/dumas-00692443/document>

[5] ZAITOUN Marc. "La chasse aux bugs, Quand la théorie aide la pratique". Informatique Théorique 1, nov. 2013, p 5

Un cours sur les exemples des bugs aux conséquences désastreuses.

<http://www.labri.fr/perso/zeitoun/enseignement/13-14/Conf-IT1-nov-13.pdf>

Méthodes et Outils de tests des logiciels

[6] BORLAND, Livre blanc « RÉUSSIR L'AUTOMATISATION DU PROCESSUS DE TEST FONCTIONNEL »

http://www.labriqueterie.fr/admin/presse_uploads/functional_testing_whitepaper.pdf

Dans ce livre blanc, l'auteur fournit des informations pratiques et précises sur l'expérience d'entreprises ayant réussi à automatiser leur processus de test fonctionnel, en évoquant la question du rôle stratégique des tests fonctionnels dans l'entreprise moderne, les enjeux et surtout quelle est meilleure approche pour garantir le succès des initiatives d'automatisation

[7] Vickoff Jean-Pierre, « METHODE AGILE, LES MEILLEURES PRATIQUES COMPREHENSION ET MISE EN OEUVRE », livre publié le 02/2009, 224 pages, ISBN: 978-2912843074.

Ce livre, propose, de sélectionner parmi 16 techniques fondamentales des méthodes agiles, celles dont l'usage permettra d'améliorer immédiatement la performance des projets et la qualité des applications. L'Agilité englobe les aspects organisationnels, technologiques et sociologiques de l'évolution afin de proposer une réponse adaptée à sa complexité croissante.

<http://www.lmet.fr/GSWeb/lmet.gswa/5/dr/CSHOPDirectAccess/getProduct/PK=636F64653D323332363739?HPLResultsID=70726F64756374734C697374547970652D31367C626173654C69737449442D7365617263684D6F64656C7C76616C75653D7669636B6F66667C&gswsid=CF946A531DB900C5BC7B7678A476DF4C0A951F2B>

[8] NEUMANN Idriss, Article: « Présentation des méthodes agiles et Scrum », publié le 6 décembre 2012 - Mis à jour le 13 juillet 2014, sur le site Developpez.com:

http://ineumann.developpez.com/tutoriels/alm/agile_scrum/#LV

Cet article est rédigé par Idriss NEUMANN dans de sensibiliser les consultants et futurs informaticiens à la gestion de projet à l'aide des méthodes agiles. Elle a aussi pour but de présenter leurs avantages et inconvénients par rapport à d'autres méthodes de gestion de projet à l'exemple du cycle en V.

[9] RAPPORT DU STANDISH GROUP, "The Chaos Report (1994)", Par Fabrice AIMETTI le vendredi, 19 février 2010.

<https://cs.nmt.edu/~cs328/reading/Standish.pdf>

[10] Eric RIOU Du COSQUER], Trésorier, CFTL (Comité Français du Test Logiciel)

L'article : « Ne confondez pas « Test Agile » ou « Test dans un projet agile », rédigé le 13/12/12, sur le Journal du Net :

Dans cet article l'auteur livre tous les principes fondamentaux et les incontournables du test logiciel, tels ceux recensés par les différents Syllabi de l'ISTQB, se retrouvent sur un projet Agile. Faut-il parler de « Test Agile » ou de « Test dans un projet Agile » ? Une fausse

question selon l'auteur: il n'y a qu'un test valable, celui qui permet d'augmenter la satisfaction des utilisateurs et la rentabilité des développements !

<http://www.journaldunet.com/solutions/expert/52988/ne-confondez-pas--test-agile---ou---test-dans-un-projet-agile.shtml>

[11] DALISYS, blog du test logiciel industrialisé, «Tests fonctionnels automatisés : 5 facteurs clés de succès», publié le 13 octobre 2013.

La décision de passer à l'automatisation des tests fonctionnels , n'est pas une garantie en soi de la réussite d'une campagne de tests. L'auteur, nous nous livre son point de vu sur ce sujet, et propose 5 facteurs clés de succès d'une démarche d'automatisation des tests fonctionnels :

<http://www.dalisy.com/blog/category/bonnes-pratiques-test-logiciel/>

La veille concurrentielle

[12] CHARTRON Ghislaine. La veille dans le contexte de l'enseignement supérieur et la recherche. 61p. Support de cours.

