



CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE RÉGIONAL ASSOCIÉ DE VERSAILLES

MÉMOIRE

Présenté en vue d'obtenir le diplôme d'Ingénieur C.N.A.M.

en Ingénierie et Architecture Système et Logiciel

par

M. FRIESS Raphaël

Améliorer le processus de test pour atteindre l'échelon 1 de TPI

JURY

PRESIDENT : M. BARKAOUI Kamel

MEMBRES : M. GEACHAN Émile
M. SCHOUHMANN Michel

Soutenu le

Résumé

Ce document traite d'un projet d'amélioration du processus de test de logiciel réalisé dans l'entreprise D2T dont la principale activité est orientée vers le développement et la mise au point de moteurs thermiques. Pour ce faire, elle développe des logiciels destinés au pilotage des bancs d'essai de moteurs.

Le modèle d'amélioration Test Process Improvement (TPI) a été utilisé pour évaluer le processus courant et ensuite à déterminer les opportunités d'amélioration. Nous nous sommes inspirés des processus vérification et validation de ISO 12207 sur le cycle de vie du logiciel et du standard IEEE 830 pour la gestion des exigences du logiciel.

La réalisation du projet a débuté en décembre 2009 à la suite d'une phase de lancement consistant à définir son organisation. Celle-ci a été réalisée sur la base de préconisations issues du Project Management Body Of Knowledge (PMBOK).

La définition du nouveau processus a été réalisée de manière itérative et incrémentale par les utilisateurs du processus eux-mêmes. Il est ainsi basé sur une capitalisation de leurs compétences.

Abstract

This document deals with a project of software test process improvement realized at the company D2T. The main activity of the company is directed to develop and calibrate internal combustion engines. D2T develops software intended for the engine's testbeds management and control.

The model Test Process Improvement (TPI) was used to estimate the current process and then to define the improvement opportunities. The ISO 12207 verification and validation processes from the software life cycle and the norm IEEE 830 for software requirement management are the basis used to develop the software test process.

The project started in December 2009 following a launch phase consisting in defining its organization. This launch phase was realized on the basis of recommendations originating from the Project Management Body Of Knowledge (PMBOK).

The definition of the new process was realized in an iterative and incremental way by the users of the process, so it is based on a capitalization of their skills.

Remerciements

Je remercie les personnes suivantes :

Mon chef de projet adjoint, M. Adrien AUCLAIR, et la Directrice Qualité de D2T, Mme Martine ROUTHIER, pour leurs conseils et leur soutien pendant le projet et pour la relecture du mémoire.

Ma femme pour son soutien indéfectible tout au long de mon cursus et surtout pour m'avoir donné la motivation de reprendre mes études en m'ouvrant la voie.

Ma fille pour sa compréhension envers le fait que je n'ai pas toujours été disponible autant qu'elle l'aurait désiré.

Mon tuteur, M. Michel SCHOUHMANN, pour son aide dans l'organisation et la rédaction de mon mémoire, et tous les professeurs du CNAM, que j'ai rencontré tout au long de ces années, pour leur contribution à faire de moi ce que je suis à présent.

Enfin, la Direction de D2T de m'avoir accordé leur confiance en me proposant ce projet et l'ensemble de mes collègues pour y avoir participé.

Table des matières

1. Présentation générale	8
2. L'amélioration du processus de test et ses attendus	11
2.1. <i>Présentation de D2T</i>	12
2.1.1. D2T et IFP Énergies Nouvelles	12
2.1.2. Le département Produits	13
2.1.3. Améliorer la qualité des produits, un axe stratégique	15
2.2. <i>Constat sur le processus de gestion des tests</i>	16
2.2.1. Présentation du processus de développement incluant les tests	17
2.2.2. Constat qualitatif du processus existant	18
2.2.3. L'état de l'Art dans le domaine des tests n'est pas pris en compte	20
2.2.4. Dans l'échelle TPI le processus de test existant n'atteint pas le premier échelon	20
2.3. <i>Les attendus à l'issue du projet (objectifs)</i>	21
2.3.1. Introduire les secteurs-clef de l'échelon 1 de TPI	22
2.3.2. Définir une politique de test	22
2.3.3. Gérer les exigences pour définir le périmètre du test	23
2.3.4. Introduire la pratique du bilan des tests pour capitaliser et améliorer en continu	24
2.4. <i>Les exigences projet</i>	24
2.4.1. Le coût	24
2.4.2. Le délai	25
2.4.3. La qualité	25
3. Le processus de test intégré	27
3.1. <i>Présentation du processus de test intégré</i>	28
3.2. <i>Le document « Politique de test »</i>	32
3.3. <i>Le référentiel des exigences et des tests</i>	33
3.3.1. Gestion de configuration	33
3.3.2. Gestion des exigences	34
3.3.3. Gestion des tests	39
3.3.4. Bilan des tests	41
3.4. <i>La formalisation des tests</i>	45
3.4.1. La fiche de description de test (FDT)	45
3.4.2. La fiche de résultats de test (FRT)	50
3.5. <i>Tableau récapitulatif des exigences satisfaites</i>	53
4. Conduite du projet	55
4.1. <i>Phase planification et lancement</i>	56
4.1.1. Acquisition des normes et standards du projet	56
4.1.2. Planification	65
4.2. <i>Réalisation</i>	76
4.2.1. L'activité définition de la politique de test	76
4.2.2. Activité intégration de la politique de test (Accompagnement au changement)	77
4.2.3. Définition du nouveau processus	83
4.2.4. La vérification	87
4.2.5. La validation	89
4.3. <i>Réorganisation du projet et déploiement</i>	93
4.3.1. Situation du projet en octobre 2010	93
4.3.2. Les propositions de réorganisation	94
4.3.3. Planification de la nouvelle organisation	95
5. Bilan	99

Améliorer le processus de test pour atteindre le niveau 1 de TPI

5.1.	<i>Bilan projet</i>	100
5.2.	<i>Apports du projet à l'entreprise</i>	101
5.3.	<i>Retour d'expérience</i>	102
5.3.1.	Ce qui a bien fonctionné	102
5.3.2.	Ce qui aurait pu être amélioré	103
5.4.	<i>Bilan personnel</i>	103
6.	Conclusion générale	105
7.	Annexes	106
7.1.	<i>Les secteurs-clef TPI : description détaillée</i>	106
7.2.	<i>Les modèles de document</i>	110
7.3.	<i>Matrice de traçabilité des exigences</i>	118
7.4.	<i>Tableau des risques projet</i>	120
8.	Glossaire	121
9.	Références	123
9.1.	<i>Bibliographie</i>	123
9.2.	<i>Webographie</i>	124

Liste des figures

Figure 1 : structure de D2T	12
Figure 2 : la pyramide des produits D2T	13
Figure 3 : répartition des produits par type	14
Figure 4 : diagramme organisationnel du département Produits	14
Figure 5 : les chantiers d'amélioration	16
Figure 6 : Cycle de développement des produits	17
Figure 7 : traçabilité exigence/test	23
Figure 8 : les activités de test intégrées au processus de développement	29
Figure 9 : exemple de tableau récapitulatif des tests dans le dossier de validation	31
Figure 10 : zone dédiée à l'identification et la description des exigences fonctionnelles	34
Figure 11 : zone dédiée à l'identification et la description des fonctions	36
Figure 12 : zone dédiée à la gestion des risques produit	38
Figure 13 : zone dédiée à l'organisation des tests	39
Figure 14 : page « couverture et écarts » du bilan	42
Figure 15 : page « anomalies » du bilan	43
Figure 16 : page « budget » du bilan	44
Figure 17 : page retour d'expérience du bilan	45
Figure 18 : ordonnancement des activités de tests	46
Figure 19 : tableau de gestion des versions	47
Figure 20 : éléments constituant un cas de test	48
Figure 21 : exemple de cartouche de FDT	48
Figure 22 : exemple de cartouche de FRT	51
Figure 23 : exemple de tableau de synthèse	51
Figure 24 : exemple de FRT suite à un retest	52
Figure 25 : les phases du projet	57
Figure 26 : structure des secteurs-clefs	58
Figure 27 : structure d'un domaine de processus	62
Figure 28 : domaines de processus de l'ingénierie	64
Figure 29 : composition du comité de pilotage du projet	65
Figure 30 : ordonnancement des diverses phases du projet d'amélioration	67
Figure 31 : Planning macroscopique du projet d'amélioration	69
Figure 32 : organigramme des ressources du projet	73
Figure 33 : planning de l'activité "définir une politique de test"	76
Figure 34 : planning du projet pilote 2	85
Figure 35 : matrice de traçabilité du contenu du projet	87
Figure 36 : application du processus de développement D2T au projet d'amélioration	89
Figure 37 : procédure d'évaluation	89
Figure 38 : planning prévisionnel d'un projet de validation	90
Figure 39 : proposition de réorganisation 1	94
Figure 40 : proposition de réorganisation 2	95
Figure 41 : planning du projet de déploiement	97

Liste des tableaux

Tableau 1: résultats de l'évaluation avant amélioration	21
Tableau 2 : objectifs d'amélioration pour atteindre l'échelon 1 de TPI	22
Tableau 3 : objectifs d'amélioration pour atteindre le niveau d'aptitude 2 pour REQM	24
Tableau 4 : exigence relative au bilan des tests.....	24
Tableau 5 : synthèse des activités et livrable du processus de test.....	28
Tableau 6 : l'échelle de niveaux de test	33
Tableau 7: type de risques produit.....	38
Tableau 8 : exemple de justification d'anomalies non corrigées	43
Tableau 9 : exemple de scénario de test	49
Tableau 10 : Récapitulatif des exigences du projet d'amélioration des processus et leur satisfaction ..	53
Tableau 11: les vingt secteurs-clefs du modèle TPI.....	58
Tableau 12 : répartition des exigences par secteurs-clefs et par niveaux d'approfondissement	60
Tableau 13 : matrice de maturité des tests (source : [8]).....	61
Tableau 14 : structure de découpage du projet.....	67
Tableau 15 : répartition des charges par activité	69
Tableau 16 : rôles et responsabilités des ressources permanentes de l'équipe projet	74
Tableau 17 : rôles et responsabilités des ressources par activité.....	74
Tableau 18 : liste des actions de communication	75
Tableau 19 : les divers types de formalisation de test	79
Tableau 20 : les types de tests communément utilisés chez D2T	80
Tableau 21 : répartition des exigences par projet pilote.....	83
Tableau 22 : bilan coût et délai des projets pilotes 1 à 3	86
Tableau 23 : gestion de configuration - règle d'étiquetage.....	88
Tableau 24 : composition de l'équipe d'évaluation.....	91
Tableau 25 : critères d'évaluation des pratiques	92
Tableau 26 : exemple de bilan d'évaluation	92
Tableau 27: les apports liés au modèle TPI.....	93
Tableau 28 : les apports non liés au modèle TPI.....	93
Tableau 29 : les apports liés au domaine REQM du modèle CMMI.....	93
Tableau 30 : coût du projet de déploiement	97
Tableau 31 : bilan coût (en k€).....	100
Tableau 32 : bilan délai (en mois)	100
Tableau 33 : bilan de la couverture des exigences fonctionnelles (en nombre d'exigences).....	100

1. Présentation générale

D2T, filiale d'IFP Énergies Nouvelles, a centré ses activités dans deux domaines :

- l'Ingénierie de Développement des Groupes Motopropulseurs (voiture, camion, engins agricoles,...),
- l'Ingénierie des Équipements des Moyens d'Essais (IEME), dont l'activité consiste à proposer des solutions clé en main pour bancs d'essais (machines de charge, systèmes d'automatisation,...).

La direction générale a fixé, en 2009, parmi les axes stratégiques, l'amélioration de la qualité des produits. Ceci s'est traduit au sein du département Produit d'IEME par le démarrage de deux chantiers : la gestion des projets et la gestion des tests. Les produits développés, au sein du département, sont à 80 % de type logiciel. Le projet d'amélioration de la gestion des tests m'a été confié en tant que chef de projet Assistance à la Maîtrise d'Ouvrage, AMO. J'ai été choisi pour ma pratique professionnelle en test des logiciels de D2T et pour mes connaissances en génie logiciel acquises au CNAM.

Pour mener ce projet, je me suis appuyé sur le modèle d'amélioration Test Process Improvement (TPI) de la société SOGETI qui permet d'évaluer la maturité d'un processus de test pour ensuite définir un ensemble d'améliorations à réaliser pour faire progresser la maturité du département.

La qualité d'un logiciel en test est tributaire de la couverture des exigences c.-à-d. ce que demande le client qui constitue l'essentiel des données d'entrée du processus de test. Nous sommes convenus d'inclure au périmètre du projet l'amélioration de la gestion des exigences. Pour cela je me suis appuyé sur le domaine de processus Requirements Management du Capability Maturity Model Integration (CMMI).

L'objectif du projet est d'atteindre l'échelon 1 de TPI ce qui suppose :

- d'introduire des pratiques relatives aux préconisations des sept secteurs-clefs de l'échelon 1 de TPI,
- de définir une politique de test,
- d'introduire les pratiques de gestion des exigences,
- d'introduire la pratique de bilan des tests.

À l'issue d'une phase de préparation du projet, le coût de ce dernier est estimé à 222 000 €. Il est majoritairement lié au besoin en ressources humaines correspondant à 2 600 heures. En termes de planification, le projet a été organisé de manière à ne pas excéder une période d'un an (Déc. 2009 à Déc. 2010).

Le nouveau processus de test, obtenu à l'issue du projet, est intégré au processus de développement. Il se concrétise par un ensemble de modèles, d'outils et de documents explicatifs venant en support à des activités à mettre en oeuvre.

Les activités de ce nouveau processus débutent pour certaines très tôt dès l'élicitation¹ des exigences du logiciel à développer. Les activités introduites au processus de développement sont :

- **la gestion des exigences** : les exigences du logiciel à développer sont identifiées et mises sous contrôle,
- **la planification** : une organisation des tests est définie à partir des exigences et d'une stratégie de test qui s'inscrit dans la politique de test de l'entreprise,
- **la conception** : selon la criticité des exigences la formalisation des tests est réalisée de manière plus ou moins stricte,

¹ Élicitation des exigences : activité consistant à rassembler, traiter et de suivre les besoins servant de base à la définition du produit [7]

- **l'exécution** : à la fin de la phase de réalisation du logiciel, les tests sont exécutés selon la planification,
- **le bilan** : une analyse des écarts est réalisée entre les objectifs prévus par la planification initiale et finale pour capitaliser l'expérience acquise.

Les documents et outils, sous forme de documents ou modèles Word et Excel, venant en support à ces activités sont :

- **une politique de test** : elle établit les objectifs généraux de test que le département Produits souhaite atteindre. Elle permet aux chefs de projet de développement logiciel de mieux appréhender les objectifs en termes de tests,
- **un modèle de référentiel des exigences et des tests** : sous la forme d'un fichier Excel, cet outil est l'élément central du processus en permettant d'effectuer le suivi et la maîtrise des exigences et des tests d'un projet de développement de logiciel,
- **des modèles de fiches de description de test et fiches de résultats de test** : ces documents proposent une structure nécessaire à la formalisation des tests.

Pour organiser et par la suite suivre et maîtriser mon projet je me suis appuyé sur les recommandations du Projet Management Body Of Knowledge (PMBOK).

Le projet a été précédé par une étude de faisabilité durant laquelle j'ai défini le déroulement du projet pour le soumettre au responsable du département (client du projet). Durant cette phase, les contraintes de projet ont été précisées et ordonnées afin d'avoir une meilleure visibilité sur l'avancement du projet. Sur cette base une Work Breakdown Structure (WBS) a été réalisée permettant de faciliter la planification et l'estimation de la durée totale du projet ainsi que des ressources nécessaires.

Le contrôle du projet est réalisé par le responsable du département, la Direction du système de management de la qualité et deux responsables de service du département Produits. Ce groupe de personnes forme le comité de pilotage, bien que cette notion n'existe pas en tant que telle dans l'entreprise.

L'équipe projet est composée de deux groupes d'intervenants envers lesquels je n'ai qu'une responsabilité de type fonctionnelle :

- **les intervenants permanents affectés au projet dès le début** : ce groupe est composé d'un chef de projet adjoint, d'un assistant en communication et d'un assistant en qualité,
- **les intervenants ponctuels affectés au projet selon le besoin du sujet à traiter** : ce sont des personnes provenant de diverses équipes de projet de développement logiciel ou d'autres services de l'entreprise.

Le client a imposé un certain nombre de contraintes :

- **visibilité** : sur l'avancement du projet pour éviter « l'effet tunnel »,
- **participation** : faire participer les collaborateurs du département dans la définition des nouvelles pratiques,
- **adaptivité** : avoir un processus adaptable à la diversité de taille et de complexité des projets de développement de produits logiciels.

Ces contraintes m'ont amené à organiser le projet selon une méthode de développement de type itératif et incrémental. La planification et l'évaluation du projet en sont facilitées car les itérations sont de taille plus aisément gérable. Cette méthode permet d'expérimenter et valider très tôt les itérations du processus.

Le projet a ainsi été découpé en une itération de sous projets. Le premier consistant à définir une échelle de niveaux de tests qui sera le support de la politique de test. Il est suivi par une phase de six projets pilotes et de quatre projets de validation qui consiste à définir les processus par une capitalisation de bonnes pratiques. Celles-ci sont déployées par la suite au sein du département.

Les projets pilotes ont pour but d'implémenter les améliorations du nouveau processus. Les projets de validation sont des projets de développement de produit lors desquels le nouveau processus de test est mis en œuvre pour être évalué selon TPI et CMMI.

La mesure de l'avancement et de l'efficacité du projet est basée sur les indicateurs :

- **de coût** : par le suivi de l'utilisation des ressources,
- **des fonctions** : par le suivi des exigences réalisées par rapport au nombre d'exigences prévues,
- **du délai** : par le suivi de la planification.

Le nouveau processus de test, obtenu à l'issue du projet, permet une avancée significative de la qualité des produits, bien que la totalité des attendus du projet n'ait pas été implémentée. Il est une aide au management du département en termes de visibilité sur la partie test des projets ainsi qu'aux chefs de projet de développements logiciels et leur équipe pour être en mesure d'effectuer le suivi et la maîtrise des activités de tests. Les améliorations ont eu, en ce sens, pour effet d'élever le niveau d'indépendance des tests. Les outils, mis en place par le projet d'amélioration (Référentiel, fiches de description...), sont exploitables et exploités par les équipes projets. Il est ainsi possible de considérer, en termes d'efficacité, que l'objectif du projet a été atteint.

Le niveau d'efficience du nouveau processus n'a, à ce stade, pas pu être déterminé. À cela deux raisons, d'une part certaines hypothèses sur lesquelles l'organisation du projet a été basée se sont vues invalidées et d'autre part mon départ du projet pour intervenir, à la demande de la Direction, dans une autre filiale d'IFP Énergies Nouvelles. De fait, le projet a nécessité une réorganisation en novembre 2010 après avoir implémenté plus de 50 % des exigences fonctionnelles pour un coût de 65 % du budget estimé. Cette réorganisation consiste en une mise en application du processus obtenu et implicitement une suspension de l'évolution de ce dernier.

Ce document est organisé en quatre parties.

La première partie (cf. §2) est dédiée à la présentation de la société D2T ainsi qu'à un état des lieux pour mettre en évidence les opportunités d'amélioration. Les objectifs et contraintes du projet d'amélioration y sont détaillés.

La deuxième partie (cf. §3) présente la résolution proposée, reposant principalement sur le modèle TPI. Une présentation du nouveau processus de test y est faite ainsi que les outils mis en place pour le supporter.

La troisième partie (cf. §4) expose l'organisation et la conduite du projet. Cette organisation s'appuie sur les préconisations du PMBOK. Dans cette partie sont également présentés les différents standards utilisés au cours du projet.

Enfin la dernière partie (cf. §5) dresse un bilan projet et personnel.

2. L'amélioration du processus de test et ses attendus

Objectif de cette partie du document

Le test est nécessaire afin de détecter les défauts contenus dans le logiciel lors de sa fabrication. La détection de défauts nécessite la mise en œuvre d'activités et de tâches formant le processus de test. Ce dernier est adapté au contexte dans lequel il est mis en œuvre.

D2T a mené un programme, ensemble de projets, sur l'amélioration de la qualité des produits de la société. Le projet qui m'a été confié se situe au sein du département Produits et consiste à mettre en place des améliorations au processus de test.

Le chapitre 2.2 est un état des lieux de l'existant et présente les méthodes et pratiques de développement et de test en place avant les améliorations apportées par le projet.

Les deux dernières sections présentent les attendus sur le produit et le projet. Les exigences projet sont constituées par le coût, le délai et la qualité de la réalisation du produit.

2.1. Présentation de D2T

Cette section est consacrée à la présentation (1) de la société D2T et ses domaines d'activité ainsi qu'à son actionnaire IFP Énergies Nouvelles ; (2) du département Produits dans lequel se déroule mon projet et des utilisateurs du futur processus de test.

2.1.1. D2T et IFP Énergies Nouvelles

IFP Énergies Nouvelles est un organisme public de recherche dont la mission est de développer des technologies performantes, économiques, propres et durables dans les domaines de l'énergie, du transport et de l'environnement. IFP Énergies Nouvelles a cinq priorités stratégiques dont une consiste à développer des transports économes et à faible impact environnemental. Pour cela, il a besoin de partenaires pour industrialiser ses innovations technologiques dans le domaine des véhicules propres et économes. Ce partenariat s'est traduit par l'acquisition de D2T en 2006 et à faire évoluer l'entreprise pour qu'elle devienne un acteur majeur, au niveau français.

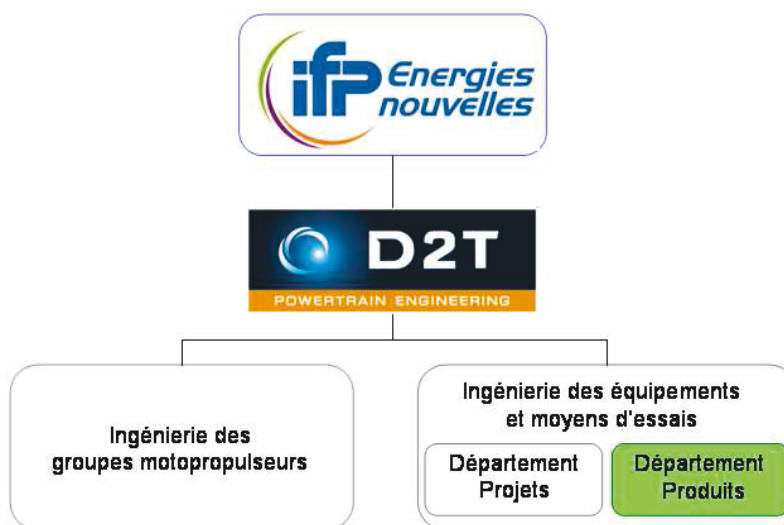


Figure 1 : structure de D2T

D2T est une société française dont le siège social se situe à Trappes dans le Yvelines. Elle est également implantée, à travers des filiales, en Allemagne, aux États-unis, en Chine et en Corée. L'acquisition, par IFP Énergies Nouvelles, de D2T, a accéléré le développement de cette dernière. D2T a doublé son effectif pour atteindre en 2009 le nombre de 330 salariés. Le personnel est réparti sur trois sites : Trappes, Rouen et Lyon. Cette augmentation des effectifs a nécessité une restructuration de l'entreprise ainsi que le changement des méthodes de travail.

D2T a centré ses activités dans deux domaines (Figure 1) : l'Ingénierie des Groupes MotoPropulseurs et l'Ingénierie des Équipements des Moyens d'Essais.

La direction technique **Ingénierie des Groupes MotoPropulseurs** (IGMP) est spécialisée dans le développement et la mise au point des groupes motopropulseurs. Les prestations proposées sont diverses, comme par exemple :

- de l'expertise moteur, en effectuant l'analyse statique du moteur par des mesures spécifiques ou pour la compréhension de défaillances suite à un incident,
- des essais de mise au point, pour valider le potentiel de solutions choisies ou mettre au point les stratégies des calculateurs.

Ces prestations sont réalisées directement dans les centres d'essais (site regroupant plusieurs bancs d'essais) de D2T ou par du personnel D2T détaché chez le client afin d'exploiter les infrastructures de ce dernier.

La direction technique **Ingénierie des Équipements des Moyens d'Essais (IEME)** est spécialisée dans l'ingénierie et la commercialisation de moyens d'essais pour les moteurs et les groupes motopropulseurs. Elle est divisée en deux départements (Projets et Produits – voir Figure 1) proposant chacune des solutions répondant aux attentes des constructeurs (ex. Renault, BMW, PSA, Liebherr,...) et équipementiers (ex. Valeo, Delphi, Magneti Marelli,...).

Le département Projets est en charge de la conception et la réalisation de moyens d'essais chez les clients. Les prestations vont de l'installation d'un nouvel équipement jusqu'à l'étude et réalisation de projets sur mesure de construction de bancs d'essais. Ce département est aussi en charge du support et de la maintenance de moyens existants ainsi que de la formation.

Le département Produits est en charge de la conception et la réalisation d'équipements d'essais (machine de charge, logiciel de supervision de bancs d'essais). C'est au sein de ce département que se déroule mon projet.

2.1.2. Le département Produits

Une présentation détaillée du département Produits est faite pour deux raisons. D'une part, c'est dans ce département que se déroule mon projet, d'autre part, ce département regroupe la majeure partie des utilisateurs du futur processus de test que le projet doit définir. Les produits développés par le département et le personnel qui le compose sont présentés.

2.1.2.1. Les produits

Les constructeurs et équipementiers mettent au point les innovations destinées aux véhicules des années à venir. La complexité des systèmes et le marché les obligent à exploiter au mieux et au maximum leurs bancs d'essais et centres d'essais en les rendant plus performants et plus productifs.

Le département Produits développe des équipements répondant à certains de ces besoins. Ces produits sont regroupés en sept familles, chacune répondant à un besoin spécifique. La Figure 2 présente ces familles sous la forme d'une pyramide pour indiquer que chacune d'elle s'appuie sur une ou plusieurs familles de niveau inférieur et fournit des services aux familles de niveau supérieur.

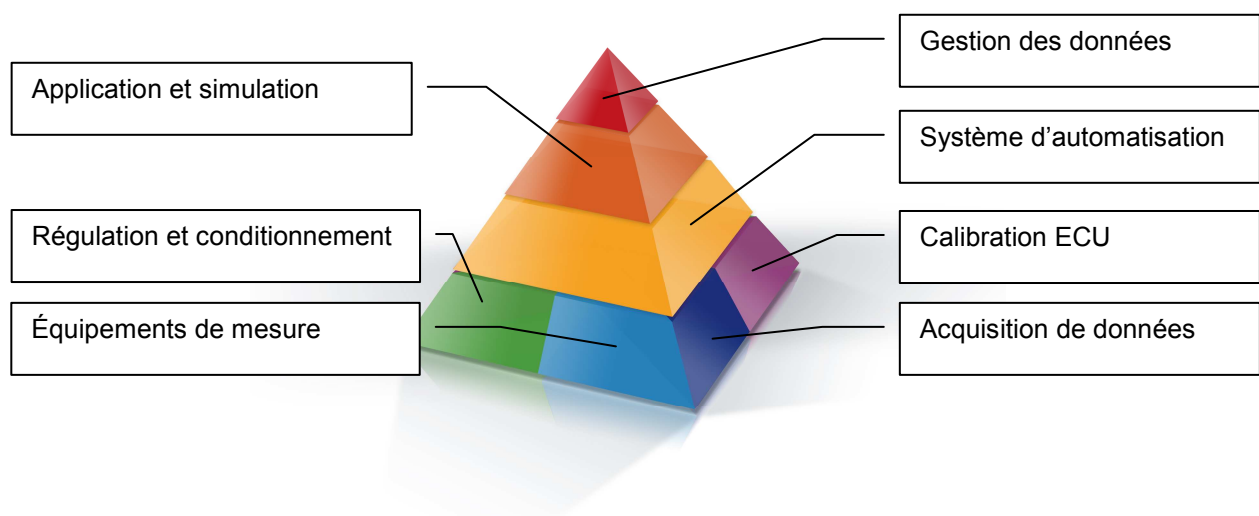


Figure 2 : la pyramide des produits D2T

Les produits logiciels sont développés dans un environnement Microsoft® Windows. La grande majorité, environ 60%, sont des systèmes temps réel. 20 à 25 % sont des Systèmes de Gestion de Base de Données (SGBD) s'appuyant sur les bases de données : Oracle®, MySQL, Microsoft® Access. Le reste correspond à des applications de configuration.

Les produits sont des systèmes de type logiciel, électronique, mécanique ou un mélange de ces types. Toutefois l'effort de développement n'est pas uniformément réparti sur ces trois types (Figure 3). Les développements portent majoritairement sur les logiciels avec 75% de l'effort alloué à ces systèmes. Le dernier quart est dédié à 20% pour les systèmes électroniques et 5% pour le développement de système mécanique.

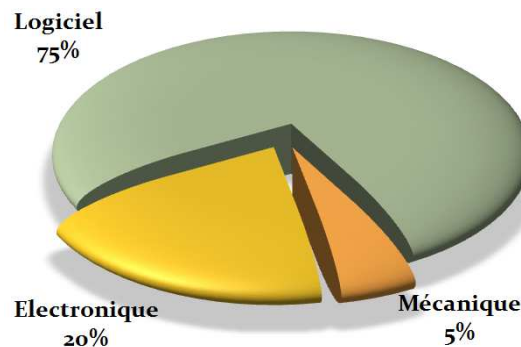


Figure 3 : répartition des produits par type

2.1.2.2. Les utilisateurs du processus de test

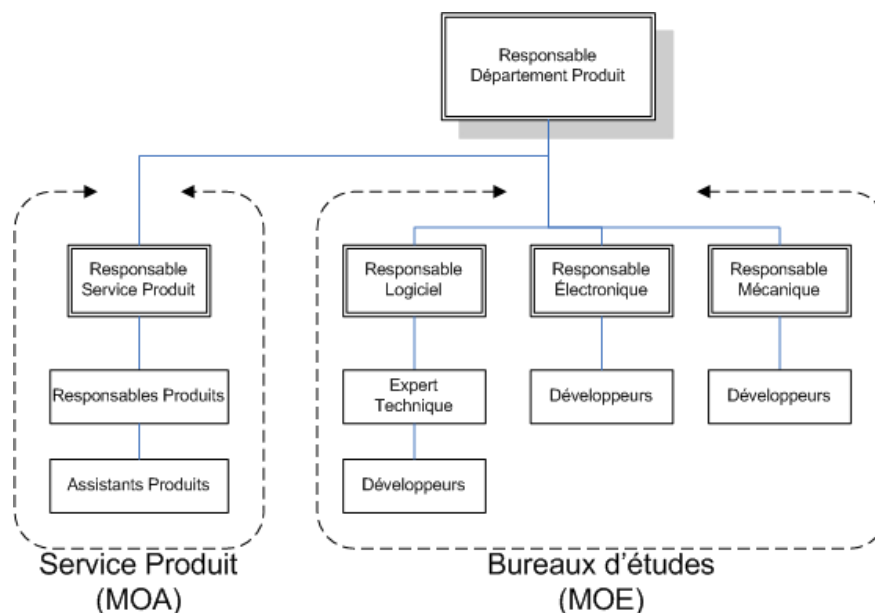


Figure 4 : diagramme organisationnel du département Produits

Le département Produits, composé de 37 personnes, peut être vu comme deux entités (Figure 4). L'une fait office de maîtrise d'ouvrage (MOA) et est appelée Service Produits. La seconde entité peut être considérée comme faisant office de maîtrise d'œuvre (MOE) et est composée de bureaux d'études dédiés aux développements des logiciels, systèmes électroniques et mécaniques.

Le service Produits (branche gauche du diagramme organisationnel - Figure 4) rédige les spécifications et réalise les tests de réception. Les spécifications sont rédigées sur la base de cahiers

des charges clients et / ou suite à des études de benchmarking. Le service est principalement composé de responsables produits (RP), un par famille, en charge :

- d'établir les spécifications métiers,
- de définir le contenu de la documentation utilisateur,
- de valider les nouveaux produits, ou les modifications des produits existants,
- de réaliser le support au client sur leurs produits.

Selon la taille de la famille de produits et de la charge de travail, le responsable peut se voir attribuer un ou plusieurs assistants produits. L'ensemble du service est constitué de 13 personnes et est piloté par un responsable de service.

Les bureaux d'études sont composés pour chaque type de système d'un responsable et de plusieurs développeurs. Chaque développeur est en charge de plusieurs produits provenant de diverses familles. Ils sont en charge :

- de la réalisation des étapes de conception,
- du développement et l'intégration des nouvelles fonctionnalités,
- de la réalisation des tests conformément à la conception,
- de la rédaction de la documentation technique interne.

Les bureaux d'études sont composés de 2 à 12 personnes. L'équipe du bureau d'études logiciel est la plus importante et peut selon les besoins être renforcée par des prestations externes.

Toutes les équipes sont géographiquement très proches. Cette proximité a l'avantage de permettre une grande réactivité pour la résolution des problèmes techniques. Mais elle a comme principal inconvénient de réduire la production de documents techniques. Une grande partie de la transmission des informations se fait oralement. Des documents sont, bien sûr, produits au cours du développement mais ils ne sont généralement pas maintenus à jour et deviennent rapidement obsolètes.

Les clients du département sont majoritairement le département projet (Figure 1 – page 12) et les filiales de D2T. Mais il fournit aussi directement des produits à des intégrateurs réalisant des prestations de maintenance, de remise à niveau ou réalisation de nouveaux bancs chez les constructeurs ou équipementiers.

2.1.3. Améliorer la qualité des produits, un axe stratégique

Les méthodes et pratiques de gestion des développements mises en œuvre au sein du département produits entraînent divers problèmes tels que des dépassements du délai et/ou du coût prévu ou un manque de visibilité sur le contenu du logiciel par rapport aux exigences.

Consciente que la qualité des produits est indispensable au maintien de l'entreprise à un niveau concurrentiel sur le marché, la direction a déclaré, en 2009, l'amélioration de la qualité des produits comme faisant partie des axes stratégiques de l'entreprise.

La solution proposée par le responsable du département Produits consiste à travailler sur deux chantiers : la **gestion des projets** et la **gestion des tests** (Figure 5). Un responsable a été désigné pour mener à bien chacun d'eux. Une des premières actions du chantier « gestion des projets » a été de mettre en place un mode de fonctionnement par projet et de définir la mission de chef de projet.

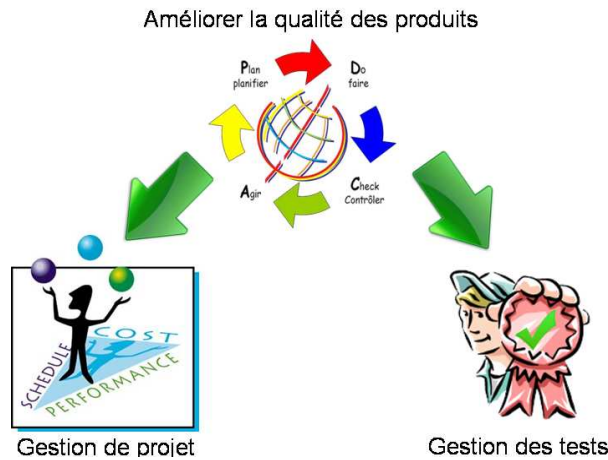


Figure 5 : les chantiers d'amélioration

La réalisation du chantier « amélioration de la gestion des tests » m'a été confiée du fait de mon expérience acquise dans le domaine du test logiciel. Ces connaissances ont été acquises de par ma pratique professionnelle. J'occupe depuis plusieurs années un poste d'assistant produits et j'ai réalisé les tests de validation de la majorité des logiciels du département. Ma formation au CNAM a également contribué au fait que le projet m'ait été confié. Mes connaissances acquises, dans le domaine du génie logiciel, des normes et surtout dans les pratiques du test logiciel, au cours de mon cursus au CNAM ont contribué à cette prise de responsabilité en tant que chef de projet AMO.

Les attendus du projet ont été élaborés dans une phase d'initialisation à travers plusieurs réunions de travail entre le responsable du département Produits (MOA) et moi (AMO). Il s'agissait de préciser l'objectif et le périmètre du projet d'amélioration de la gestion des tests.

2.2. Constat sur le processus de gestion des tests

La majeure partie des produits développés, au sein du département, est à 75 % de type logiciel. La qualité perçue d'un produit logiciel par un utilisateur tient à la réponse à deux questions simples : Le logiciel fait-il ce qu'on attend de lui ? Est-il fiable ?

En général à la première question l'utilisateur ne peut avoir de réponse qu'en comparant le comportement du logiciel en opération et la spécification qu'il ne possède pas toujours. Pour la réponse à la seconde question c'est plus simple ça marche ou ça ne marche pas lorsque le nombre de défaillances lui semble trop important.

Le ressenti du département Produit était que les produits logiciels étaient corrects sans qu'une étude spécifique ait été menée mais l'amélioration permanente de la qualité des produits est devenue une nécessité pour les entreprises. La qualité si elle augmente un peu le coût de fabrication réduit le coût sur le cycle de vie du produit en réduisant le coût de la maintenance.

Pour établir un constat objectif sur l'état des lieux il fallait retenir un outil d'évaluation qui permette de se mesurer par rapport à une échelle « objective » indépendante. Je reviendrai plus en détail sur ce point dans la présentation de mon travail pour l'heure retenons que je me suis inspiré de l'International Software Testing Qualification Board (ISTQB), organisme qui vise à professionnaliser le métier de testeur et de TPI, Test Process Improvement, de SOGETI qui évalue les processus de test à l'aide de niveaux de maturité.

2.2.1. Présentation du processus de développement incluant les tests

Voyons le socle méthodologique existant afin de permettre au lecteur d'avoir une vision des méthodes et pratiques de développement et de test avant les modifications apportées par le projet.

La Figure 6 présente le cycle de développement des produits. Celui-ci est appliqué à tous les développements. Toutefois, la présentation des phases de ce cycle sera focalisée sur le développement des logiciels. Ce choix est dû au fait que :

- la majeure partie des développements est de type logiciel,
- le projet d'amélioration du processus de test est orienté test logiciel.

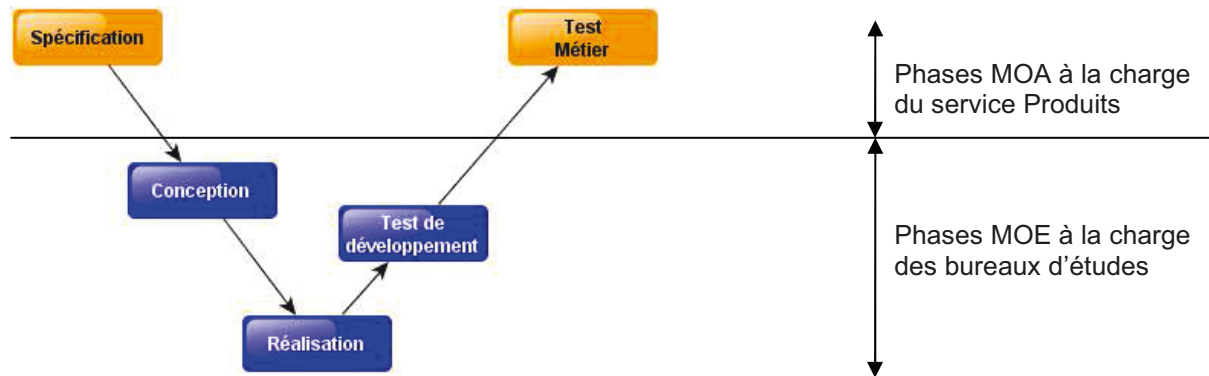


Figure 6 : Cycle de développement des produits

Les développements réalisés sont de type maintenance ou évolution. Les développeurs, liés à des produits, sont amenés à faire indifféremment des développements :

- **de maintenance** : changements ou ajouts au code dans le but de corriger des anomalies ou réaliser de petites améliorations des fonctionnalités du logiciel,
- **d'évolution** : changements ou ajouts de nouvelles fonctionnalités, et/ou amélioration significative des performances du logiciel. Dans ce cas l'évolution est considérée mineure. L'évolution est considérée comme majeure, lorsque des perfectionnements techniques additionnels et/ou des innovations techniques majeures sont apportées au logiciel.

Le cycle de développement est un cycle de type en V relativement simple constitué de cinq phases. Les phases hautes du V, spécification et test métier, sont à la charge du service Produits. Quant aux trois phases formant la pointe du V, « conception », « réalisation » et « test de développement », elles sont du ressort du bureau d'études.

2.2.1.1. La spécification

La rédaction des spécifications est à la charge des responsables produits. Toutes les spécifications sont rédigées selon le type informel (utilisation du langage naturel accompagné ou non de représentation graphique d'interface utilisateur). La principale raison de ce choix est qu'il permet de faire rédiger des spécifications par n'importe qui (assistants produits, développeurs, personnes provenant des départements projet ou IGMP, voire les clients externes). Il a également comme avantage d'être compréhensible par tous les lecteurs. L'inconvénient, en contre partie, est qu'il est sujet à interprétations et peut comporter des ambiguïtés à l'origine des erreurs.

De même, la structure du document est complètement libre. Chaque rédacteur de spécification a toute liberté d'organiser les informations (les exigences) selon ses compétences dans le domaine. Les exigences sont, de fait, difficiles à identifier car souvent noyées dans le flot d'informations bruyantes.