Ce cours donne un aperçu général sur la veille et présente l'étude de cas sur la cellule de veille de l'INRP.

[13] AFNOR. Norme XP X 50-053 : Prestations de veille – Prestations de veille et prestations de mise en place d'un système de veille. Paris, Afnor, 1998. 23p. ISSN : 0335-3931

La norme expérimentale définit les différents termes liés à la veille et les différentes caractéristiques du processus de veille. Elle définit également les prestations de veille et leur méthode de réalisation.

[14] LESCA Humbert, Veille stratégique, concepts et démarche de mise en place dans l'entreprise.

Guide pour la pratique de l'information scientifique et technique. [En ligne] Ministère de l'Education Nationale, de la Recherche et de la Technologie. 1997. 27 p.

<http://veille-strategique.eolas-services.com/docs/plaquette-20418.pdf>, (consulté le 02/10/2010).

L'article présente le concept de veille stratégique ainsi qu'une démarche pour installer un dispositif de veille stratégique dans l'entreprise.

[15] GFFII, Livre Blanc: "Nouveaux usages de la veille - 5 pratiques en émergence",

Le groupe de travail « Intelligence Economique et Economie de la Connaissance » du GFII traite des outils et solutions d'accès et de traitement de l'information. Pour ce nouveau livrable destiné à la fois aux clients de la veille, comme aux fournisseurs de solutions du marché de l'information, le groupe de travail a souhaité apporter un double éclairage : une

étude de l'art sur les pratiques de veille actuelles et une approche prospective qui permette de dégager des tendances fortes dans ces pratiques.

<http://www.gfii.fr/uploads/docs/Livre%20blanc%20Nouveaux%20usages%20de%20la%20veille.pdf>

http://formation-e-eputation.fr/isabellegonon/veille_site_web_2012/co/veille_entreprise.html

[16], MIAUX Jean-François, Mise en œuvre d'une activité de veille: Le cas de Réseau Ferré de France. MEMOIRE Titre professionnel Niveau 1, INTD 2010, p 68.

http://memsic.ccsd.cnrs.fr/mem_00575043

[17], Mouvement WALLON pour la qualité, « Le BENCHMARKING ; se comparer pour s'améliorer »

<http://www.mwq.be/servlet/Repository/?ID=3182>

[18], Glossaire marketing :

<http://www.definitions-marketing.com/Definition-Benchmark>

D'autres ressources consultées (non citées dans le corps du mémoire)

- Industrialisation et automatisation des tests

[19], Article: "Automatisation des tests fonctionnels : les bonnes questions à se poser"

-Source: le blog du test logiciel industrialisé:

<http://www.dalisys.com/blog/automatisation-des-tests-fonctionnels-les-bonnes-questions-a-se-poser/>

- Méthodes Agiles

[20] Article : VITTORIO CAPELLANO « L'Agilité du point de vue des testeurs », Revue: "Génie logiciel" N° 107 en Décembre 2013.

Pour savoir plus : <http://www.acial.fr/images/blog/201312ArticleGL-VCO.pdf>

[21], -Livre : Claude Aubry, « SCRUM, le guide pratique de la méthode Agile la plus populaire ; ETUDES, DEVELOPPEMENT, INTEGRATION »

Ce livre présente les pratiques agiles et donne des pistes pour adapter l'agilité au contexte des organisations. Il s'adresse à tous ceux qui sont impliqués dans le développement logiciel, et qui souhaitent s'initier aux méthodes agiles. L'auteur explique l'importance de la notion d'équipe, avec les rôles emblématiques de ScrumMaster et Product Owner. Il montre comment organiser le travail de l'équipe dans un backlog pour produire des versions à chaque sprint en suivant un cérémonial qui a fait la preuve de son efficacité. Le lien vers ce livre :

<http://www.dunod.com/informatique-multimedia/developpement/methodes-modelisation-uml/ouvrages-professionnels/scrum>

[22], -Article: Antoine Crochet Damais, "Les méthodes Agiles, une autre conduite des projets informatiques"

l'auteur évoque les avantages des méthodes Agiles, et souligne les quatre valeurs et les grands principes fondamentaux de la méthode Agile développés par le "Manifeste Agiles" :

<http://www.journaldunet.com/developpeur/algo-methodes/conseil/les-methodes-agiles-une-autre-conduite-des-projets-informatiques.shtml>

[23], Article, Florent Lothon " Introduction aux méthodes agiles et Scrum, comprendre les principes et le fonctionnement des méthodes agiles", Date de publication : juin 2013.