Enfin, il arrive que, selon l'urgence ou par manque de ressource et de temps, des développements commencent ou sont réalisés sans spécifications écrites. Les exigences sont alors transmises oralement.

2.2.1.2. La conception

Comme pour les spécifications, la documentation issue de la conception est soumise à peu de formalisme. Lorsqu'elle existe, elle est rédigée en langage naturel voire constituée de copies de portion de code source. La formalisation de la documentation de conception a été tentée grâce de la modélisation UML sans pour autant que cette pratique se généralise.

La rédaction de la documentation de conception est à la charge du développeur. Elle est généralement réalisée au cours, voire à l'issue, de la phase de réalisation, d'où la présence de code source.

2.2.1.3. La réalisation

La phase de réalisation correspond à l'écriture du code source des logiciels. Cette tâche incombe bien sûr aux développeurs. Cette phase inclut également l'activité de débogage consistant à trouver, analyser et éliminer les causes de défaillance dans les logiciels.

Cette phase est la seule phase outillée du cycle. Les développements sont réalisés dans l'Environnement de Développement Intégré (EDI ou IDE en anglais) Visual Studio de Microsoft®. Les langages utilisés sont principalement le C++, C#, SQL

L'intégration consiste à s'assurer que tous les éléments développés au cours de la phase de réalisation arrivent à communiquer entre eux et avec ceux constituant le logiciel. Cette phase intervient généralement à la suite de l'écriture du code source. Elle n'est pas identifiée sur la branche montante du V car elle est considérée faisant partie du travail de réalisation. Dans la majorité des cas, cette situation est satisfaisante du fait qu'un développement de logiciel est réalisé par un seul développeur.

2.2.1.4. Taille, planification et mise à disposition

La taille d'un développement est exprimée en hommes/jours qui ne préjuge en rien de la durée du développement. Cette dernière dépendant du nombre de personnes impliquées dans le développement et de l'organisation de celui-ci. La taille est définie sur la base d'estimations d'experts selon un contenu fonctionnel envisagé. Ce contenu fonctionnel est déterminé pour répondre à des attentes clients ainsi que pour suivre les tendances du marché.

Le SMQ de D2T impose une planification des développements basée sur des techniques type Gantt ou PERT. Une date de fin est proposée par les responsables produits et des ressources sont allouées pour atteindre cet objectif.

Les produits sont mis à disposition selon besoin, le plus généralement pour respecter des dates de livraisons contractuelles quitte à réduire le contenu fonctionnel aux seules attentes du client. Les fonctions non réalisées étant reportées dans la version suivante.

2.2.2. Constat qualitatif du processus existant

Le constat qui est fait dans cette section vise à donner au lecteur une vision qualitative du processus de test avant que le projet ne le fasse évoluer.

2.2.2.1. Le test de développement

Cette phase est réalisée par les développeurs et correspond à des tests effectués sans préparation formelle des tests, pas d'utilisation de technique de conception de tests reconnue. Il n'y a pas d'attente spécifique de résultats et le hasard guide les activités de test. L'ISTQB et le Comité Français des Tests Logiciels (CFTL) qualifient ce type de test de « test ad hoc » [13].

Ces tests sont difficilement traçables car peu, voire aucune, documentation n'est produite durant cette phase.

2.2.2.2. Le test métier

Les tests réalisés lors de cette phase sont aussi appelés tests fonctionnels métier ou tests de validation. Ils sont à la charge du responsable produits qui les délègue à l'assistant produits lorsqu'il existe. Ces tests visent à s'assurer de l'adéquation du logiciel livré avec les spécifications. Dans les faits, ces tests sont fréquemment des tests ad hoc.

Une documentation est produite lors de cette phase qui vise surtout à fournir une preuve que le test a été effectué. Elle contient généralement un ensemble de constatations faites lors de l'utilisation du logiciel. Cette documentation ne permet ni reproductibilité de tests, ni traçabilité vers les exigences de la spécification. Il est donc difficile de déterminer, à l'issue de cette phase si le logiciel livré correspond aux besoins exprimés. Le logiciel est généralement accepté par le responsable produits sur son ressenti à la suite de plusieurs reprises du code source.

2.2.2.3. Indépendance des tests

ISO 12207 suggère de réaliser les tests de manière indépendante. La norme IEEE 1012 définit le niveau d'indépendance de la vérification et validation [24] selon trois paramètres :

- **financier** : le contrôle du budget est dévolu à une organisation indépendante du développement,
- **technique** : utilisation de personnel non impliqué dans le développement du logiciel,
- **managérial** : la responsabilité de l'effort de test est séparée du développement et de la gestion du projet.

Élevé le niveau d'indépendance, revient à élever un ou plusieurs de ces paramètres.

2.2.2.4. L'assurance qualité

D2T est certifié ISO 9001. Parmi les procédures prévues par le Système de Management de la Qualité (SMQ), une est applicable au développement de produits. Celle-ci reprend le cycle en V présenté en section 2.2.1 et lui adjoint un ensemble d'activités visant à assurer d'obtenir la qualité attendue. Une de ces activités est la revue de développement. Lors des revues, un ensemble de personnes doit évaluer la gestion du développement (planning, avancement,...) et les éléments techniques (complétude, conformité, ...) pour décider du passage à la phase suivante. À ces revues, ayant un caractère formel, sont conviées les personnes en charge des phases se terminant et celle leur succédant ainsi que des parties prenantes. Par exemple, à la revue de spécification, conditionnant le lancement de la phase de conception, sont conviés :

- le responsable produits (en charge de la rédaction de la spécification),
- le responsable du bureau d'études logiciel (en charge de la conception),
- un représentant du marketing (partie prenante),
- le responsable du département (partie prenante),
- un représentant du commercial (partie prenante),
- un correspondant qualité (en charge de l'assurance qualité du développement).

Les anomalies sont gérées différemment selon qu'elles sont détectées lorsque le logiciel est en cours de développement ou en production chez le client, aussi appelée vie série.

En vie série, lorsqu'un problème est identifié sur un logiciel par un client, il appelle ou envoie un e-mail au service support aussi appelé Hotline. Le client reçoit en retour un numéro de dossier comme confirmation de la prise en compte de sa demande. Le dossier est ensuite traité par un technicien. Lors du traitement, s'il était avéré que le problème client est dû à une anomalie d'un logiciel, le dossier est transmis au responsable produits en charge du logiciel. Le RP évalue la gravité de l'anomalie et décide selon cette gravité s'il est nécessaire de réaliser une version de maintenance ou si cela peut attendre la sortie d'une version d'évolution. Il transmet sa décision au service support qui en informe le client.

En cours de développement, les incidents détectés sont répertoriés dans l'outil open source TRAC. L'utilisation d'un outil pour gérer le processus de traitement des incidents permet une bonne traçabilité. Cette gestion des incidents, par cet outil, est toute récente (depuis moins d'un an).

2.2.3. L'état de l'Art dans le domaine des tests n'est pas pris en compte

Les utilisateurs du processus de test sont le personnel du département Produits. Selon le Système de Management de la Qualité (SMQ) de D2T la gestion du processus de développement et donc implicitement du processus de test incombe au responsable du département. Le responsable du département Produits est de fait à considérer comme étant la MOA.

Selon l'International Software Testing Qualification Board (ISTQB), organisation internationale reconnue dans le domaine du logiciel et visant à promouvoir la professionnalisation du métier de testeur logiciel, la qualité d'un système est hautement influencée par la qualité des processus pour développer ce système. Or selon la norme ISO 12207, norme définie par l'Organisation Internationale de Normalisation qui recense et décrit les processus couramment mis en œuvre dans les projets informatiques, les processus qui régissent le test sont les processus de vérification et de validation.

La norme ISO 12207 est un cadre d'accueil pour organiser les bonnes pratiques dans le domaine du cycle de vie du logiciel. Elle indique les processus et leurs activités, le quoi faire pour effectuer de la vérification et de la validation. Elle donne les directives aux normalisateurs pour affiner le contenu des processus.

Le processus de **vérification** est le processus qui permet de déterminer si les produits logiciels d'une activité satisfont les exigences ou conditions que les activités précédentes leur imposent. Le processus de **validation** est le processus qui permet de déterminer si les exigences et le système ou logiciel finaux satisfont un emploi spécifique visé. En d'autres termes, la vérification permet de s'assurer qu'on a **bien** fait un logiciel et la validation qu'on a fait le **bon** produit [3].

Les tests du logiciel sont une des activités qui entrent dans les deux processus.

2.2.4. Dans l'échelle TPI le processus de test existant n'atteint pas le premier échelon

Il s'avère que le niveau d'indépendance des tests est faible dans le département Produits. Au niveau financier le budget pour les tests de développement n'est généralement pas identifié (confondu avec celui de développement). Pour le test métier un budget est défini mais son utilisation n'est pas clairement identifiée. Au niveau technique, seuls les projets pour lesquels un assistant produits est désigné peuvent être considérés comme ayant une indépendance technique, ce qui n'est pas la normalité. Enfin, au niveau managérial aucune responsabilité concernant l'effort de test n'est du ressort de l'assistant produits.

Pour évaluer la performance du processus de test il fallait disposer d'un référentiel d'audit. Au moment où ce travail m'a été confié je n'avais pas connaissance de l'existence de normes spécifiques sur lesdits processus. Plutôt que d'inventer un processus ad hoc qui n'aurait eu aucune référence en dehors de l'entreprise j'ai retenu le Test Process Improvement (TPI). Le site Web de SOGETI [28] met à disposition un outil pour procéder à une auto-évaluation quantitative du processus de test. Ce produit permet d'évaluer le processus selon le modèle d'amélioration TPI qui attribue des niveaux de maturité aux pratiques en place. En gros les classent dans une échelle de valeur. Le modèle TPI est présenté dans le §4.1.1.2 page 57.

L'évaluation du processus de test du département Produits, à l'aide de cet outil, a été réalisée par deux personnes : un chef de projet développement et moi-même. Chacun a évalué les pratiques de test d'un à deux développements de son choix. Les développements évalués devaient être terminés et réalisés sur la période de 2008-2009. À l'issue, les résultats obtenus ont été comparés. En cas de divergence, entre les deux évaluations, les points de contrôle divergents ont été discutés et chaque évaluateur a argumenté sa décision jusqu'à obtention d'un consensus sur le point de contrôle.

Cette évaluation a montré que le processus du département Produits de D2T n'a pas encore atteint l'échelon 1 de TPI.

Le Tableau 1 récapitule les résultats obtenus à l'issue de l'évaluation. Les deux colonnes de gauche indiquent les secteurs-clefs (n° et nom) concernés. Les deux colonnes de droites indiquent les nombres de points de contrôle « valide » donc pour lesquels des preuves existent démontrant que des pratiques, relatives au secteur-clef, sont réalisées en regard du nombre « total » de points de contrôle pour ce secteur-clef.

		Points de contrôle	
Secteur-clef		valide	total
1	Stratégie de test	0	4
2	Modèle de cycle de vie	1	14
5	Techniques de spécification de test	0	2
11	Engagement et motivation	3	6
15	Reporting	0	1
16	Gestion des anomalies	7	8
18	Gestion du processus de test	0	1
	TOTAL	11	36

Tableau 1: résultats de l'évaluation avant amélioration

2.3. Les attendus à l'issue du projet (objectifs)

Au vu des résultats obtenus à l'issue d'une évaluation (cf. Tableau 1), il s'avère que les objectifs du projet sont de deux types :

- **maintenir** : 11 exigences sont satisfaites, mais cela ne signifie pas qu'elles sont systématiquement mises en œuvre au cours de chaque développement. Le projet devra déployer et maintenir, à l'ensemble des projets de développement, les pratiques actuellement existantes,
- **définir** : 25 exigences ne sont pas satisfaites. Pour chacune d'entre elles, le projet devra proposer et déployer des pratiques.

Les attendus en fin de projet seront donc :

- Mettre en place un référentiel d'amélioration qui soit accepté par tous et permette comme le suggère ISO 9000 de procéder à des améliorations continues. Ce référentiel, au sens qui fait référence, sera TPI ;
- Mettre en place un cadre de définition de la politique de test de sorte que chaque projet puisse la transposer à ses exigences spécifiques ;
- Mettre en place un processus de gestion des exigences fonctionnelles et non fonctionnelles pour définir le périmètre des tests ;
- Vérifier l'efficacité des documents produits en les faisant utiliser par les équipes de test dans des projets réels.

En fait ce qui sera attendu de la part du projet c'est un référentiel documentaire concernant le processus de test utilisable par tous les chefs de projet, une sorte de manuel qualité pour les tests. Évidemment il sera nécessaire de valider ce « manuel » au contact du terrain en conseil auprès de chefs de projet.

2.3.1. Introduire les secteurs-clef de l'échelon 1 de TPI

Les exigences en termes d'améliorations du processus de test sont regroupées dans le Tableau 2. Elles ont été définies à partir des points de contrôle de l'outil d'évaluation et des suggestions d'améliorations proposées par le modèle TPI, sachant que les définitions suivantes s'appliquent :

- **Maintenir** : signifie que le livrable, issu de la pratique relative à TPI, est conservé à jour de sa création jusqu'au bout du projet de développement.
- **Test en aval** : tests réalisés lors de la branche montante du cycle de vie de développement en V.
- **Cas de test** : ensemble des jeux de données (valeurs en entrée et sortie), des scénarii et des résultats attendus, utilisé pour la réalisation d'un test.
- **Test de confirmation** : test qui exécute des cas de test qui ont été en échec la dernière fois qu'ils furent exécutés, de façon à vérifier le succès des actions de corrections [13].
- **Technique informelle** : signifie que la personne rédigeant le cas de test a une grande liberté pour le définir [8].

n°SC	Objectif principal	Détail de l'objectif
1	Établir et maintenir une stratégie de test pour les tests en aval	Identifier les risques en fonction des exigences
		Adapter les tests en fonction du risque
		Choisir la technique en fonction du risque
		Identifier le type de ré-exécution en cas de test de confirmation
2	Établir et maintenir un modèle de cycle de vie des tests : Planification, Description, Exécution	Rédiger un plan de test
		Établir une description des tests
		Exécution des tests
5	Établir des techniques informelles de description de test	Documenter les techniques de construction de cas de test
11	Établir l'engagement et la motivation : Affectation de budget et de temps	Faire considérer les tests comme une chose nécessaire et importante.
		Affecter aux tests des ressources disponibles et un budget
		Piloter les tests par le budget et les ressources
		Faire monter en compétences les personnes en charge des tests
15	Établir du reporting des anomalies	Faire des comptes-rendus périodiques
16	Gérer les anomalies (en internes)	Gérer les étapes de recherche d'anomalies
		Enregistrer les anomalies
18	Gérer le processus de test : Planification et exécution	Dresser un plan de test

Tableau 2 : objectifs d'amélioration pour atteindre l'échelon 1 de TPI

Les concepts de plan et stratégie de test étant nouveaux et mal appréhendés, par les personnes en charge de les définir, il est nécessaire, afin de leur apporter une aide à la détermination de l'objectif et du périmètre, de :

- créer une politique de test permettant de définir l'objectif de test (§ 2.3.2),
- gérer un référentiel des exigences servant de périmètre du test (§ 2.3.3).

2.3.2. Définir une politique de test

L'objectif de test d'un logiciel est souvent exprimé sous la forme : « Il faut tester le logiciel », « Il faut que cela fonctionne ». Tester n'est pas un objectif c'est une activité, que le logiciel fonctionne est un objectif mais est trop ambigu pour être atteignable. Un objectif de test vise généralement à contrôler une caractéristique (fonctionnelle, performance, robustesse...) du logiciel ou d'une de ces parties [3].

Définir un objectif de test en termes de caractéristiques est trop ambitieux pour démarrer le projet d'amélioration du processus de test.

Il est donc nécessaire de proposer un moyen simple de définir un objectif de test pour chaque projet de développement. Ce moyen est une politique de test qui doit permettre à chaque chef de projet de définir un objectif et une stratégie de test de manière commune.

2.3.3. Gérer les exigences pour définir le périmètre du test

La notion d'exigence provient d'une traduction du terme « requirement » issu du vocabulaire anglo-saxon. Ce terme est défini par la norme IEEE 610 - Standard Glossary of Software Engineering Terminology comme « une condition ou la capacité qui doit être remplie ou possédée par un système ou un de ses composants pour satisfaire à un contrat, une norme, une spécification ou tout document imposé de façon formelle ». Une exigence est la vue de ce que le « client » demande et la spécification est la traduction « plus » technique de l'exigence. Par exemple, le client demande « Authentification du demandeur » et la spécification précise la façon dont l'authentification sera effectuée comme saisie d'un identifiant et d'un mot de passe.

Dans ce document, les termes « exigences » et « spécifications » seront considérés comme équivalents. L'intérêt d'utiliser « exigence » est de mettre l'accent sur l'aspect traçabilité envers les tests.

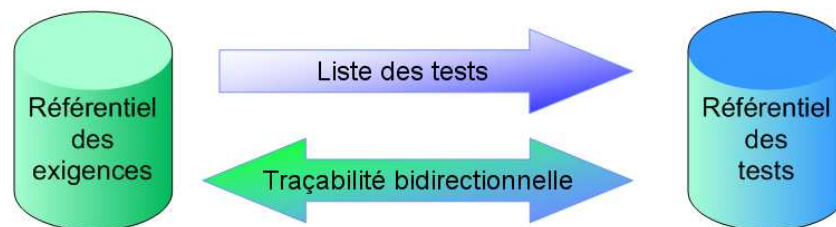


Figure 7 : traçabilité exigence/test

Le moyen répertoriant l'ensemble des exigences constitue ce que l'on appelle le référentiel des exigences [2] et définit le périmètre d'un développement. L'ensemble des tests permettant de contrôler la satisfaction des exigences par le développement est appelé le référentiel de test. Le rapport entre les deux référentiels correspond à la couverture des exigences par les tests. Pour pouvoir déterminer cette couverture, il est nécessaire de mettre en place un moyen de faire un lien entre les deux référentiels. Ce lien est appelé la traçabilité bidirectionnelle (Figure 7). Elle permet de savoir quel test permet de contrôler une exigence et inversement.

Le référentiel des exigences est un élément important au bon fonctionnement du processus de test. Il est généralement considéré comme préexistant. Cela est vrai si l'on considère les experts métiers comme faisant office de référentiel des exigences. La situation actuelle fait que ce dernier est constitué de documents et matériels disparates tels que des documents Word, des échanges d'e-mail, d'anciennes versions du produit ou directement par l'expert métier. Cette situation n'est pas favorable à la traçabilité bidirectionnelle exigences/tests.

Afin de pouvoir démarrer une amélioration du processus de test, il faut indispensablement que les exigences soient structurées dans un même référentiel et que celui-ci soit maintenu à jour tout au long du projet [2]. Il a donc été convenu avec le responsable du département (MOA) d'inclure une partie de formalisation du référentiel.

Cette action de formalisation et de gestion du référentiel est appelée, dans la terminologie Capability Maturity Model Integration (CMMI), la gestion des exigences. Elle y est référencée comme le domaine de processus REQM pour REquirements Management. Ce sont le modèle CMMI et la norme IEEE 830- Recommended Practice for Software Requirements Specifications qui sont utilisés pour

évaluer la maturité du processus et déterminer les améliorations à apporter. Le modèle CMMI et la norme IEEE 830 sont respectivement présentés en §4.1.1.4 page 62 et §4.1.1.5 page 64.

Ce sont les objectifs et pratiques du domaine de processus REQM qui sont utilisés comme objectif (Tableau 3) à mettre en œuvre au sein de mon projet. Cet objectif vise à atteindre un niveau d'aptitude 2 du domaine de processus REQM de CMMI.

n°	Objectif principal	Détail de l'objectif
1	Gérer les exigences	Obtenir une compréhension des exigences
		Obtenir l'engagement sur les exigences
		Gérer les modifications aux exigences
		Maintenir la traçabilité bidirectionnelle des exigences
		Identifier les incohérences entre les produits du projet et les exigences
2	Institutionnaliser le processus	Mener des actions pour maintenir le processus dans le temps

Tableau 3 : objectifs d'amélioration pour atteindre le niveau d'aptitude 2 pour REQM

2.3.4. Introduire la pratique du bilan des tests pour capitaliser et améliorer en continu

Selon le paradigme ETCS [4], un processus est constitué de quatre caractéristiques : les Entrées, la Tâche, le Contrôle et les Sorties (ETCS). Les entrées du processus sont les exigences. Le contrôle et les sorties correspondent aux critères qui doivent être satisfaits pour que la tâche puisse être déclarée terminée. Ces critères peuvent, par exemple, être « tous les tests prévus ont été réalisés », mais ce n'est pas suffisant et pour aider à améliorer les estimations (et aussi à mieux organiser les tests lors des projets futurs). Il était nécessaire de capitaliser en conservant un historique de la phase de test du développement.

Dans cette optique, il est nécessaire de mettre en place un moyen permettant la réalisation d'un bilan à l'issue de chaque phase de test. L'exigence au niveau du projet sera de créer ce moyen pour réaliser le bilan des tests et ainsi capitaliser des données relatives à la phase de tests de manière commune à l'ensemble des projets de développements.

Cette exigence vient s'ajouter aux objectifs définis pour atteindre l'échelon 1 de TPI décrit dans le Tableau 2 (page 22) et dont un extrait est donné dans le Tableau 4.

n°SC	Objectif principal	Détail de l'objectif
2	Établir et maintenir un modèle de cycle de vie des tests :	Réaliser le bilan des tests

Tableau 4 : exigence relative au bilan des tests

2.4. Les exigences projet

Les exigences projet sont de trois types : Coût, Délai et Qualité et forment avec les exigences produit (les Fonctions) décrites dans le chapitre 2.3 le CQFD du projet. C'est sur le CQFD que sera jugé si le projet a atteint son objectif.

2.4.1. Le coût

Aucune contrainte budgétaire n'a été imposée au projet. Cela ne signifie pas pour autant que le budget est illimité. Le responsable du département Produits m'a demandé d'estimer le coût de réalisation préalablement au lancement du projet afin que ce coût soit validé et officiellement accepté par la direction. Ce coût de réalisation a été estimé 222000€ et correspond principalement à du temps

homme. Il n'est pas prévu d'achat de logiciel pour améliorer le processus de test. Le détail et la méthode qui a permis d'obtenir cette estimation sont détaillés dans le chapitre 4 Conduite du projet.

2.4.2. Le délai

Aucune contrainte de délai n'est imposée. Le responsable du département et moi-même avons toutefois estimé que le projet était réalisable dans un délai d'un an. Le détail de cette estimation est fourni dans le chapitre 4 Conduite du projet. Une fin de projet au dernier trimestre 2010 est donc à prendre en compte comme contrainte de délai.

2.4.3. La qualité

Les exigences de type qualité correspondent à des exigences, imposées par le responsable du département Produits (le client), contraignant la conduite du projet.

Visibilité : Il désire avoir une bonne visibilité sur l'avancement du projet et ainsi éviter « l'effet tunnel ». Pendant toute la durée du projet, il désire avoir régulièrement un point d'avancement sur le budget consommé, les exigences produites réalisées et restant à faire.

Facilité d'utilisation : Ce projet va avoir un impact sur les pratiques de travail des personnes du département. Il faut donc accompagner le changement et les faire participer dans la définition des nouvelles pratiques. Le nouveau processus doit être basé sur la capitalisation. Le responsable du département ne veut pas imposer un processus et des méthodes toutes faites mais quelles soient définies sous la forme d'un mode participatif par le personnel du département. Mon rôle consiste à les piloter et les guider dans la définition du processus de test.

Adaptivité : Les projets de développement de produits logiciels sont de taille et complexité différentes. Le nouveau processus devra pouvoir être adapté selon la circonstance.

Conformité : D2T est certifié ISO 9001. Le nouveau processus doit respecter les contraintes de cette norme.

Évolutivité : Le nouveau processus doit pouvoir être exploitable très tôt. Le responsable du département ne veut pas attendre la fin du projet pour que le nouveau processus soit mis en exploitation mais qu'au fur et à mesure des avancés du projet des versions intermédiaires du processus soit appliquées, du moins sur certaines projets de développement de logiciels.

Les contraintes de coût, délai et qualité forment avec les exigences produites l'objectif CQFD du projet. Néanmoins il est généralement difficile de mener à bien un projet en respectant ces quatre types de contraintes. Il est donc nécessaire de définir une priorité de respect de chacune des composantes de l'objectif afin de savoir sur lesquels il est possible d'agir lors de prise de décisions en cours de projet. Il m'a été demandé de respecter prioritairement le coût du projet ensuite les fonctions attendues et la qualité occupe le niveau de priorité 2 et le délai le niveau 3.

Conclusion de cette partie du document

Dans ce chapitre, nous avons vu les deux secteurs d'activité de la société D2T. La branche IGMP spécialisée dans le développement et la mise au point de groupes motopropulseurs et la branche IEME dont l'un des départements est dédié au développement de systèmes pour les bancs et centres d'essais. C'est le département dans lequel se déroule mon projet.

Le projet, dont j'ai la charge en tant que chef de projet AMO, vise à améliorer le processus de test. Ces améliorations sont définies grâce au modèle Test Process Improvement.

Les quatre principales exigences en termes d'amélioration du processus de test sont :

- Introduire les secteurs-clef de l'échelon 1 de TPI
- Définir une politique de test
- Gérer les exigences pour définir le périmètre du test
- Introduire la pratique du bilan des tests pour capitaliser et améliorer en continu

Le projet se déroule sur une période d'un an pour un coût estimé de 222 000€.

3. Le processus de test intégré

Objectif de cette partie du document

Cette partie est dédiée à la présentation du processus de test intégré (sous-entendu au processus de développement) tenant compte des exigences soumises au projet. Tout va tourner autour de la mise en place de documents d'exigences sur le déroulement du projet et sur l'implication des acteurs à chaque étape.

Le chapitre débute par la présentation du nouveau processus de test intégré au processus de développement, détaille les principaux outils mis en place pour soutenir les activités du nouveau processus de test et les actions à mener pour l'utilisation desdits outils.

Ces outils se présentent, pour la plupart, sous la forme de modèles de document Word ou Excel. Les outils du processus sont :

- **la politique de test** : vise à définir un cadre de test commun à tous les projets de développement de logiciel au département Produits de D2T. Elle définit cinq niveaux de test correspondant chacun à un objectif de test. Un chef de projet de développement est ainsi en mesure de définir une stratégie de test selon le niveau à atteindre pour son projet,
- **le référentiel des exigences et des tests** : est l'outil central du processus de test, il permet d'effectuer la gestion des exigences d'un projet de développement pour ensuite y définir le plan de test, le suivi, la maîtrise et enfin réaliser le bilan des tests,
- **la fiche de description de tests** : est un modèle qui sert à formaliser les cas de test,
- **la fiche de résultats de test** : est un modèle qui sert à conserver des traces de l'exécution.

Afin de donner un aperçu de la couverture de ces exigences par les solutions mises en place, un tableau récapitulatif est présenté dans le chapitre 3.5 contenant un rappel de l'ensemble des exigences et la section du chapitre répondant à l'exigence.

3.1. Présentation du processus de test intégré

L'ensemble des activités du processus de test intégré est conforme à la politique de test du département Produits. Le Tableau 5 récapitule les activités et les résultats attendus spécifiquement aux tests.

Phase du projet	Activité de test	Résultat de l'activité - livrable
Lancement	Détermination de l'objectif de test en accord avec la politique de test Le chef de projet définit le niveau de test visé qui s'inscrive dans la politique de test de l'entreprise et la note dans le référentiel. Cette activité peut être réalisée en début de planification des tests.	Initialisation du référentiel des exigences et des tests
Analyse	Identification, caractérisation et mise sous contrôle des exigences et des risques produit Le chef de projet dresse la liste des exigences dans le référentiel. Le chef de projet identifie et caractérise dans le référentiel les risques liés au développement du produit.	Mise à jour du référentiel avec les exigences du produit à développer et les risques associés
Conception	Planification des tests Le testeur définit le plan de test et doit être validé par le chef de projet.	Plan de test au niveau du référentiel
Réalisation	Conception des tests formels Lorsque le plan de test prévoit la réalisation de tests formels, le testeur, le chef de projet ou le développeur rédige les fiches de description des tests.	Fiche de description des tests
Tests	Exécution des tests selon le plan Les tests sont réalisés conformément au plan. Un dossier de validation est rempli par le testeur. Le testeur conserve les traces de l'exécution des tests dans des fiches de résultat. Il effectue un reporting de l'avancement au chef de projet à une périodicité convenue avec ce dernier. En fin de phase de test, le testeur réalise un bilan qu'il fournit au chef de projet.	Fiche de reporting Fiche de résultat de test Bilan au niveau du référentiel

Tableau 5 : synthèse des activités et livrable du processus de test

Pour aider le chef de projet un « Manuel d'utilisation du processus de test » lui est remis. Il contient un ensemble de modèles de documents et le descriptif sur la façon dont ils seront à renseigner.

Ce manuel se compose de :

- la politique de test,
- un modèle de référentiel des exigences et des tests,
- un modèle de plan de test,
- un modèle de fiche de description des tests,
- un modèle de fiche de reporting,
- un modèle de fiche de résultat de test.

La Figure 8 page 29 présente les nouvelles activités du processus de test (blocs en vert) positionnées au niveau du processus de développement. À ces étapes sont associés les nouveaux documents à compléter tout au long du projet de développement du produit.

Améliorer le processus de test pour atteindre le niveau 1 de TPI

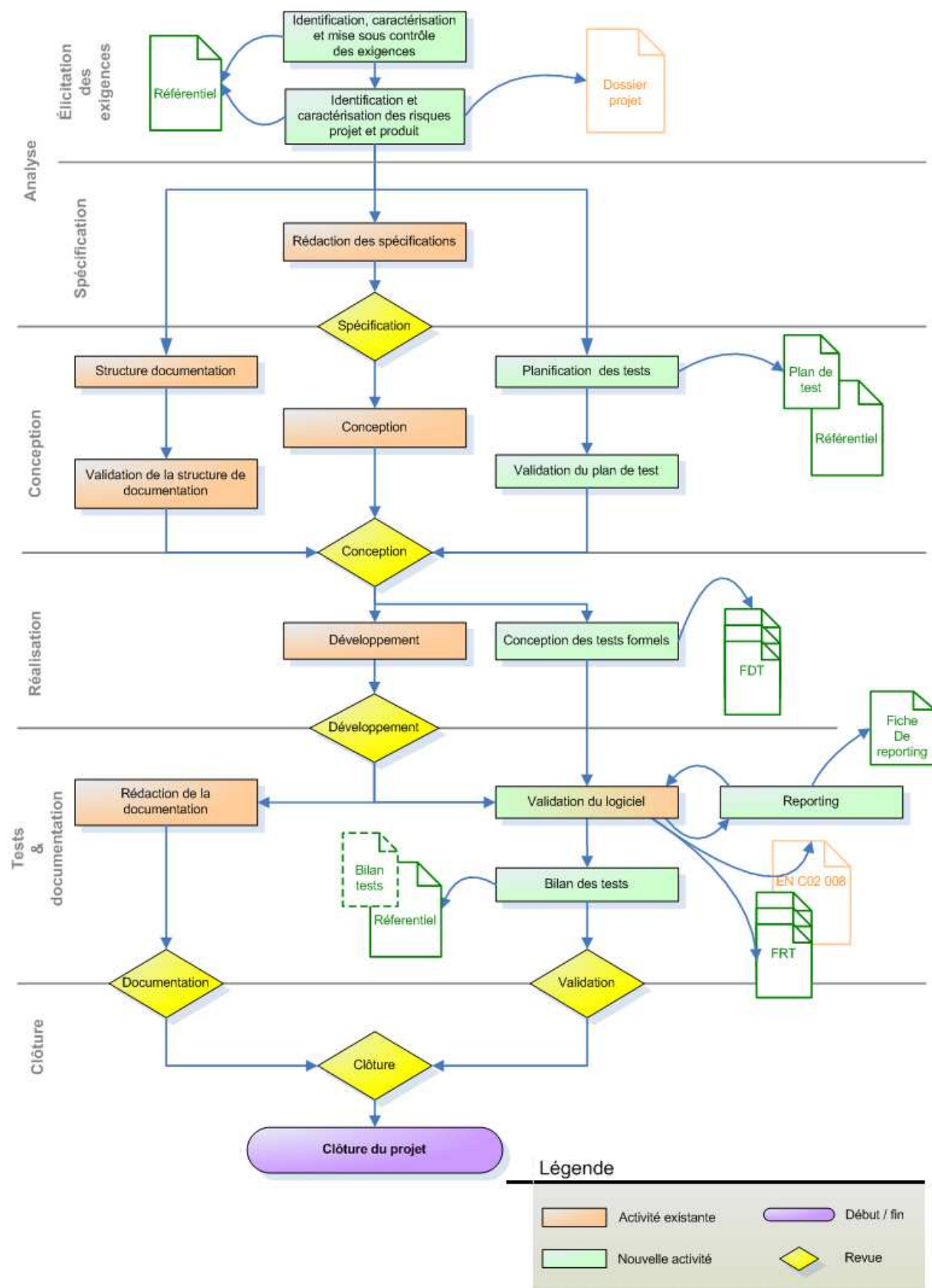


Figure 8 : les activités de test intégrées au processus de développement



Identification, caractérisation et mise sous contrôle des exigences

Lors de cette activité, le chef de projet dresse la liste des exigences dans le référentiel. Les exigences fonctionnelles ou non fonctionnelles sont résumées en une phrase avec un identificateur, un niveau de priorité et un lien entre les exigences. Cette liste est maintenue à jour par le chef de projet tout au long du projet de développement du produit. La zone du référentiel dédiée à cette activité est présentée dans la section 3.3.2 Gestion des exigences.

La zone du référentiel sert à partager et accepter avec l'ensemble des intervenants les mêmes références dans le but de fournir un produit conforme aux attentes du client.

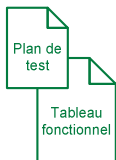


Identification et caractérisation des risques produit et projet

Un risque produit est un risque qui porte sur les exigences fonctionnelles du produit (livraison d'un logiciel défectueux, caractéristiques logicielles de moindre qualité,...) par opposition à un risque projet lié à l'organisation du projet et qui a un impact sur le coût ou le délai du projet (disponibilité des intervenants, utilisation de nouveaux outils ou technologies, maîtrise du processus de test,...)

Les risques produits sont identifiés et caractérisés dans le référentiel. La zone du référentiel dédiée à cette activité est présentée dans le chapitre 3.3.2 Gestion des exigences.

Les risques projet liés aux tests sont gérés dans le dossier projet. Le dossier projet a pour but de formaliser le suivi des projets dans un cadre unique et commun. Ce document est défini par l'équipe projet en charge de l'amélioration de la gestion de projet.



Planification de test

Le plan de test est défini par le testeur avec le soutien du chef de projet et validé par toute l'équipe projet. Il sert à :

- organiser le temps et le travail de tests,
- préparer le travail en amont,
- affiner les spécifications et les estimations,
- passer en revue les choix liés aux tests.

Il contient les exigences à tester ou à ne pas tester, l'ordre des tests, et toutes les informations concernant l'organisation des tests. La réalisation du plan de test s'effectue au niveau du référentiel (La zone du référentiel dédiée à cette activité est présentée dans le chapitre 3.3.2 Gestion des exigences). Dans certaines situations (projet de taille importante, mise en place d'environnement de test particulier, intervention d'expert en tests externe, ...) le plan de test défini au niveau du référentiel peut être complété par un document Word dont le modèle est basé sur la norme IEEE 829 [16].

Les objectifs et la stratégie de test sont définis au niveau de la politique de test. Cette dernière est présentée dans le chapitre 3.2 Le document « Politique de test ».

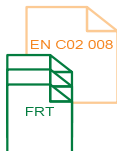


Conception des tests formels

L'activité de conception de test vise à définir la démarche à suivre pour les tests et de réduire leurs temps d'exécution. Les tests sont décrits de manière à pouvoir être exécutés par une personne non experte du logiciel à tester. La création de ces tests vise également à pouvoir les utiliser comme test de non régression. Pour cela, une Fiche de Description de Test (FDT) est créée pour chaque exigence devant être testée de manière formelle. Elle contient :

- l'état initial et final attendu,
- la séquence d'opération à réaliser,
- l'environnement du test.

Les fiches de description de tests sont rédigées par le chef de projet, le testeur ou le développeur et vérifiées par une personne extérieure au projet. Le détail du contenu et la structure d'une FDT est présenté dans le chapitre 3.4.1 La fiche de description de test (FDT).



Validation du logiciel

La phase de validation consiste en l'exécution des tests. Un dossier de validation est créé et rempli par le testeur. Le dossier de validation est un document regroupant l'ensemble des informations relatives à l'exécution des tests. Il contient un tableau récapitulatif des tests effectués dont un exemple est présenté en Figure 9. Ce tableau constitue le livrable contractuel pouvant être fourni aux clients lorsqu'ils demandent des preuves de tests. Ce tableau référence :

- un numéro de test,
- l'exigence testée (colonne « Titre »),
- la manière dont le test a été réalisé (selon le cas, des liens vers d'autres documents est indiqué) ainsi que le résultat à l'issue de l'exécution (colonne « Commentaire / Résultats »),
- L'identifiant du testeur,
- La date d'exécution.

Test	Titre	Commentaire / Résultats Valide /NON valide	Test réalisé par	Date
1	FP2.2 : Affichage des coordonnées de chaque point	Test informel Validé	RF	20/09/10
2	FP2.3.1 : Permettre de définir avec la souris une zone rectangulaire du graphique à zoomer.	Test semi-formel Voir fiche « test n°2 » Validé	RF	19/09/10
3	FP5.1.2 : Recherche de fichiers	Test formel FDT_FP5.1.2_VIEWER_V1.0.doc FRT_FP5.1.2_WIEVER_V1.0.doc Validé	RF	04/10/10

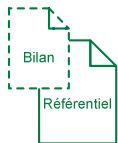
Figure 9 : exemple de tableau récapitulatif des tests dans le dossier de validation



Reporting

Les reportings sont faits par le testeur pour le chef de projet à intervalles réguliers, la périodicité étant convenue entre le testeur et le chef de projet. Un modèle de document a été créé afin de formaliser le reporting de manière commune à l'ensemble des projets.

Il sert à avoir une vision de l'état d'avancement des tests, du reste à faire et à contrôler le planning. Il contient l'état de l'activité des tests (non commencé, en cours, terminé) et son avancement (nombre de tests réalisés et le consommé par rapport au budget total).



Bilan des tests

Le bilan des tests est rédigé par le testeur. Il sert à faire un retour d'expérience avec une vision synthétique entre objectifs et résultats et ainsi capitaliser pour les prochains projets.

Il contient :

- les indicateurs de couverture : % exigences couvertes, % non couvertes, % non testables,
- les écarts par rapport au plan de test,
- un état des anomalies : nombre détectées, corrigées, non corrigées,
- le nombre de test (formel, semi formel, informel) par exigences,
- le coût du test (temps passé pour réaliser l'ensemble des activités du processus de test par rapport au temps total du projet).

Le bilan des tests se définit au niveau d'une zone du référentiel dédié à cet effet. Une description détaillée de cette zone est présentée au chapitre 3.3.4 Bilan des tests.

3.2. Le document « Politique de test »

La politique de test (PdT) décrit la philosophie de l'organisation des tests au sein d'un projet. Il a été mis en place un document Politique de test pour l'entreprise, où l'on trouve la définition des objectifs généraux de test que le département Produits souhaite atteindre. Elle vise également à référencer un standard pour la terminologie des tests à utiliser. Elle apporte ainsi une réponse à l'exigence TPI0 du projet d'amélioration, à savoir : définir une politique de test.

Le Tableau 6 page 33 présente, de manière synthétique, une échelle des niveaux de test qui est le cœur de la PdT. Cette échelle est constituée de 5 niveaux allant du niveau « très faible » à « très bon » et visant à instaurer des objectifs communs et en cohérence entre les différents projets de développement.

Chaque niveau est défini par quatre critères qui sont :

- **le référentiel des exigences** : indique la qualité des données d'entrée du processus de test,
- **le formalisme** : indique la manière de formaliser et de tracer les tests,
- **la couverture** : est un objectif en termes de couverture des exigences définies dans le référentiel,
- **les testeurs** : indique la provenance des personnes impliquées dans le test.

Niveau	Référentiel des exigences	Formalisme	Couverture		Testeur
			élevé	bas	
Très faible	pauvre	Non testé ou informel	≥ 0%	≥ 0%	BEI et/ou client
Faible	Pauvre + version antérieure existante	informel	≥ 50%	≥ 0%	BEI PRD (client)
Moyen	Riche + version antérieure existante	Σ (Semi formel ou informel) = objectif fixé	≥ 80%	≥ 0%	BEI PRD (client)
Bon	Riche priorité exigence + version antérieure existante	Σ (formel + Semi formel ou informel) = objectif fixé	= 100%	≥ 50%	BEI PRD (client)
Très bon	Riche + priorité exigence + version antérieure existante + niveau de tests antérieur bon ou très bon	Σ (formel + Semi formel ou informel) = objectif fixé + automatisation	= 100%	≥ 80%	BEI PRD (client)

Tableau 6 : l'échelle de niveaux de test

Lorsqu'aucun niveau n'est imposé au lancement d'un projet de développement, le chef de projet définit le niveau qu'il désire atteindre. Partant de là, la personne en charge de mettre en œuvre le processus de test au sein du projet connaît, grâce aux critères, les objectifs de tests et est ainsi en mesure de définir une stratégie et une organisation de test visant à atteindre cet objectif.