Cet article est une introduction sur la méthode agile: D'où vient-elle ? Comment s'applique-t-elle concrètement ? C'est un guide pour comprendre les principes fondamentaux d'une approche agile, ses fonctionnements et ses enjeux au sein des entreprises de développement : <http://www.agiliste.fr/fiches/introduction-methodes-agiles/>

Le métier des consultants :

[24], Livre de Didier Durandy « Le bon, la brute et le consultant ».

Dans ce livre, l'auteur revient sur son histoire et sur son parcours de consultant en France, et à l'international pour des hommes clés qui font appel à lui, souvent dans l'urgence. Son ouvrage est un guide de conseils pour les consultants dont les missions sont parfois périlleuses, comme le laisse sous-entendre son sous-titre «Kit de survie pour réaliser des missions en terrain miné » :

<http://www.widoobiz.com/confidences-d-entrepreneurs/le-bon-la-brute-et-le-consultant-le-guide-cle-pour-les-c>

[25], Un guide : « Comment choisir un consultant ? Risque psychosociaux » :

Rédigé par la DGT avec l'appui de l'Agence nationale pour l'amélioration des conditions de travail (ANACT), l'Institut national de recherche et de sécurité (INRS) et la Caisse nationale d'assurance maladie (CNAM-TS) :

Ce guide comporte des conseils pour vous aider à définir votre besoin et clarifier vos attentes vis-à-vis du consultant, une grille d'aide au choix du consultant, des recommandations pour construire la « feuille de route » du consultant, une annexe qui permet de situer l'offre d'un consultant au regard des différents niveaux de prévention. Pour aller plus loin :

<http://www.travailler-mieux.gouv.fr/Guide-Comment-choisir-un.html>

Témoignages sur les méthodes agiles

-Article, Cyrille Chausson, Editor-in-Chief - Tech Target – LeMagIT « Tests agiles : une certification se prépare pour dynamiser un marché qui se transforme », publié le 23 janv. 2014.

Dans cet article l'auteur donne des chiffres clés et quelques témoignages sur l'émergence de l'agilité dans les tests :

<http://www.lemagit.fr/actualites/2240213041/Tests-agiles-une-certification-se-prepare-pour-dynamiser-un-marche-qui-se-transforme>

-Vidéo : « "Oubliez le cahier des charges", "soyez agiles" »

Résumé : L'auteur de la vidéo, "Anthony Bleton de Novius » explique très bien en 4 minutes en quoi l'approche agile se distingue de l'approche traditionnelle :

<http://www.agiliste.fr/fiches/introduction-methodes-agiles/>

-Vidéo : « Méthodes agiles en action »:

Résumé : cette vidéo illustre les méthodes agiles avec Scrum et eXtreme Programming, pour mener un Projet de développement pour la SNCF :

<http://www.agiliste.fr/videos/video-methodes-agiles-en-action/>

Annexes

Annexe 1 « Pour aller plus loin »

Nous vous proposons des adresses des sites internet (sites spécialisés) des grands acteurs des services de la qualité des logiciels et de la gestion de projets, des cabinets de conseil en informatique qui proposent des solutions et les outils. (des acteurs cités dans le corps du mémoire) :

1-Acial :

<http://www.acial.fr/fr/slideshow/126-le-referentiel-tmmi-un-outil-pour-lamelioration-de-lactivite-de-test8.html>

2-ConseilOrga :

<http://www.conseilorga.com/Pages/Recettetestsetqualificationinformatique.aspx>

3-EUROGROUP CONSULTING :

<http://www.eurogroupconsulting.fr/-Une-approche-singuliere->

4- Altea-conseil :

<http://www.altea-conseil.com/consulter-nos-offres.html>

5- Akal-consulting:

<http://www.akal-consulting.com/#!resolution-incident/cd0d>

6-Virage-groupe :

<http://www.viragegroup.com/fr/services/nos-services>

7- Hsc

[:http://www.hsc.fr/services/conseil.html.fr#ouvrage](http://www.hsc.fr/services/conseil.html.fr#ouvrage)