À la fin d'une campagne de tests, un bilan doit être réalisé et a pour but de fournir une vision synthétique de l'écart qui existe entre le niveau de tests défini en début de projet dans le plan de test et le niveau réellement atteint à la fin de la campagne.

Dans la section 4.2.2 - Activité intégration de la politique de test, l'ensemble des éléments constituant cette échelle de niveau de tests est présentée de manière détaillée afin d'aider à sa mise en application.

3.3. Le référentiel des exigences et des tests

Le référentiel est un classeur Excel dans lequel le projet liste les exigences fonctionnelles d'un produit logiciel afin de :

- les caractériser,
- les gérer tout au long du projet,
- gérer les risques produit associés,
- gérer les tests associés,
- réaliser le bilan des tests.

L'outil est constitué de plusieurs parties matérialisées par des onglets. C'est ainsi que le référentiel d'un processus de test comprend : un onglet Gestion des configuration (cf. 3.3.1), un onglet Gestion des exigences (cf. 3.3.2), un onglet Gestion des tests (cf. 3.3.3), un onglet Bilan des tests (cf. 3.3.4). Ce référentiel est initialisé au début du projet en tenant compte de la politique de test du projet.

3.3.1. Gestion de configuration

Comme beaucoup d'éléments gérés lors d'un projet, le référentiel des exigences et des tests est amené à évoluer au cours du projet. L'outil, étant basé sur un classeur Excel, n'a pas de numéro de version. Le nom du classeur reste identique. Un seul tableau est associé à un seul projet. Chaque projet contient donc un référentiel.

Un onglet du fichier, nommé « Historique », permet de tracer les modifications faites sur tout le document Excel. Il se compose de trois colonnes à remplir :

- **qui** : permet de renseigner la personne qui a réalisé la modification,
- **commentaires et modifications** : section qui permet de lister les modifications réalisées,

- **quand** : permet de renseigner les dates des modifications.

Cet onglet est complété par la personne effectuant des modifications sur le fichier.

3.3.2. Gestion des exigences

La gestion des exigences est réalisée au niveau d'un onglet nommé « Gestion des fonctions ». Celui-ci permet, comme son nom l'indique, de gérer les exigences fonctionnelles. Cette gestion est réalisée selon divers points de vue :

- l'identification,
- l'état et les caractéristiques,
- les risques,
- le budget,
- les tests.

Chacun de ces points de vue est traité par une zone de l'onglet. Chacune de ces zones est décrite dans la suite de cette section.

3.3.2.1. Identification et description des exigences fonctionnelles

Cette zone, présentée en Figure 10, permet de lister l'ensemble des exigences fonctionnelles du produit logiciel. Elle est remplie par le chef de projet en début de projet, avant d'écrire les spécifications, puis compléter tout au long de la phase de spécifications.

Identification et description		
N°	Type	DESCRIPTION DES FONCTIONS
<i>nn</i>	<i>FPnn:</i>	<i>Fonction Principale attendue pour répondre à un besoin.</i>

Figure 10 : zone dédiée à l'identification et la description des exigences fonctionnelles

Cette zone est composée des trois colonnes suivantes :

- N°,
- Type,
- Description des fonctions.

Colonne N°

La colonne « N° » référence le numéro de la fonction et des sous-fonctions. Exemple : 1.0 pour la première fonction principale 4.3 pour la troisième sous fonction de la fonction principale 4.

Colonne Type

La colonne « type » référence le type de fonction et son identificateur. Une fonction principale et une sous fonction ont le même identificateur mais le numéro associé définira le niveau de la fonction.

- **Fonction principale** (identifiant de type : FP) : C'est la fonction qui satisfait le besoin. Elle assure la prestation du service rendu. C'est la raison pour laquelle le produit a été créé. Ex pour un calendrier : FP1 : Fournir des informations sur la date.
- **Sous fonction** (identifiant de type : FP) : C'est une action attendue d'un produit pour répondre à un élément du besoin. Ex. (Calendrier) FP1.1 pouvoir consulter le calendrier en vue journalière, FP1.2 : pouvoir consulter le calendrier en vue hebdomadaire, etc.
- **Niveau et lien entre fonctions** : Une fonction peut être répartie en plusieurs sous fonctions. Pour gérer ce niveau et lien entre fonctions principales et sous fonctions une séquence de nombres, séparés par un point est ajoutée à la suite de l'identifiant de type. Son rôle est double :
 - permettre de différencier les exigences les unes des autres,
 - permettre d'identifier les liens qui existent entre les fonctions et sous fonctions.

Cette séquence définit le niveau hiérarchique des exigences. Le nombre le plus à gauche indique le numéro de l'exigence. S'il est unique dans la séquence, cela signifie que l'exigence est de niveau 1 (niveau le plus élevé). Au fur et à mesure que des nombres apparaissent à la droite du numéro de niveau 1, ils indiquent :

- le niveau hiérarchique de la sous fonction (niveau 2 ; 3 ou +),
- identifient celle-ci par rapport à l'ensemble des sous fonctions de même niveau.

Ex : L'identificateur FP18.1.2 signifie que l'exigence est une sous-fonction.

Le nombre « 18 » indique qu'elle est rattachée à la 18^{ème} exigence de niveau 1.

Le nombre « 1 » indique qu'elle est rattachée à la 1^{ère} sous-fonction.

Le nombre « 2 » indique que c'est la 2^{ème} sous-fonction de la sous-fonction FP18.1.



Cette identification des exigences correspondant à un besoin de traçabilité, un numéro et un type d'exigence créés ne doivent pas être détruits sous peine de le réutiliser pour une exigence future sans rapport avec la précédente. Pour une exigence qui n'a plus cours, le numéro est conservé et l'état de la fonction (cf.3.3.2.2) est placé à « Annulé ».

Colonne Description des fonctions

Cette colonne contient la description de la fonction en une ou deux phrases. Cette colonne doit permettre de cerner rapidement ce que fait la fonction.

La description d'une fonction commence toujours par un verbe à l'infinitif ou un groupe verbal caractérisant l'action, et suivi de compléments représentant les éléments du milieu extérieur concernés par la fonction. Le sujet de la phrase n'apparaît pas, mais il renvoie toujours au produit.

Règles d'usage à respecter :

- la forme passive est à éviter,
- la forme négative est à éviter.

Exemple de description :

- Enregistrer les paramètres renseignés par l'utilisateur.
- Afficher les informations du fichier.

3.3.2.2. Caractérisation et gestion des états

Cette zone (Figure 11) permet de définir les divers critères caractérisant l'exigence fonctionnelle du produit logiciel. Elle est remplie par le chef de projet. Les colonnes constituant cette zone sont :

- classes,
- fournisseur,
- engagement,
- état,
- échéance,
- raison.

Classes	FOURNISSEUR	Engagement	État	Échéance	Raison (Annulé / reporté)
	<i>Nom de l'entreprise + Nom de l'interlocuteur</i>		<i>Acc = 'Accepté' Att = 'En Attente' Ann = 'Annulé' Rep = 'Reporté'</i>	<i>En cas d'attente de la décision (client, interne...), spécifier la date d'échéance. Sinon => Fonction annulée</i>	<i>En cas d'annulation ou de report, définir pour quelle raison.</i>

Figure 11 : zone dédiée à l'identification et la description des fonctions

Colonnes Classes

La classe permet de définir le niveau de priorité d'une exigence. Cette colonne est remplie par le chef de projet en concertation avec les divers intervenants du projet (équipe projet, client, ...). Cette priorité se définit au travers de 4 niveaux classés par importance décroissante :

- F0 : Impératif,
- F1 : Peu négociable,
- F2 : Négociable,
- F3 : Très négociable.

Fournisseur

Un fournisseur est une entité exprimant un besoin, sous-entendu « celui qui fournit le besoin » (exemple : une personne, un service, un département ou une entreprise). Il est identifié par 2 données : le nom de l'entreprise et le nom d'une personne. Cette identification est indiquée, par le chef de projet, dans la colonne « fournisseur » lors de l'ajout d'une exigence dans le référentiel.

L'identification et la conservation de l'information concernant le fournisseur de l'exigence est important pour les raisons suivantes :

- permettre d'affiner la priorité, les risques, les tests, ... des exigences,
- indiquer l'origine d'une exigence, elle peut provenir d'un ou plusieurs demandeurs,
- clarifier l'exigence : si elle est conflictuelle ou nécessite d'être modifiée ou supprimée, son fournisseur peut être consulté ou informé,
- vérifier que l'exigence est bien satisfaite (recette demandeur).

Engagement

La colonne « Engagement » permet de s'assurer, au travers de deux étapes, que toutes les personnes, amenées à réaliser une tâche liée à la fonction, ont procédé à une:

- **évaluation** de la fonction afin de s'assurer qu'ils sont en mesure de réaliser la tâche (exemple : le testeur s'assure que la fonction est testable et le développeur qu'elle est réalisable...). Cette évaluation est faite lors de réunions techniques ou lors d'analyses individuelles de l'exigence.
- **acceptation** de réaliser la tâche liée à la fonction. Cette seconde étape finalise le travail d'évaluation de la fonction.

L'information de cette colonne se fait exigence par exigence afin de s'assurer du traitement de l'ensemble des exigences. Le chef de projet est tenu de compléter la colonne « Engagement » par une réponse binaire ('Accepté' ou 'Non-accepté') en concertation avec l'équipe projet.

État

La colonne « État » informe sur l'état courant de l'exigence. Cette colonne est remplie par le chef de projet en concertation avec le responsable produits. Quatre états sont possibles :

- **accepté** : la fonction est validée et sera développée dans cette version du produit,
- **en attente** : la fonction doit être clarifiée (en interne, avec le client, etc.),
- **annulé** : la fonction ne sera pas développée dans cette version du produit ni dans les versions suivantes,
- **reporté** : la fonction ne sera pas développée dans cette version mais repoussée dans une version ultérieure (pour cause de stratégie commerciale, par manque de moyens, etc.).

Échéance

La colonne « Échéance » contient la date jusqu'à laquelle la fonction est « en attente ». Cette colonne n'est à remplir que si la fonction est en attente d'information, d'explication, etc.

Dans le cas où la date d'échéance n'est pas respectée, une décision est prise sur le maintien ou non de l'exigence. Cette décision est prise, au plus tard et en dernier recours, en revue de spécification lors de la validation du référentiel. Ceci évite les risques de dérives du nombre d'exigences à développer durant le projet.

Raison

La colonne « Raison » contient les raisons de l'annulation ou du report de la fonction. Cette colonne doit permettre de connaître les causes d'un refus de développement et garder une trace pour les projets futurs.

3.3.2.3. Gestion des risques produit

La gestion des risques permet d'obtenir une vision commune pour le chef de projet et les autres acteurs clés, des caractéristiques et des parties plus ou moins risquées du produit à développer et à tester.

Une zone, présentée en Figure 12 (page 38), au niveau de l'onglet « gestion des exigences » est dédiée à cet effet. La présence de cette zone dans le référentiel correspond à la solution apportée à l'exigence TPI1.1 : identifier les risques en fonction des exigences. Cette partie est complétée par le chef de projet en collaboration avec son équipe.

Les colonnes composant cette zone sont :

Améliorer le processus de test pour atteindre le niveau 1 de TPI

- niveau du risque,
- catégorie du risque,
- cause du risque,
- description du risque,
- action à mettre en place.

Risques				
Niveau du risque	Catégorie du risque	Cause du risque	Description du risque	Action à mettre en place
Niveau du risque : - faible - moyen - élevé	Catégorie du risque (Produit, Moyens / Matériel, Main d'œuvre, Méthodes, Management, Milieu, Coût, Client)	Cause du risque en rapport avec son type		Actions préventives et/ou curatives à mettre en place pour lever le risque

Figure 12 : zone dédiée à la gestion des risques produit

Niveau du risque

Le degré du risque sur la fonction est laissé à l'appréciation du chef de projet. Les risques sont classés en trois niveaux :

- **faible** : l'apparition du risque n'impacte que très légèrement le projet et les tests à réaliser,
- **moyen** : un risque moyen peut entraîner un dysfonctionnement partiel d'une exigence,
- **élevé** : un risque élevé peut entraîner la suppression de l'exigence et remettre en cause tout le projet.

Catégorie du risque

Cette colonne définit le type de risque associé à l'exigence parmi les divers types proposés dans le Tableau 7 ci-dessous.

Type de risque	Description
Produit	risques techniques liés au produit,
Moyens / Matériels	risques liés au matériel utilisé (nouveau matériel, indisponibilité, etc.)
Main d'œuvre	risques liés au personnel travaillant sur le projet,
Méthodes	risques liés aux méthodes de travail,
Management	risques liés à la gestion du projet (planning serré, etc.)
Milieu	risques liés à l'environnement de travail (travail à distance, etc.)
Coût	risques liés aux coûts du projet (dépassement de budget, etc.)
Client	risques liés au client (contraintes clients, etc.)

Tableau 7: type de risques produit

Cause du risque

Une description en quelques lignes des causes du risque.

Description du risque

Une brève description du risque et des choix faits pour son niveau et ses causes.

Action à mettre en place

Description des actions mises en place pour pallier ou anticiper le risque afin de diminuer les impacts qui peuvent nuire à l'atteinte des objectifs. Si possible ces actions devront être quantifiées.

3.3.3. Gestion des tests

La gestion des tests est réalisée au niveau de l'onglet « Gestion des fonctions » dans une zone prévue à cet effet. Le plan de test y est défini par le testeur en collaboration avec le chef de projet. Il est validé par toute l'équipe projet afin de s'assurer de sa complétude et de sa cohérence avec les objectifs généraux du projet de développement. La définition du plan de test au niveau du référentiel est la solution mise en œuvre pour répondre à l'exigence TPI2.1 du projet d'amélioration.

La définition du plan de test peut commencer dès que la liste des exigences à réaliser est connue et arrêtée. Les objectifs d'un plan de test sont :

- organiser le temps et le travail de test,
- préparer le travail en amont et donc de paralléliser une partie des activités de test avec d'autres activités du projet,
- affiner les spécifications (en se posant les bonnes questions très tôt dans le processus de développement) et les estimations,
- passer en revue les choix liés aux tests car les choix de test se font à un moment du projet où ils ne sont pas encore sur le chemin critique du projet.

Globalement, il sert à avoir une vision synthétique entre objectifs (définis dans le plan) et résultats (obtenus à la fin de la phase de test par rapport au plan) et ainsi capitaliser pour les prochains projets.

Selon les circonstances (taille du projet de développement important, complexité des tests à réaliser,...), ce plan peut être complété et détaillé dans un document Word dont la structure est basée sur la norme IEEE 829 (norme dédiée à la documentation des tests).

Cette zone, présentée en Figure 13, est constituée des éléments suivants :

- niveau de tests souhaité,
- formalisme,
- temps de validation prévu,
- lien vers document de test,
- formalisme réalisé,
- temps de validation réalisé.

Niveau de test souhaité :				
Bon				
plan de Tests				
Tps de validation prévu (h)	Formalisme	Tps de validation réalisé (h)	Formalisme réalisé	Lien vers documents de test
146		169		
12	Formel	16	Formel	FDT_12_GuideDeConduite_V5.0
12	Formel	4	Semi-formel	dossier validation
12	Formel	16	Formel	FDT_15_GuideDeConduite_V5.0
2	Informel	1	Informel	dossier validation

Figure 13 : zone dédiée à l'organisation des tests

Niveau de tests souhaité

En tête de cette zone, une case est prévue pour indiquer et conserver le niveau de tests à atteindre. Ce niveau est défini en début de projet. Il est l'une des informations qui permettra de définir le plan de test (cf. 4.2.2.5 Les niveaux de tests).

Formalisme

Le formalisme correspond à la définition de la manière de formaliser le test pour une exigence. Les formalismes possibles sont ceux prévus et définis dans la section 4.2.2.2 La formalisation des types de test (informel, semi-formel, formel). Le niveau de formalisme est un des facteurs permettant d'estimer du temps nécessaire à la réalisation des tests sur l'exigence.

Deux autres choix sont disponibles en plus de ces trois formalismes. Ces deux cas ne sont pas des manières de formaliser des tests. Ces choix sont :

- **non testé** : ce choix indique que l'exigence ne sera pas testée. Ce choix est une décision du chef de projet,
- **non testable** : Une exigence est désignée « non testable » lorsque des caractéristiques ou une partie des caractéristiques de l'exigence sont incomplètes ou inexistantes et ne permettent donc pas de tester l'exigence. Ce choix est sélectionné par le testeur pour indiquer que l'exigence ne sera pas testée.

Temps de validation prévu

Cette colonne est dédiée à l'estimation du temps nécessaire pour la réalisation de la phase de test du logiciel. Elle est remplie par la personne en charge des tests en indiquant un temps de réalisation par exigence (en heure). Le temps sera dépendant du niveau de formalisme souhaité. Le testeur aura, avant de renseigner cette colonne, défini le formalisme.

Le temps total est automatiquement calculé (somme des temps par exigence) en tête de colonne. Lorsque le temps de réalisation est imposé, il est alors possible de contrôler l'adéquation des choix réalisés exigence par exigence avec le budget alloué.

Exemple d'estimation du temps de validation suivant le formalisme des tests :

Informel	2 heures
Semi formel	4 heures
Formel	12 heures



dès lors que les colonnes « formalisme » et « temps de validation prévu » sont remplies et validées, elles ne doivent plus être modifiées durant le projet, pour permettre de capitaliser avec le temps réalisé.

Lien vers document de test

Cette colonne permet de référencer (ex. un lien hypertexte) le chemin du ou des documents dédiés au test (conception du test, résultats d'exécution,...).

Suivi et maîtrise des tests

Deux colonnes de la zone « plan de test » sont destinées au suivi et à la maîtrise des activités de test ainsi que d'identifier les écarts entre ce qui a été prévu initialement au plan et ce qui effectivement été réalisé. L'objectif étant de capitaliser l'expérience acquise. L'analyse des écarts prévu/réalisé doit permettre d'identifier les bonnes pratiques ou les erreurs afin de proposer des actions d'amélioration. Le suivi et la maîtrise des activités de test sont réalisés grâce aux colonnes :

- **formalisme réalisé** : permet de renseigner le niveau de formalisme réalisé au cours du projet qui peut être différent de celui initialement prévu et indiqué dans la colonne formalisme,
- **temps de validation réalisé (h)** : cette colonne permet de renseigner le temps passé à la validation de l'exigence.

3.3.4. Bilan des tests

Pour réaliser le bilan des tests un onglet est prévu à cet effet. Celui-ci est en partie rempli automatiquement grâce aux informations saisies dans l'onglet « gestion des exigences ». Cet onglet a un objectif double. Lors de la phase de test, il permet d'effectuer un suivi de la progression des tests. À la fin de la phase de test, il permet de faire le bilan. Il répond à l'exigence d'amélioration du processus de test TPI2.4 : réaliser le bilan des tests.

Bilan de fin de test

Le bilan sert à faire un retour d'expérience avec une vision synthétique entre objectifs et résultats et ainsi capitaliser pour les prochains projets.

Le bilan est présenté sous 4 aspects qui sont :

- la couverture et écarts,
- les anomalies,
- le budget,
- le retour d'expérience.

Pour chacun de ces aspects, un ensemble de données est regroupé sur une page du document. Chacune de ces pages est présentée dans la suite de ce chapitre.

La couverture et écarts

Dans cette partie est indiqué l'écart entre ce qui était prévu dans le plan de test et ce qui a été réellement réalisé en termes de niveau de tests, de nombre de tests et en temps de test exprimé en heures (cf. Figure 14 - tableau « Écart des tests »).

C'est grâce à cette partie du bilan qu'il est possible d'effectuer le suivi de la progression des tests. Deux diagrammes en camembert (Figure 14) permettent de visualiser rapidement la couverture des tests prévue initialement et leur avancement au cours du projet. Ces diagrammes présentent le niveau de couverture des exigences par les tests. Le diagramme de gauche indique la couverture souhaitée et exprimée en début de phase de test. Le diagramme de droite permet de visualiser la progression de l'exécution des tests en cours de phase.

À l'issue de la campagne de test, lorsqu'un écart existe, les raisons justifiant l'écart sont indiquées dans la zone prévue à cet effet située en bas de page. Par exemple, le niveau de tests attendu était «Bon» or le niveau atteint est «moyen». La raison de la non atteinte du niveau «bon» peut être dû au fait que la ressource prévue pour la réalisation des tests formels n'était pas disponible et de ce fait les tests formels nécessaires à l'obtention du niveau «bon» n'ont pas pu être réalisés.

BILAN DES TESTS				
Affaire II° :	0		Chef de projet :	0
Objet Affaire :	0		Version :	0
Famille :	0		Lot :	0
Équipe projet :	0			

Couverture et écarts

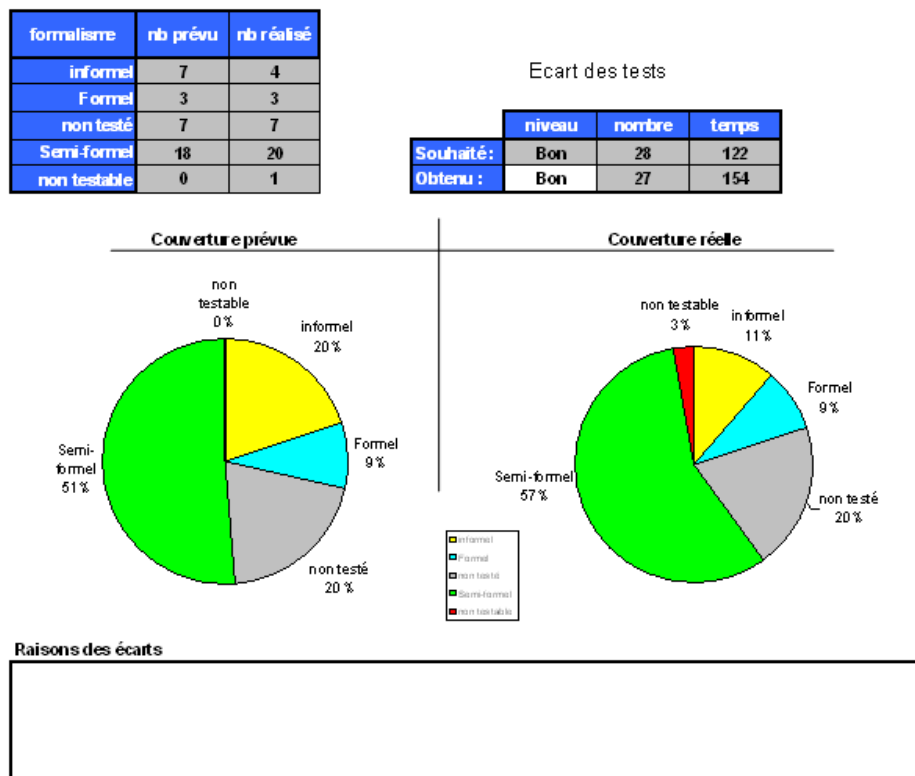


Figure 14 : page « couverture et écarts » du bilan

Anomalies

Cette partie (Figure 15) permet de dresser un bilan des anomalies. La gestion des anomalies est réalisée grâce à l'outil TRAC. Ce dernier a été mis en place peu de temps avant le lancement du projet d'amélioration. Le but est d'effectuer le bilan des anomalies est de fournir une vision synthétique du niveau de détection des anomalies et de leur correction, ainsi que de capitaliser des données qui permettront, après analyse, de déterminer de nouveaux axes d'amélioration. Le bilan des anomalies est réalisé selon 3 axes :

- Les anomalies détectées classées dans un tableau par type et par sévérité,
- Les anomalies corrigées,
- Les anomalies non corrigées. Cette partie a pour but de fournir les raisons pour lesquelles les anomalies n'ont pas été corrigées. Le Tableau 8 montre un exemple de justification d'anomalies non corrigées.

Id	Description anomalie	Type	Justification de la raison de non correction
9	Fonctionnement de la voie d'état du guide de conduite non conforme aux spécifications.	Non bloquant	Le fonctionnement proposé par le BEI convient au RP.
53	Erreur de calcul du temps dans les tolérances et hors tolérance.	Non bloquant	Ne pose pas de problème pour la livraison, sera corrigé dans la prochaine version.

Tableau 8 : exemple de justification d'anomalies non corrigées



Anomalies

Nombre détectées :

		Sévérité			
		haute	normale	basse	aucune
type	Bloquant				
	Non bloquant				
	Amélioration				
	Total	0			

Nombre Corrigées :	
Nombre non corrigées :	

Justification de la non correction des anomalies

Id anomalie	Description anomalie	Type	Justification de la raison de non correction

Figure 15: page « anomalies » du bilan

Budget

Le budget sera analysé selon 2 axes, d'une part en le comparant au budget total du projet et d'autre part en rapport avec chacune des tâches qui le compose. L'objectif étant de faire prendre conscience de la nécessité d'une bonne organisation des activités du processus de tests au cours du projet de développement.

La première partie (Figure 16) a pour but de mettre en évidence le coût des tests en termes de temps par rapport un coût total du projet. Pour cela le coût consommé pour les tests et pour l'ensemble du projet est renseigné par la personne chargée de réaliser le bilan. Le coût des tests sera aussi exprimé sous forme de ratio.

$$\text{Coût_des_tests} = \frac{\text{durée_des_tests_}(en_hrs)}{\text{durée_de_projet_}(en_hrs)}$$

La seconde partie présente les écarts entre ce qui était prévu et le temps réellement consommé pour la réalisation de toutes les activités qui composent le processus de test.

BILAN DES TESTS	
------------------------	---

Budget

Coût du test (tps test / tps projet)

	Coût projet (en hrs)	Coût phase de test (en hrs)	ratio = $\frac{hrs_phase_de_test}{hrs_projet}$
Prévu			
Réalisé			
Ecart			

Ecart par rapport au plan

Phase	Durée prévue (en hrs)	Durée réalisée (en hrs)	E cart (en %)
Analyse des spécifications			
Rédaction du plan de test			
Exécution des tests			
Préparation de la plateforme de test			
confirmation correction constat			
Bilan			

Cette liste ne se veut pas exhaustive et est uniquement fournie à titre d'exemple.

Raisons des écarts

--

Figure 16 : page « budget » du bilan

Retour d'expérience

La dernière partie du bilan (Figure 17) permet de renseigner les problèmes rencontrés lors de la réalisation des activités de test (par exemple, matériel inadapté, dérives du logiciel, mauvaise compréhension d'une exigence, ...).

BILAN DES TESTS		
Retour d'experience		
Problèmes rencontrés		
Remarques		

Figure 17 : page retour d'expérience du bilan

3.4. La formalisation des tests

Cette section présente un plan type et des bonnes pratiques pour rédiger des fiches de description de tests (FDT) ainsi que les fiches de résultats de tests (FRT) associées. La mise en place de ces deux documents vise à répondre aux exigences d'amélioration du processus de test :

- TPI1.2 : établir une description des tests,
- TPI5 : documenter les techniques de construction de cas de test.

3.4.1. La fiche de description de test (FDT)

Une FDT est un document qui décrit l'exécution d'un test de manière suffisamment formelle pour que le test soit déterministe.

Une fiche de description de test a pour objectif de :

- donner la démarche à suivre lors de l'exécution du test,
- permettre de réduire, le plus possible, le temps d'exécution des tests,
- pouvoir réaliser des tests de non-régression,
- permettre à une personne non experte d'exécuter le cas de test.

La rédaction de FDT est principalement réalisée par les testeurs en collaboration avec le chef de projet. Les développeurs peuvent être amenés, dans le cadre d'un travail de test, à rédiger des FDT.

La FDT correspond à une exigence TPI. Il fallait la créer et je me suis basé sur l'état de l'art en l'occurrence une structure s'inspirant des concepts proposés par la méthode TMap et les cas d'utilisation selon Alistair Cockburn [20]. En effet, l'équipe en charge de proposer une solution à la formalisation des tests n'arrivant pas à proposer une structure de document, j'ai proposé cette structure que l'équipe s'est, par la suite, appropriée en la faisant évoluer selon leurs besoins.

Ces fiches sont destinées dans un premier temps au testeur pour la phase d'exécution des tests suite à la livraison des exécutables. Dans un second temps, elles seront utilisées pour les tests de non-régression durant toute la vie du produit et pourront être exécutées par des personnes (non expertes) autre que le créateur de la fiche (par exemple, un rédacteur technique pour contrôler la cohérence de la documentation par rapport au produit, un metteur en service pour effectuer un contrôle préalable avant une mise en service, un nouvel embauché n'ayant pas encore de compétence du produit à tester,...)

Une fiche de test fournit principalement des informations sur :

- l'état initial du système à tester avant de commencer le test,
- la séquence d'opérations à réaliser pour effectuer le test,
- l'état final attendu après l'exécution de la séquence,
- l'environnement du test utilisé.

La rédaction de ces fiches commence après la validation du plan de test, en parallèle de la phase de développement (cf. Figure 18). Ceci afin qu'elles existent lors de la livraison des exécutables et afin de permettre de réduire le temps de la phase d'exécution des tests. Cette dernière se situe généralement sur le chemin critique du projet de développement. Tout allongement de cette phase entraîne un allongement de la durée du projet. Une telle fiche est rédigée lorsque la décision est prise de tester une fonction selon un test de type formel (cf. 4.2.2.2 La formalisation des types de test).

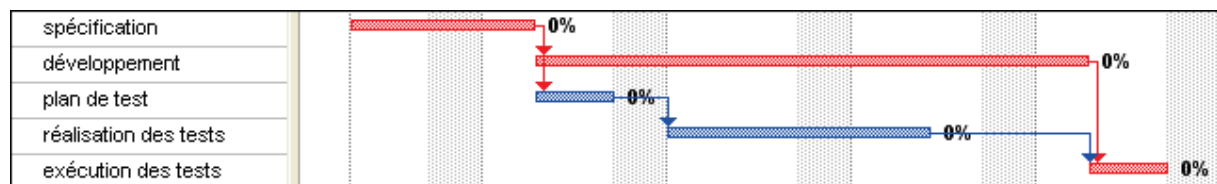


Figure 18 : ordonnancement des activités de tests

3.4.1.1. Gestion de configuration du document

Dénomination du document

L'objectif de la dénomination est de définir la règle permettant de nommer de manière explicite les fiches de tests des différents produits logiciels de D2T. L'intérêt qui en découle est une homogénéisation des dénominations ; ainsi le testeur est en mesure de nommer aisément les fiches de tests tout en facilitant leur recherche au niveau de l'archivage.

Règle : Le nom du fichier se construit de la manière suivante :

FDT_Type n° fonction_NomDuLogiciel_N°versionDuLogicielSousTest.doc

Les données importantes à retrouver dans le nom du fichier (systématiquement séparés par un tiret bas) sont :

- le **type de document** : fiche de description de test (FDT),
- l'**identificateur de la fonction** : correspond au type et au numéro de la fonction testée définit dans les colonnes « Type » et « N° » de l'onglet « gestion des fonctions » du référentiel,
- le **nom du produit logiciel** (Editeur_MORPHEE, exécutif_MORPHEE, Viewer, OSIRIS, TESTMANAGER, CAN, OSIDCOM,...),
- le **numéro de version du produit logiciel** pour lequel la fiche a été créé. Ce numéro n'est constitué que par les numéros de version majeure et évolution (le numéro de version mineur n'est pas utilisé). Ce numéro ne change que si la fiche est impactée par une modification du produit. Par exemple, la fiche a été créée pour tester la fonction FP2.3 lors de la version 1.0 du produit. Lors du développement de la version 1.1, du logiciel, la fiche peut être ré-exécutée en l'état (aucune modification du logiciel n'impacte la fiche). Le nom du fichier ne change pas. Lors de la version 1.2, une évolution du logiciel ne permet plus de passer le test décrit par la fiche. Il faut la faire évoluer. Le nom du fichier change en faisant évoluer le numéro de version de 1.0 à 1.2 et ainsi reflète l'adaptation de la fiche à l'évolution due à la version 1.2.



Exemple : FDT_FP6.1.1_Osiris_5.7.doc

Gestion de configuration

La version du document n'est pas indiquée dans le nom du document. Sa version et les modifications, relatives à la version, sont référencées dans le tableau « historique » (Figure 19) situé à l'intérieur du document.

Historique			
N° version	Date	Commentaires	Auteur
0.1	12/06/2010	création	RF
1.0	15/07/2010	Fiche validée et opérationnelle	RF

Figure 19 : tableau de gestion des versions

3.4.1.2. Pré-requis

Les fiches de test ne peuvent être définies qu'une fois les spécifications validées. Les spécifications doivent être suffisamment précises pour qu'une seule interprétation en soit possible. Si le rédacteur des fiches de test se pose des questions quant à l'interprétation des spécifications, il doit impérativement chercher à les préciser le plus rapidement possible. Cela permettra de limiter les risques de non-conformités et de re-développement lors de la phase d'exécution des tests.

3.4.1.3. Description du document

Cette section présente l'ensemble des paragraphes qui constituent une fiche de description de test. Un cas de test permet de tester une exigence du système selon un objectif défini préalablement.

La plupart de ces paragraphes correspondent à un élément constituant un cas de test tel que : les jeux de test, le scénario, le résultat attendu,... (cf. Figure 20).

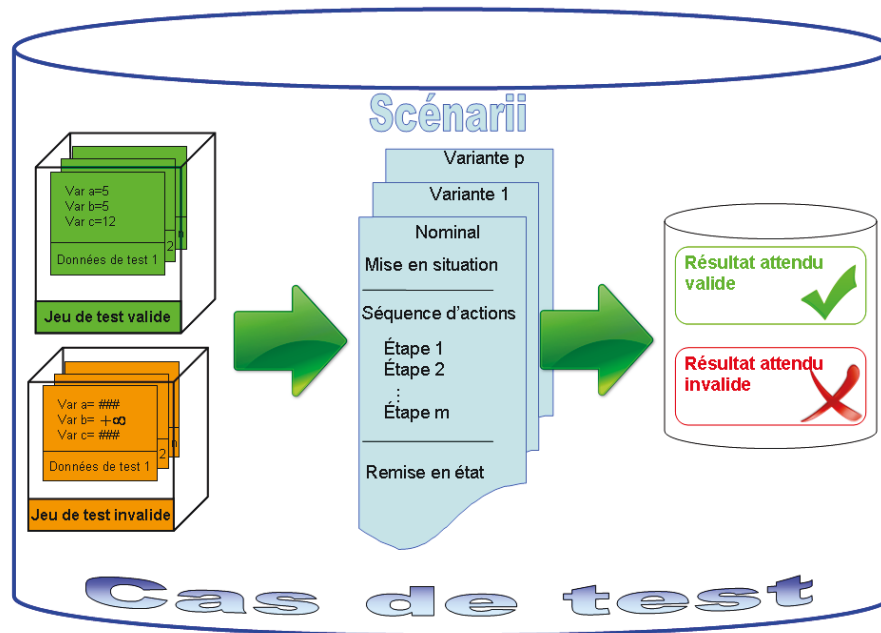


Figure 20 : éléments constituant un cas de test

Le cartouche

Famille :	3 – système d'automatisation	Produit :	<u>Viewer</u>
Fonction testée	FP5.1.2 - Recherche de fichiers		
Référence	..\02 - Spécifications\03- Spécifications\SPEC14_Viewer\0.1.pdf – chapitre 5.5.1 ..\02 - Spécifications\01- Analyse fonctionnelle\TF onctionnel VIEWER.xls		

Figure 21 : exemple de cartouche de FDT

Le cartouche d'une FDT est constitué de 4 éléments :

- **famille** : la famille à laquelle appartient le système à tester,
- **produit** : le nom du produit sous test,
- **fonction à tester** : cette zone contient l'identificateur et la description de la fonction telle que définie dans le référentiel (onglet : gestion des exigences),
- **référence** : cette zone contient les références relatives à la fonction. Au minimum la référence du dossier de spécification et le chapitre dans la spécification relatif à la fonction y sont indiqués.

Contenu d'une fiche de description de test

Une fiche de description de test contient les éléments suivants :

- la description du test,
- les données de test,
- les ressources intervenant,
- le scénario,

- les pré-conditions et post-conditions,
- le résultat attendu.

Description du test Dans ce paragraphe sont indiqués :

- l'objectif de test (ex : vérifier une règle liée à la fonction FP1.2 définie dans le dossier de spécification),
- un rappel succinct de la fonction à tester,
- une courte description du contexte dans lequel est mise en œuvre le scénario.

Données de test Sont Indiquées, dans cette section, les données utilisées pour la réalisation de ce test (fichier, valeur numérique ou chaîne de caractères,...)

Ressources intervenants Définition de toutes les ressources (matérielles, logicielles, humaines) qui interviennent dans le test.

Pré-condition Indique les conditions initiales pour démarrer la première étape du scénario.

Post-condition Indique les conditions pour remettre le système dans un état stable à l'issue du scénario. Les post-conditions sont optionnelles

Résultat attendu Il s'agit de tous les résultats (événements observables) que le testeur doit contrôler pour voir si le comportement du logiciel est conforme aux attentes. Cette partie doit renvoyer vers les spécifications voir si possible les citer afin de lever toute ambiguïté quant aux résultats attendus.

Scénario

Un scénario (cf. exemple présenté dans le Tableau 9) est une séquence d'opérations qui a pour but d'amener le système à tester d'un état initial vers un état final. L'état initial correspond à la condition dans laquelle le système et son environnement doivent se trouver avant de débiter la séquence. Cet état est atteint à l'issue de l'exécution des pré-conditions. L'état final correspond, quant à lui, au résultat attendu.

Un scénario est dit « nominal » lorsqu'il correspond à l'utilisation la plus courante d'une fonction. Il est appelé « variante » lorsque le déroulement correspond à des utilisations exceptionnelles. Ceux qui génèrent des erreurs, les cas de problèmes, les utilisations inhabituelles...

Étape	Action utilisateur	Réaction attendue du système
1	Saisie du lieu de naissance	Le lieu de naissance est affiché dans le champ correspondant
2	Choix du pays de naissance dans la liste proposée	Le pays de naissance est sélectionné
3	Validation de la saisie	Un message d'alerte indique que la date de naissance n'est pas remplie
4	Saisie de la date de naissance	La date de naissance est affichée dans le champ correspondant
5	Validation de la saisie	Les informations sont validées. Le champ âge est affiché.

Tableau 9 : exemple de scénario de test

Un scénario se présente sous la forme d'un tableau (cf. Tableau 9) constitué de trois colonnes.

Étape : permet d'identifier chaque action de la séquence par un numéro. Dans le cas

Améliorer le processus de test pour atteindre le niveau 1 de TPI

de variante du scénario, ce numéro peut commencer par un autre chiffre que 1. Cela indique l'action à partir de laquelle une différence apparaît entre le scénario nominal et la variante.

Action utilisateur : définie, sans ambiguïté, à partir des spécifications. Afin de limiter le nombre de cas de tests, les étapes redondantes (identiques aux actions et aux résultats attendus) dans les différents scénarios devront être factorisées ou éliminées.

Réaction attendue du système : est l'événement qui se produit à la suite de l'action associée. Le système ne produit pas systématiquement une réaction à la suite d'une action utilisateur. La saisie d'information de cette colonne est de fait optionnelle.

3.4.2. La fiche de résultats de test (FRT)

La FRT est un document qui permet de conserver les traces de l'exécution d'un cas de test défini par une FDT. Cette fiche est créée et renseignée par le testeur en charge de l'exécution du cas de test.

3.4.2.1. Dénomination du document

Le but est de définir la règle permettant de nommer de manière explicite les fiches de résultats de tests des différents produits logiciels de D2T. L'intérêt qui en découle est une homogénéisation des dénominations : ainsi le testeur est en mesure de nommer aisément les FDT, tout en facilitant leur recherche au niveau de l'archivage.

Règle : Le nom du fichier se construit de la manière suivante :

FRT_Type n° fonction_NomDuLogiciel_N°versionDuLogicielSousTest.doc

Les données importantes à retrouver dans le titre du fichier et systématiquement séparées par un tiret bas sont :

- le **type de document** : fiche de résultats de test (FRT),
- l'**identificateur de la fonction** : correspond au type et au numéro de la fonction testée définis dans les colonnes « Type » et « N° » de l'onglet « gestion des fonctions » du référentiel,
- le **nom du produit logiciel** (Editeur_MORPHEE, exécutif_MORPHEE, Viewer, OSIRIS, TESTMANAGER, CAN, OSIDCOM,...),
- le **numéro de version du produit logiciel** pour lequel la fiche a été créée. Ce numéro est constitué par les 3 numéros qui constituent la version du logiciel en test (majeure + évolution + mineure).



Exemple : FRT_FP6.1.1_VIEWER_V1.0.1.doc

3.4.2.2. Description du document

Le cartouche

Le cartouche (Figure 22) regroupe toutes les informations relatives à l'environnement d'exécution d'un cas de test. Tous les champs du cartouche doivent obligatoirement être renseignés. Lorsqu'il n'y a pas d'information à saisir, le testeur marquera dans le champ concerné l'acronyme « RAS » pour indiquer qu'il n'y a « Rien À Signaler ».