8-Pentalog :

http://www.pentalog.fr/systemes_d_information/testing_systemes_d_information.htm

9- Henix:

<http://www.henix.com/index.php/qui-sommes-nous-76995/presentation>

10-Mc2i : <http://www.mc2i.fr/>

11-Validit :

<http://www.validit.com/Societe.html>

12- ReferTy:

http://www.referty.fr/Presentation-ReferTy_157.html

13- NEO-SOFT :<http://www.neo-soft.fr/>

LES ORGANISATIONS PROFESSIONNELLES :

- CFTL : <http://www.cftl.fr/>
- PMI : <http://www.pmi-france.org/>
- APRIL : <http://www.april.org/association>

BLOGS/CLUB DES CONSULTANTS:

<http://ulrichinaction.blogspot.fr/2013/04/integration-des-tests-fonctionnels.html>

<http://blog-du-consultant.fr/les-septs-galeries-du-consultant-independant/>

<http://blogduconsultant.blogspot.fr/>

<http://www.osaxis.fr/blog/automatiser-ses-tests-fonctionnels-partie-2-2/>

<http://gestiondeprojets.wordpress.com/sabonner-au-blog/>

<http://blog.univ-angers.fr/qsfs/category/test-logiciel/>

<http://t37.net/tag/blogs-twitter-micro-blogging/>

<http://t37.net/preparer-ses-plans-de-tests.html#dans>

<http://t37.net/preparer-ses-plans-de-tests.html>

<http://je-choisis-ma-vie.fr/2012/11/19/ssii-consultants-changez/>

<http://www.acial.fr/club-acial-test-logiciel/>

<http://blog.soat.fr/2012/05/tests-bugs-rocknroll/>

<http://genielogicieletagiles.blogspot.fr/>

Annexe 2 Glossaire des tests des logiciels

Lexique spécifique relatif aux tests de logiciels [1, Glossaire CFTL/ISTQ]

Identification des risques : processus d'identification des risques en utilisant des techniques telles que le brainstorming, les checklist et les historiques de défaillances.

Niveau de risque : L'importance d'un risque est définie par ses caractéristiques impacts et probabilité. Le niveau de risque peut être utilisé pour déterminer l'intensité de test à accomplir. Un niveau de risque peut être exprimé de façon qualitative (p.e. haut, moyen, bas) ou quantitative

Gestion des risques : application systématique de procédures et pratiques aux tâches d'identification, d'analyse, de priorisation et de contrôle des risques.

Robustesse : le degré pour lequel un composant ou système peut fonctionner correctement en présence de données d'entrée invalides ou de conditions environnementales stressantes [IEEE 610], voir aussi tolérance aux erreurs, tolérance aux défauts.

Test de robustesse : test pour déterminer la robustesse d'un produit logiciel.

Sévérité : degré qu'à l'impact d'une anomalie sur le développement ou l'utilisation d'un composant ou système [d'après IEEE 610]

Test : un ensemble d'un ou plusieurs cas de tests [IEEE 829]

Architecte de test: (1) Une personne qui fournit des conseils et donne une orientation stratégique pour une organisation de test et pour ses relations avec d'autres métiers. (2) Une personne qui définit la façon dont les tests sont structurés pour un system donné, en incluant des sujets tels que les outils de tests et la gestion des données de tests.

Automatisation des tests : utilisation de logiciels pour exécuter ou supporter des activités de tests, p.ex. Gestion des tests, conception des tests, exécution des tests ou vérification des résultats.

Cas de test : un ensemble de valeurs d'entrée, de préconditions d'exécution, de résultats attendus et de postconditions d'exécution, développées pour un objectif ou une condition de tests particulier, tel qu'exécuter un chemin particulier d'un programme ou vérifier le respect d'une exigence spécifique

Spécification de cas de test : document spécifiant un ensemble de cas de tests (objectifs, entrées, actions de tests, résultats attendus et préconditions d'exécution) pour un article de tests [d'après IEEE 829] Voir aussi Spécification de tests.

Gestion des données de tests: Processus consistant à analyser les exigences des données de tests, à concevoir les structures des données de tests et à créer et maintenir les données de tests.

Spécification de conception de tests : Document spécifiant les conditions de tests (éléments de couverture) pour un article de test, l'approche détaillée du test et l'identification des cas de tests de haut niveau associés [d'après IEEE 829] Voir aussi Spécification de test.