Date d'exécution du test :	10/07/2010	Testeur :	RF
Nom produit :	Guide de conduite	N° de version du produit :	5.0.1.04
Références de l'environnement d'exécution	PC de la zone de test : IO210		
Type OS :	Win XP SP2	Langue de l'OS :	FR
Commentaire	RAS		

Figure 22 : exemple de cartouche de FRT

Le cartouche est constitué des éléments suivants :

- **date d'exécution du test** : date à laquelle le test est exécuté,
- **testeur** : initiales de la personne qui effectue le test,
- **nom produit** : nom du produit sous test,
- **n° de version du produit** : ce numéro est impérativement constitué des 4 numéros qui composent la version du produit sous test, à savoir les numéros de la version majeure, évolution, mineur et le numéro de la bêta. (ex : 1.10.3.4 ou 1.2.9B5),
- **références de l'environnement d'exécution** : dans ce champ sont indiquées toutes les informations relatives aux éléments matériels et/ou logiciels nécessaires à l'exécution du test,
- **type d'OS** : permet de définir l'environnement d'exploitation (ex : Windows XP SP2, Linux RedHat,...) utilisé lors des tests,
- **langue de l'OS** : pour indiquer la langue du système d'exploitation utilisé lors du test,
- **commentaire** : ce champ permet de saisir toute information relative au test et pour laquelle il n'existe pas de champ dans le cartouche.

Le tableau de synthèse d'exécution

Test	Scénario	Commentaire	Résultats Valide /Non Valide
1	Nominal		OK
2	Variante 1		OK
3	Variante 2	Le ticket n°23 a été ouvert dans la base TRAC	NOK

Figure 23 : exemple de tableau de synthèse

Le tableau de synthèse (Figure 23) permet de conserver les traces de l'exécution d'un cas de test défini par une FDT. Ce tableau est composé des 4 colonnes suivantes :

- **numéro de test** : pour identifier aisément le test dans le tableau,
- **scénario** : dans cette colonne sont indiqués les scénarii exécutés,

Améliorer le processus de test pour atteindre le niveau 1 de TPI

- **résultats** : afin de formaliser la réussite (**OK**) ou l'échec (**NOK**) d'un test. Un test est déclaré « réussi » lorsque le résultat obtenu, à l'issue de l'exécution du scénario, est identique au résultat attendu (défini dans la FDT). Le test est déclaré « échoué » dans le cas contraire,
- **commentaire** : cette colonne permet d'apporter des informations sur l'exécution du scénario. En cas d'échec, les informations relatives au constat sont indiquées dans cette colonne.

3.4.2.3. Règle de gestion de la fiche en cas de retest

Lors de l'exécution des tests des scénarii sont amenés à échouer. Lorsque l'échec est dû à une anomalie du logiciel et que la décision est prise de la corriger, il est nécessaire de confirmer la correction en ré-exécutant le cas de test qui a échoué. Cette opération est appelée retest ou test de confirmation.

Un retest est formalisé au niveau de la FRT en dupliquant le cartouche et le tableau de synthèse d'exécution à la fin de la fiche (voir Figure 24). Les 2 tableaux sont renseignés avec les informations relatives au retest.

XX XXX XXX X
 Le 06/07/2010

Fiche de description de test



1^{ère} exécution

Date d'exécution du test :	10/07/2010	Testeur :	RF
Nom produit :	Guide de conduite	N° de version du produit :	5.0.1.04
Références de l'environnement d'exécution	PC de la zone de test : IO210		
Type OS :	Win XP SP2	Langue de l'OS :	FR
Commentaire	RAS		

Retest pour test de confirmation

Test	Scénario	Commentaire	Résultats Valide / Non Valide
1	Nominal		OK
2	Variante 1		OK
3	Variante 2	Le ticket n°23 a été ouvert dans la base TRAC	NOK

Retest pour test de confirmation

Date d'exécution du test :	23/07/2010	Testeur :	RF
Nom produit :	Guide de conduite	N° de version du produit :	5.0.1.06
Références de l'environnement d'exécution	PC de la zone de test : IO210		
Type OS :	Win XP SP2	Langue de l'OS :	FR
Commentaire	RAS		

Test	Scénario	Commentaire	Résultats Valide / Non Valide
1	Nominal		OK
2	Variante 1		OK
3	Variante 2		OK

Figure 24 : exemple de FRT suite à un retest

FRIESS Raphaël

Page 52 / 124

Date : 06 Avril 2012

3.5. Tableau récapitulatif des exigences satisfaites

Nous allons résumer les améliorations qui ont été apportées par le projet en récapitulant les exigences et les sections dans le document qui présentent la manière dont elles ont été satisfaites.

Le Tableau 10 récapitulatif ci-après rappelle les diverses exigences soumises au projet (cf. §2.3 page 21) en vue d'améliorer le processus de test. Les colonnes 1, 2 et 3 présentent les exigences et la colonne 4 renvoie aux sections, de cette partie, présentant les solutions mises en œuvre pour les satisfaire.

ID	Exigence principale	Niveau détaillé	Section
TPI0	Définir une politique de test		§ 3.1
TPI1	Établir et maintenir une stratégie de test pour les tests en aval	Identifier les risques en fonction des exigences	§ 3.3.2.3
		Adapter les tests en fonction du risque	
		Choisir la technique en fonction risque	
		Identifier le type de ré-exécution en cas de test de confirmation	§ 3.4.2.3
TPI2	Établir et maintenir un modèle de cycle de vie des tests :	Rédiger un plan de test	§ 3.3.3
		Établir une description des tests	§ 3.4.1
		Exécution des tests	§ 3.4.2
		Réaliser le bilan des tests	§ 3.3.4
TPI5	Établir des techniques informelles de description de test	Documenter les techniques de construction de cas de test	§ 3.4.1
TPI11	Établir l'engagement et la motivation : Affectation de budget et de temps	Faire considérer les tests comme une chose nécessaire et importante.	
		Affecter aux tests des ressources disponibles et un budget	§ 3.3.3
		Piloter les tests par le budget et les ressources	§ 3.3.3
		Faire monter en compétences les personnes en charge des tests	
TPI15	Établir du reporting des anomalies	Faire des comptes-rendus périodiques	§ 3.1
TPI16	Gérer les anomalies (en internes)	Gérer les étapes de recherche d'anomalies	
		Enregistrer les anomalies	§ 3.3.4
TPI18	Gérer le processus de test		§ 3.1
REQM1	Gérer les exigences	Obtenir une compréhension des exigences	§ 3.3.2.1
REQM2	Gérer les exigences	Obtenir l'engagement sur les exigences	§ 3.3.2.2
REQM3	Gérer les exigences	Gérer les modifications aux exigences	
REQM4	Gérer les exigences	Maintenir la traçabilité bidirectionnelle des exigences	§ 3.3
REQM5	Gérer les exigences	Identifier les incohérences entre les produits du projet et les exigences	
REQM6	Institutionnaliser le processus	Mener des actions pour maintenir le processus dans le temps	

Tableau 10 : Récapitulatif des exigences du projet d'amélioration des processus et leur satisfaction

Conclusion de cette partie du document

Ce chapitre a présenté le nouveau processus de test intégré. Le choix fondamental a été de mettre en place des documents décrivant la politique et le nouveau processus de test ainsi que des modèles de documents servant de support à la mise en œuvre du processus.

Ce nouveau processus de test correspond à un ensemble d'activités (planification des tests, conception des tests, bilan des tests...) intégré tout au long du processus de développement.

La politique de test définit une échelle de niveaux de tests. Elle permet d'avoir une vision commune et cohérente des tests entre les différents projets de développement de logiciels. Elle apporte une aide au chef de projet et au testeur dans la définition d'une stratégie et d'un plan de test.

Le référentiel des exigences et des tests est l'outil central du processus de test. Il a pour principale fonction de permettre de gérer les exigences et leurs tests, ainsi que de fournir une traçabilité bidirectionnelle entre ces deux éléments.

Enfin les fiches de description et fiches de résultats de tests sont des documents modèles permettant de structurer la documentation des tests.

Le Tableau 10 dans le chapitre 3.5 présente la couverture des exigences par les solutions du projet d'amélioration. Il est possible de voir que toutes les exigences n'ont pu être satisfaites. Cela n'empêche pas pour autant à ce nouveau processus d'être exploitable. Les exigences non satisfaites correspondent à celle ayant le niveau de priorité le moins élevé. Les exigences d'amélioration ont successivement été traitées afin de créer le processus de test de manière itérative et incrémentale. La manière d'organiser le projet pour mettre en œuvre ce mode opératoire est décrit dans le chapitre suivant : conduite du projet.

4. Conduite du projet

Objectif de cette partie du document

Cette partie décrit l'organisation et le déroulement du projet. Pour cela, je me suis appuyé sur les pratiques préconisées par le Project Management Body Of Knowledge (PMBOK).

Pour que le lancement du projet soit accordé, il m'a été demandé de décrire la manière dont je comptais mener l'organisation et les moyens mis en œuvre pour garantir que le projet atteindra son objectif. J'ai répondu à cette demande par un plan de management de projet que j'ai présenté à la direction de D2T et détaillé dans le chapitre 4.1.2. L'application de ce plan étant par la suite suivie, de manière régulière, par un comité de pilotage.

Cette organisation est basée sur un ensemble de groupes de travail qui auront à définir de manière itérative et incrémentale le nouveau processus. Ces groupes de travail sont de trois types. Le premier de ces groupes doit définir la politique de test. Il est constitué de personnes provenant de divers secteurs de D2T ayant un lien avec les logiciels développés par le département. Le deuxième type de groupe de travail sont les projets pilotes : ce sont des équipes de développement auxquels il est demandé de définir des bonnes pratiques visant à améliorer le processus de développement pour ensuite le mettre en œuvre au sein même de leur développement. Le dernier type de groupe, les projets de validation, est constitué d'équipes destinées à valider les itérations du processus. La mise en œuvre de ces groupes de travail est présentée dans le chapitre 4.2.

Au début du mois de novembre 2010, la phase réalisation du projet d'amélioration a été arrêtée suite au constat d'une dérive du délai trop importante. L'action en réponse à cette dérive a été une réorganisation du projet. J'ai proposé deux possibilités de réorganisations dont l'une consistait à remplacer les trois derniers projets pilotes restant par un unique groupe de travail. La seconde réorganisation part du principe que le nouveau processus est à un état suffisant pour produire des effets d'amélioration sur la qualité des produits et qu'il est donc possible de le déployer en l'état. C'est cette seconde réorganisation qui a été retenue par le comité de pilotage. Le détail de ces deux propositions ainsi que la justification du choix retenu sont expliquées dans le chapitre 4.3

4.1. Phase planification et lancement

Comme nous l'avons vu dans le chapitre 2, le projet d'amélioration a débuté par une phase d'initialisation pour définir les contours du projet et sa finalité. Le sujet du projet était clair « apporter des améliorations aux tests des produits ». Il n'en reste pas moins que la démarche à adopter et les références dans ce domaine étaient inexistantes dans l'entreprise. Il m'a donc fallu doter le projet d'outils méthodologiques sur le management de projet, ce sera le **PMBOK**, et sur le domaine du logiciel, ce seront les normes et standards suivants :

- **ISO 12207** incontournable dès qu'il s'agit des bonnes pratiques du cycle de vie du logiciel, son développement, sa validation et sa vérification,
- **CMMI** pour l'évaluation des pratiques adoptées (ISO 12207 dit le QUOI faire et CMMI évalue COMMENT le QUOI a été fait),
- **IEEE 830** pour le suivi des exigences,
- **TPI** et **TMap** pour le QUOI faire pour améliorer le processus de test et le comment s'y prendre.

Ensuite j'ai été en mesure de proposer une démarche et un planning. On peut considérer le principe de mise en œuvre suivant. Le chef de projet, en l'occurrence moi, se charge de :

1. faire des propositions en partant des documents de référence normatifs cités ci-dessus,
2. confronter la théorie avec les pratiques internes et avec les opérationnels définir les nouvelles pratiques à implanter,
3. formaliser les nouvelles pratiques dans le référentiel interne,
4. suivre les équipes pour la mise en œuvre des nouvelles pratiques pour vérifier l'applicabilité des pratiques et pérenniser leur mise en œuvre.

À la demande des opérationnels, la vérification de la mise en œuvre a été répartie sur six projets réels appelés ici projets pilotes (PP) chacun appliquant une partie des recommandations. Cette décision d'organisation a eu pour objectif de ne pas faire porter la charge lourde d'implantation et de vérification qu'à un seul projet. Mais ce qui n'avait pas été prévu, on le verra par la suite, c'est que des projets réels n'ont pas vu le jour et que la vérification complète a dû être retardée.

4.1.1. Acquisition des normes et standards du projet

Cette section est consacrée à la présentation des divers standards utilisés dans le cadre du projet. Ainsi le PMBOK est présenté pour la conduite du projet. Le modèle d'amélioration Test Process Improvement (§ 4.1.1.1) est présenté et les raisons pour lesquelles il a été retenu sont justifiées. TMap (§ 4.1.1.3) est la méthode sur laquelle je m'appuie pour guider l'implémentation du nouveau processus de test. En ce qui concerne la mise en œuvre de la gestion des exigences les standards utilisés sont CMMI (§ 4.1.1.4) et IEEE 830 (§ 4.1.1.5). Le premier m'a servi à déterminer les améliorations et évaluer leur niveau de mise en pratique, le second a été utilisé comme guide pour l'implémentation.

4.1.1.1. Le PMBOK comme guide pour conduire le projet

Mon projet débute en même temps que celui chargé de définir le processus de gestion de projet. En attendant que ce dernier soit défini, aucune méthode de gestion de projet n'est imposée. Chaque personne amenée à gérer un projet est libre d'utiliser la méthode de son choix dès lors qu'elle respecte les obligations du Système de Management de la Qualité (SMQ).

Les obligations du SMQ sont de respecter le processus du département Produits, décrit dans le chapitre 2.2.1. Ce processus est orienté développement de système (logiciel, électronique, mécanique). Mon projet est orienté « développement de méthodes de travail ». Il m'est impossible d'appliquer le processus imposé par le SMQ de manière stricte. Il me faut, de fait, soumettre un plan de management du projet à la Direction et à la qualité qui, sur un accord favorable, autorisera le démarrage du projet.

Soumettre l'organisation du projet revient à définir un plan de management du projet, donc une méthode pour définir le plan de management qui permette d'atteindre les objectifs.

Lors de mon cursus au CNAM, j'ai été amené à étudier le Project Management Body of Knowledge (PMBOK). Ce dernier est un guide établi par le Project Management Institute² (PMI) et regroupe l'ensemble des connaissances du domaine du management de projet. Son objectif est de définir de bonnes pratiques dans ce domaine.

Il est stipulé au début du PMBOK que la « notion de bonne pratique ne signifie pas que la connaissance décrite doit être uniformément appliquée à tous les projets. Il appartient à l'équipe de management de projet de déterminer ce qui est approprié pour un projet spécifique » [17]. Je n'applique donc pas le PMBOK de manière strict, mais l'utilise comme un guide m'indiquant la démarche à suivre.

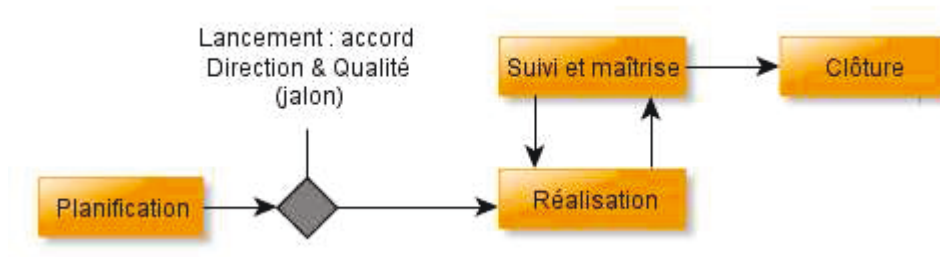


Figure 25 : les phases du projet

La première étape de cette démarche consiste à identifier les phases de mon projet. Ces phases sont présentées dans la Figure 25 :

- **planification** : consiste à définir l'organisation du projet formalisé dans un plan de management,
- **lancement** : est un jalon est non une phase d'où la représentation sous forme de losange. Cette étape est nécessaire afin d'obtenir l'accord en termes de coût, délai, qualité de la part de :
 - **la direction générale** : l'engagement est demandé au niveau direction générale étant donné que le projet découle de la stratégie de l'entreprise et implique des ressources d'autres départements,
 - **la direction qualité** : les livrables du projet impactent les processus de l'entreprise,
- **réalisation** : consiste en la réalisation du projet selon le plan de management,
- **suivi et maîtrise** : consiste à s'assurer que l'exécution se déroule selon la planification et de mener des actions dans le cas contraire,
- **clôture** : consiste à s'assurer que toutes les activités du projet sont finalisées et faire un bilan du déroulement du projet.

Mais avant de décrire de manière détaillée des diverses phases du projet, une description succincte des différents standards utilisés est faite dans la suite de cette partie.

4.1.1.2. Le modèle TPI pour améliorer et évaluer

4.1.1.2.1. Choix d'un modèle d'amélioration

Il existe plusieurs modèles d'amélioration de processus regroupés en deux familles :

- les modèles de référence interne (Systematic Test and Evaluation Process (STEP), Critical Test Process (CTP)) où le référentiel, constitué par les processus de la société sont analysés avant de déterminer les améliorations nécessaires,

² Le Project Management Institute, fondé en 1969, est une association professionnelle à but non lucratif qui propose des méthodes de Gestion de projet. Son siège est à Philadelphie en Pennsylvanie (États-Unis)

- les modèles de référence externe (Test Maturity Model Integration (TMMI), Test Process Improvement (TPI)) où l'on compare les processus de la société à un référentiel idéal.

Mon choix s'est porté sur le modèle TPI pour les raisons suivantes :

- le niveau de connaissance requis (consultant senior expérimenté) pour mettre en œuvre un modèle basé sur le référentiel interne (CTP, STEP,...) n'est pas disponible en interne chez D2T et il n'est pas envisagé (coût financier trop important) de faire intervenir un expert externe,
- TPI présente un schéma de référence complet et surtout très détaillé,
- TPI est associé à la méthode TMap pour Test Management approach [6], développée par SOGETI et présentée dans la section 4.1.1.3, qui permet une mise en œuvre des améliorations,
- les processus actuels (planification de projet, analyse des exigences, ...) sont amenés à évoluer suite à l'adoption d'un standard de conduite de projet (cf. chapitre 1). Ils ne peuvent être pris comme référence (nécessité dans le cas des modèles à référentiel interne) pour définir les améliorations du processus de test sous peine de sortir du périmètre du travail qui m'est demandé.

4.1.1.2.2. Structure du modèle TPI

TPI définit, un processus de test, par vingt **secteurs-clefs** qu'il faut introduire dans une organisation pour faire progresser son niveau de maturité [8]. Un niveau de maturité est obtenu lors de l'atteinte d'un **échelon** particulier (par exemple, 5 pour le niveau « contrôler »). Un échelon correspond à un ensemble de secteurs-clefs mis en œuvre selon un certain **niveau d'approfondissement**, ce dernier étant constitué d'**exigences** à satisfaire (cf. Figure 26). Chacun des éléments de la structure du modèle est expliqué dans cette section.

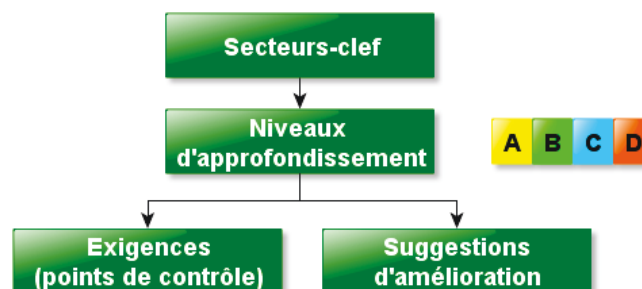


Figure 26 : structure des secteurs-clefs

4.1.1.2.3. Les secteurs-clefs (SC)

Certains domaines du processus de test requièrent une attention particulière ce sont les secteurs-clefs (SC). Ils sont essentiels pour conduire et structurer une démarche de test. Le Tableau 11 contient les noms des vingt SC et une description détaillée de chacun d'entre eux est fournie en annexe 7.1 de ce document.

Secteurs-clefs		
Stratégie de test	Outillage des tests	Reporting
Modèle de cycle de vie	Environnement de test	Gestion des anomalies
Début du processus de test	Environnement de travail	Métriques
Estimation et planification	Engagement et motivation	Communication
Techniques de spécification de test	Gestion du processus de test	Évaluation
Techniques de test statique	Périmètre méthodologique	Tests en amont
Fonctions de test et formation	Gestion des dossiers de test	

Tableau 11: les vingt secteurs-clefs du modèle TPI

4.1.1.2.4. Les exigences

Introduire un secteur-clef dans une organisation consiste à satisfaire un certain nombre d'exigences composant celui-ci. Chaque SC est constitué, comme le montre le Tableau 12 page 60, de deux à vingt trois exigences. Au total, l'ensemble du modèle est constitué de 233 exigences à satisfaire pour atteindre le niveau de maturité le plus élevé.

Par « satisfaire une exigence » il faut comprendre qu'une activité ou une tâche est réalisée et des preuves de leur réalisation sont produites. Par exemple, une exigence dans le cas du secteur-clef 15 (reporting) consiste à rapporter périodiquement les anomalies trouvées par : nombre d'anomalies résolues et non résolues.