Environnement de test : Environnement contenant le matériel, les instruments, les simulateurs, les outils logiciels et les autres éléments de support nécessaires à l'exécution d'un test [d'après IEEE 610]

Exécution des tests : Processus consistant à exécuter un test sur un composant ou système en test, en produisant le(s) résultat(s) obtenu(s).

Phase d'exécution des tests : Période de temps dans le cycle de développement logiciel pendant laquelle les composants d'un produit logiciel sont exécutés, et le produit logiciel est évalué pour déterminer si les exigences ont été satisfaites ou non. [IEEE 610]

Calendrier d'exécution des tests : Schéma d'exécution des procédures de test. Les procédures de tests sont incluses dans le calendrier d'exécution dans leur contexte et dans l'ordre où elles doivent être exécutées

Technique d'exécution des tests : Méthode utilisée pour effectuer l'exécution des tests, soit manuellement, soit de façon automatisée

Entrées de test : Données reçues d'une source externe par l'objet de test pendant son exécution. Les sources externes peuvent être matérielles, logicielles ou humaines.

Niveau de test : Groupe d'activités de test organisées et gérées ensemble. Un niveau de test est lié aux responsabilités dans un projet. Les exemples de niveaux de test sont les tests de composants, les tests d'intégration, les tests système et les tests d'acceptation.

Test Manager (Gestionnaire de test)) : Personne responsable du test et de l'évaluation d'un objet de test. Individu qui dirige, contrôle, gère l'organisation et règle l'évaluation d'un objet de test.

Gestion des tests : Planification, estimation, suivi et contrôle des activités de test généralement effectué par un gestionnaire des tests.

Oracle de test : Source utilisée pour déterminer les résultats attendus à comparer avec les résultats obtenus de l'application en cours de test. Un oracle peut être le système existant (comme point de référence), un manuel utilisateur ou la connaissance spécialisée d'un individu, mais ne devrait pas être le code.

Plan de test : Document décrivant l'étendue, l'approche, les ressources et le planning des activités de test prévues. Il identifie entre autres les éléments et caractéristiques à tester, l'affectation des tâches, le degré d'indépendance des testeurs, l'environnement de test, les techniques de conception des tests et les techniques de mesure des tests à utiliser ainsi que

tout risque nécessitant la planification de contingence. Il constitue la documentation du processus de planification de test.

Processus de test : Processus de test fondamental comprenant la planification, la spécification, l'exécution, l'enregistrement et la vérification de l'achèvement.

Stratégie de test : Document de haut niveau définissant, pour un programme, les niveaux de tests à exécuter et les tests dans chacun de ces niveaux (pour un ou plusieurs projets).

Type de test : Groupe d'activités de test dont l'objectif est de tester un composant ou système sur un ou plusieurs attributs liés entre eux. Un type de test est focalisé sur un objectif de test spécifique (ex. : test de fiabilité, d'utilisabilité, de régression, etc) et peut couvrir un ou plusieurs niveaux de test et une ou plusieurs phases de test.

Testabilité : Capacité d'un produit logiciel à permettre le test du logiciel modifié. [ISO 9126] voir aussi Maintenabilité

Testeur : Professionnel techniquement qualifié impliqué à tester un composant ou système.

Test : Processus consistant en toutes les activités du cycle de vie, statiques et dynamiques, concernant la planification et l'évaluation de produits logiciels et produits liés pour déterminer s'ils satisfont aux exigences et démontrer qu'ils sont conformes aux objectifs et détecter des anomalies

Test de cas d'utilisation : Technique de conception de test boîte noire dans laquelle les cas de tests sont conçus pour exécuter des scénarios de cas d'utilisation.

Qualité basée sur l'utilisateur : Vision de la qualité centrée sur la capacité de satisfaire les besoins, désirs et envies de(s) l'utilisateur(s). Un produit ou un service qui ne répond pas à des besoins ne trouvera probablement pas d'utilisateurs. C'est une approche conditionnelle et dépendante du contexte puisque différentes caractéristiques métier requièrent différents niveaux de qualité pour un produit. Voir aussi Qualité basée sur la fabrication, Qualité basée sur le produit, Qualité transcendante, Qualité basée sur la valeur.

Modèle en V : Structure décrivant les activités du cycle de développement logiciel, depuis la spécification des exigences jusqu'à la maintenance. Le modèle en V illustre comment les activités de test peuvent être intégrées dans chaque phase du cycle de développement.

Technique de conception de test boîte blanche : Procédure documentée utilisée pour dériver et sélectionner des cas de tests basés sur une analyse de la structure interne d'un composant ou système

Test boîte blanche : Test basé sur une analyse de la structure interne du composant ou système

Annexe 3 Des solutions logicielles open sources

Ci-dessous des solutions logicielles open sources identifiées :

Checkstyle : Un outil qui permet d'automatiser le processus de vérification du code Java ; l'idéal pour les développeurs qui souhaitent appliquer une norme de codage dans les projets informatiques : <http://checkstyle.sourceforge.net/>

JUnit : Est un cadre simple d'écrire des tests reproductibles. Il s'agit d'un exemple de l'architecture xUnit pour les cadres de tests unitaires : <http://junit.org/>

PHPUnit : est un cadre de tests orientés programmeur PHP. Il s'agit d'un exemple de l'architecture xUnit pour le cadre de tests unitaires : <http://phpunit.de/>

FindBugs : « trouvent des bogues dans les programmes Java ». Un programme qui utilise l'analyse statique pour trouver des bogues dans le code Java. C'est un logiciel libre, distribué sous les termes de la Lesser GNU Public License : <http://findbugs.sourceforge.net/>

CruiseControl : CruiseControl est un logiciel open source et est développé et maintenu par un groupe de dévoués bénévoles. Il comprend des dizaines de plugins pour une variété de contrôle des sources, construire des technologies et des systèmes, y compris les notifications e-mail et la messagerie instantanée. CruiseControl est écrit en Java, mais est utilisé sur une grande variété de projets : <http://cruisecontrol.sourceforge.net/>

ReferTy : « Manage your test » : Une jeune entreprise Française editrice du logiciel ReferTest "Référentiel de test logiciel" destinée aux petites, moyennes et grandes entreprises soucieuses de développer rapidement des applications performantes en s'appuyant sur une méthode de l'industrialisation et l'automatisation des tests fonctionnels. Des consultants engagés à développer un référentiel de test logiciel manuel/ou automatisé qui répond parfaitement aux besoins des équipes de test et validation.
http://www.referty.fr/Presentation-ReferTy_157.html

NEO-SOFT : Créée en 2005, par Laurent FLORENTIN et Soig LE BRUCHEC, NEO-SOFT est une société de service en ingénierie informatique. Parmi ses missions, Néo-Soft, accompagne ses clients dans la stratégie de test de leurs applications ou infrastructures avec le meilleur compromis coût/risque et prend en charge sa réalisation. <http://www.neo-soft.fr/>

ALTEA : « TEST ET QUALIFICATION » : ALTEA Conseil est une société de service informatique spécialisée dans les tests et la qualification logicielle. L'objectif est d'assurer l'aboutissement de projets informatiques et organisationnels complexes en nous appuyant sur notre expertise, nos méthodes, et nos connaissances fonctionnelles.

Altea Conseil intervient sur toutes les phases de spécifications avec les utilisateurs ou l'organisation:

Phase de conception : tests fonctionnels documentaires

Phase de développement : tests unitaires manuels ou automatisés

Phase d'intégration : tests d'intégration fonctionnels et techniques,

Phase de qualification : test de qualification métier (UAT/VABF)

Phase de mise en exploitation : tests d'exploitabilité, tests de fiabilité (plans de reprise)

Phase post-mep : tests de VSR (Vérification du Service Régulier) :

<http://www.altea-conseil.com/consulter-nos-offres.html>

Henix : Cabinet de conseil en Système d'Information. Sa mission principale est d'accompagner les grands comptes en amont en amont et en aval (qualification et déploiement) du cycle de vie logiciel. Ses services s'articulent autour de deux métiers

L'Assistance à Maîtrise d'Ouvrage : (Analyse métier et fonctionnelle, conduite de Projet / PMO, accompagnement du changement

La qualité logicielle : (Conseil et outillage, notamment open source, Qualification fonctionnelle, qualification technique, sécurité applicative, performance :

<http://www.henix.com/index.php/qui-sommes-nous-76995/presentation>