Ces exigences sont également utilisées pour évaluer la maturité d'un processus de test. On parlera alors de « points de contrôle ».

4.1.1.2.5. Les niveaux d'approfondissement

Il n'est bien sûr pas envisageable de mettre en œuvre simultanément les 233 exigences. Certaines exigences sont dépendantes des exigences d'autres SC voire du même SC et devant être mise en œuvre préalablement. Elles sont, de faites, organisées sous forme de quatre niveaux d'approfondissement (A ; B ; C et D). Chaque niveau d'approfondissement cherche à atteindre un objectif particulier et pour cela un certain nombre d'exigences lui sont attachées pour un secteur-clef. Par exemple dans le cas du secteur-clef 15 : Reporting (Tableau 12 page 60) l'objectif du niveau d'approfondissement vise à informer :

- **au niveau A**, sur les anomalies détectées,
- **au niveau B**, sur l'avancement (état des tests et des produits), les activités (coûts, délais, jalons), les anomalies avec priorité,
- **au niveau C**, sur les risques et recommandations justifiées par des métriques,
- **au niveau D**, sur les recommandations pour l'amélioration du processus logiciel.

			Niveaux D'approfondissement			
Secteurs-clefs		Total	A	B	C	D
1	Stratégie de test	20	4	4	6	6
2	Modèle de cycle de vie	22	14	8		
3	Début du processus de test	4	1	1	1	1
4	Estimation et planification	4	2	2		
5	Techniques de spécification de test	5	2	3		
6	Techniques de test statique	3	2	1		
7	Métriques	17	8	5	2	2
8	Outils des tests	22	3	6	13	
9	Environnement de test	9	5	3	1	
10	Environnement de travail	2	2			
11	Engagement et motivation	19	6	6	7	
12	Fonctions de test et formation	16	4	8	4	
13	Périmètre méthodologique	8	3	3	2	
14	Communication	12	3	7	2	
15	Reporting	8	1	2	4	1
16	Gestion des anomalies	23	8	13	2	
17	Gestion des dossiers de test	10	3	3	2	2
18	Gestion du processus de test	8	1	4	3	
19	Évaluation	9	4	5		
20	Tests en amont	12	4	3	5	

Nombre d'exigences à satisfaire pour atteindre le niveau d'approfondissement du secteur-clef

Tableau 12 : répartition des exigences par secteurs-clefs et par niveaux d'approfondissement

4.1.1.2.6. Suggestions d'amélioration

Pour aider à la mise en œuvre d'un niveau d'approfondissement, le modèle fournit également des suggestions d'amélioration. Ce sont des idées ou des astuces, leur recours reste néanmoins facultatif. Chaque niveau est doté de plusieurs suggestions d'amélioration.

4.1.1.2.7. Les échelons

À ce stade, il reste encore difficile d'envisager d'introduire les vingt SC simultanément à un niveau d'approfondissement donné. Le modèle définit pour cela des échelons, au nombre de 13, permettant de les introduire de manière ordonnée. Un échelon est un ensemble de SC introduit dans l'organisation selon un niveau d'approfondissement donné. Le niveau d'approfondissement n'est pas forcément le même pour chaque SC. Par exemple, l'échelon 5 (Tableau 13 page 61) implique une introduction des SC :

- « Métriques » à un niveau d'approfondissement A,
- « Engagement et motivation » à un niveau d'approfondissement B,
- « Gestion des anomalies » à un niveau d'approfondissement B,
- « Gestion des dossiers de test » à un niveau d'approfondissement B.

L'ensemble des vingt secteurs-clefs et des treize échelons forment ce que l'on appelle la matrice de maturité des tests (Tableau 13 page 61). Elle a une double utilité. Elle permet dans un premier temps d'évaluer l'échelon auquel se situe le processus de test d'une organisation. Dans un second temps, elle permet de définir les améliorations à y apporter.

Les cases vides (ne contenant pas de lettre) entre les différents échelons n'ont aucune signification en elles-mêmes. Elles indiquent simplement que l'obtention d'une maturité plus élevée pour un secteur-clef dépend de la maturité des autres secteurs-clefs. Par exemple, pour passer de l'échelon 5 à 6, seuls

Améliorer le processus de test pour atteindre le niveau 1 de TPI

les SC 1, 3, 6, 12, 15, 19 et 20 (cf. Tableau 13) nécessitent de satisfaire de nouvelles exigences. Pour tous les autres SC, il faut s'assurer de maintenir les exigences déjà en place.

Secteurs clef \ Echelons	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Stratégie de test		A					B				C		D	
2 Modèle de cycle de vie		A			B									
3 Début du processus de test			A				B				C		D	
4 Estimation et planification				A							B			
5 Techniques de spécification de test		A		B										
6 Techniques de test statique					A		B							
7 Métriques						A			B			C		D
8 Outillage des tests				A				B			C			
9 Environnement de test				A				B						C
10 Environnement de travail				A										
11 Engagement et motivation		A				B						C		
12 Fonctions de test et formation				A			B			C				
13 Périmètre méthodologique					A						B			C
14 Communication			A		B							C		
15 Reporting		A			B		C					D		
16 Gestion des anomalies		A				B		C						
17 Gestion des dossiers de test			A			B				C				D
18 Gestion du processus de test		A		B								C		
19 Évaluation							A			B				
20 Tests en amont					A		B		C					

Tableau 13 : matrice de maturité des tests (source : [8])

4.1.1.2.8. Conditions nécessaires pour atteindre un échelon

Un échelon est considéré comme atteint lorsque :

- tous les secteurs-clefs, pour lesquels un niveau d'approfondissement est indiqué à l'intersection de la colonne de l'échelon et la ligne du SC, satisfont aux exigences de ce niveau d'approfondissement,
- tous les autres SC satisfont les exigences du niveau d'approfondissement qu'ils ont déjà atteint.

Par exemple, les conditions nécessaires pour atteindre :

- l'échelon 1** : les secteurs-clefs 1, 2, 5, 11, 15, 16 et 18 doivent satisfaire aux exigences définies par le niveau d'approfondissement A. Tous les autres SC restant ne nécessitent de satisfaire aucune exigence. Ces SC peuvent ne pas être définis au niveau de l'organisation,
- l'échelon 2** : les secteurs-clefs 3, 14 et 17 doivent satisfaire aux exigences définies par le niveau d'approfondissement A et les conditions de l'échelon 1 sont maintenues.

On constate que la définition du processus de test se fait progressivement par ajout incrémental de SC et/ou de nouvelles exigences pour des SC déjà introduits.

Pour conclure cette présentation du modèle TPI, il est intéressant de souligner ce caractère inclusif des échelons indiqués dans ce paragraphe. Celui-ci est exploité lors de la réalisation d'une évaluation préalable à la détermination des améliorations à réaliser et de l'échelon visé.

4.1.1.3. La méthode TMap pour mettre en œuvre

TPI préconise des exigences d'amélioration sans pour autant indiquer la manière de les implémenter. Afin d'identifier les solutions les plus adaptées à la situation du département Produits, nous avons convenu, le responsable du département et moi-même, d'impliquer le plus possible le personnel du département.

Toutefois, il m'a semblé nécessaire :

- d'éviter des situations de blocage par manque d'expérience du personnel du département sur le sujet,

- d'avoir un guide pour pouvoir orienter la recherche des solutions.

J'ai choisi, pour cela, de m'inspirer de la méthode TMap (pour Test Management Approach – Approche structurée des tests). TMap est la démarche de gestion des tests proposée par la société SOGETI. Elle est en lien direct avec le modèle d'amélioration TPI et propose des solutions adaptables pour chaque exigence du modèle. La méthode TMap propose par exemple un cycle de vie des tests complet, une méthode de détermination de métrique, une description détaillée des rôles et responsabilités des intervenants du test (testeur, analyste, coordinateur,...), des techniques de conception de test,...

Le choix de TMap n'est pas uniquement dû au fait qu'elle est fournie par la société SOGETI en complément de TPI. TMap est actuellement la méthode de référence dans le domaine du test. L'ISTQB la préconise et base ses formations sur la méthode. Les éditeurs d'outils de développement mettent en avant l'aptitude de leurs outils à permettre de respecter la méthode (ex. Visual Studio 2010 de Microsoft). La maîtrise de la méthode est un critère de sélection pour les recruteurs des professionnels du test logiciel. Dernier point qu'il est bon de souligner et qui a eu une influence sur le choix du modèle d'amélioration est le large choix de documents disponibles sous la rubrique de téléchargement du site de la société SOGETI [28] venant en support à la méthode. Tous ces supports sont des outils qu'il est possible d'adapter à sa convenance ou dont il est possible de s'inspirer pour définir ses propres outils de test.

Ces documents sont de type :

- liste pour le contrôle de la conception des tests, du processus de test, de l'environnement de test...,
- modèles Word tel que le plan de test, le script de test, le rapport d'avancement...,
- divers autres documents tel que le choix de la technique de conception selon la caractéristique qualité à tester.

Toutefois, bien que la méthode soit complète et extrêmement bien documentée, elle n'a pas pour vocation d'être mise en œuvre de manière stricte. Elle est utilisée comme un guide à la production d'une méthode personnalisée. La volonté du responsable du département est d'avoir un processus adapté au contexte du département Produits et que les solutions viennent du personnel du département.

4.1.1.4. Le domaine de processus REQM de CMMI pour améliorer et évaluer

4.1.1.4.1. Structure de CMMI

Le Capability Maturity Model Integration (CMMI) est un recueil de bonnes pratiques organisées en vingt-deux domaines de processus répartis en quatre catégories. Chaque domaine de processus est constitué, comme le montre la Figure 27, par des objectifs spécifiques et génériques. À chacun de ces types d'objectifs sont rattachées des pratiques.

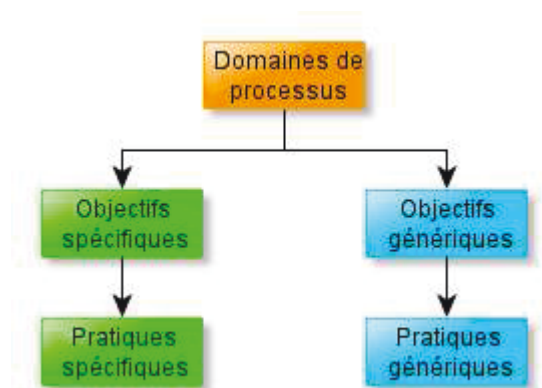


Figure 27 : structure d'un domaine de processus

Les domaines de processus constituent un ensemble de pratiques liées dans un domaine qui, une fois mises en application collectivement, satisfont à un ensemble d'objectifs considérés comme importants pour l'amélioration de ce domaine. Les vingt-deux domaines sont répartis en quatre catégories que sont :

- **la gestion de projet** : cette catégorie contient les domaines de processus qui traitent des activités de planification, de suivi et de contrôle d'un projet,
- **l'ingénierie** : cette catégorie couvre les domaines de processus qui traitent des activités de développement et de maintenance,
- **le support** : cette catégorie regroupe les domaines de processus qui servent dans la mise en œuvre d'autres domaines de processus comme par exemple la gestion de la configuration ou l'assurance qualité,
- **la gestion du processus** : cette catégorie regroupe des domaines de processus qui traitent des activités de définition, de déploiement et d'amélioration des processus.

Les objectifs correspondent aux composants requis qu'une organisation (entreprise, département ou projet) doit réaliser pour satisfaire à un domaine de processus. Cette réalisation doit être mise en œuvre de façon visible dans les processus d'une organisation. Les pratiques expriment une action normalement accomplie dans le cadre d'un processus qui respecte CMMI. Elle décrit un comportement attendu de la part de l'organisation.

CMMI peut être mis en œuvre au sein d'une organisation selon deux approches :

- **la représentation étagée** : on parle de **maturité** de l'organisation. Cette représentation est constituée de 5 niveaux de maturité allant de 1 à 5. Chaque niveau correspond à un ensemble, prédéfini de domaine de processus à introduire au niveau de l'organisation.
- **la représentation continue** : on parle d'**aptitude** de l'organisation. Cette représentation comporte six niveaux allant de 0 à 5. Cette représentation utilise une vision plus ciblée de l'évolution des pratiques. Elle exprime l'aptitude de chaque processus pris séparément au sein de l'organisation.

Concrètement, une organisation a atteint le niveau de maturité 2 lorsque les 7 processus, constituant ce niveau, sont maîtrisés par l'organisation. La mise en œuvre est de type représentation étagée. D'un autre côté, une organisation peut vouloir, par exemple, uniquement maîtriser les processus de « gestion des exigences » et de « planification de projet » en les amenant à un niveau d'aptitude 2. L'organisation mettra alors en œuvre une approche de type représentation continue.

Dans le cadre de mon projet, j'utilise CMMI pour déterminer les pratiques à introduire au niveau du département Produits pour permettre aux équipes projet de développement logiciel d'être en mesure de gérer les exigences (cf. §2.3.3 page 23). Dans cette optique seul le domaine de processus REquirements Management (REQM) est mis sous contrôle. Cela revient, en langage CMMI, à amener le domaine de processus au niveau d'aptitude 2.

4.1.1.4.2. Le processus gestion des exigences (REQM)

Le processus REQM fait partie de la catégorie « ingénierie » constitué de 6 processus présentés dans la Figure 28 ainsi que les liens entre chacun de ces différents processus. REQM est un processus transverse à l'ensemble des processus de cette catégorie.

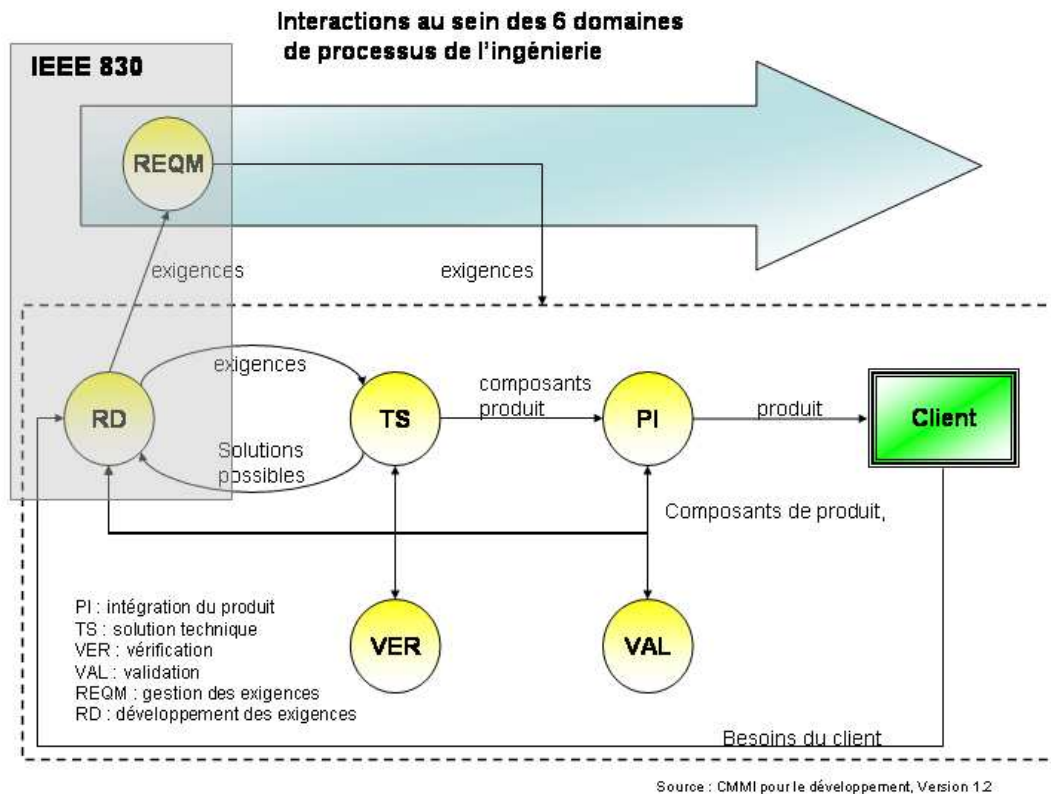


Figure 28 : domaines de processus de l'ingénierie

Selon CMMI [23], l'intention du domaine de processus REQM est de gérer les exigences des produits et composants de produit du projet, et d'identifier les incohérences entre ces exigences et les plans et produits d'activité du projet (un cas de test est un produit d'activité du projet). Les processus de gestion des exigences gèrent toutes les exigences reçues ou générées par le projet, qu'elles soient techniques ou non techniques.

Une partie de la gestion des exigences consiste à documenter les raisons et les modifications, et à maintenir une traçabilité bidirectionnelle entre les exigences source et toutes celles des produits et composants de produits. C'est principalement cet aspect de la gestion des exigences qui est visé, à savoir obtenir une traçabilité entre les exigences du logiciel et les tests qui permettent de les vérifier et les valider.

4.1.1.5. La norme IEEE 830 pour mettre en œuvre

Le modèle CMMI, tout en étant une collection de bonnes pratiques, ne constitue pas pour autant une description de processus. On n'y trouve pas de description quant à la façon et à l'ordre selon lesquels les activités liées à un domaine de processus doivent être exécutées, pas plus d'ailleurs que la liste de ces activités et leur description. Pour mettre en œuvre la gestion des exigences, j'ai choisi de m'appuyer sur la norme IEEE 830.

La norme IEEE-830-1998 Recommended Practice for Software Requirements Specifications décrit les approches recommandées dans la préparation de spécifications de logiciel. Cette norme va donc plus loin que la simple gestion des exigences comme cela est indiqué dans la Figure 28. Elle est également

utilisée dans le cadre de l'amélioration du développement des exigences « Requirements Definition » (RD) dans la terminologie CMMI.

Cette norme vise à obtenir des spécifications bien rédigées, complètes et non ambiguës. Cette partie de développement des exigences est hors périmètre de mon projet. La norme IEEE 830 n'est donc pas mise en œuvre dans son intégralité. Seules les recommandations relatives aux objectifs décrits dans le Tableau 3 page 24 sont utilisées, comme par exemple toutes les recommandations relatives à la traçabilité. A l'instar de TMap, il n'est pas prévu de mettre en œuvre la norme IEEE 830 de manière strict mais de l'utiliser pour éviter les blocages, comme guide pour déterminer des solutions aux exigences de REQM.

4.1.2. Planification

Ce chapitre a pour but de décrire la démarche permettant de définir l'organisation du projet formalisée dans un plan de management. Il reprend la structure du PMBOK sous forme d'un ensemble de domaine de gestion.

La démarche vise, en partant du contenu du projet, à définir la manière dont sont gérés :

- les risques,
- le coût,
- le délai,
- les ressources,
- la qualité,
- la communication.

Un chapitre est dédié à chacun de ces domaines pour en indiquer l'impact sur le projet et la manière dont le domaine est suivi et maîtrisé au cours de la phase d'exécution. Le chapitre 4.1.2.1 est, quant à lui, dédié à la présentation de l'ensemble des personnes impliquées dans le pilotage du projet.

4.1.2.1. Pilotage du projet

Il n'existe pas au sein du département Produits de comité de pilotage. Il m'a toutefois semblé nécessaire, pour le bon déroulement de mon projet d'identifier l'ensemble des managers et décideurs qui sont responsables de la façon dont les collaborateurs travaillent. Ce groupe de personnes doit être impliqué dans les décisions prises tout au long du projet et doit veiller au bon fonctionnement du projet (suivi du projet et validation des choix). Ils peuvent, à ce titre, être considérés comme le comité de pilotage du projet.

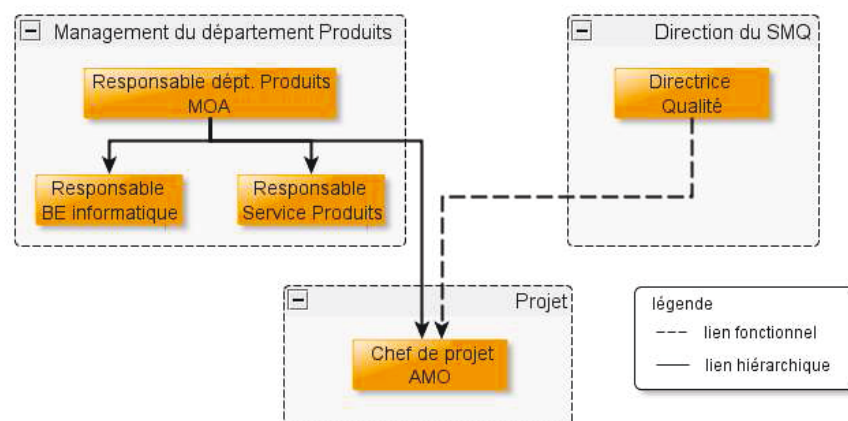


Figure 29 : composition du comité de pilotage du projet

Dans le cas de mon projet, le comité de pilotage (Figure 29) est composé des parties intéressées par le livrable du projet, à savoir :

- **le management du département Produits** : ce sont les trois personnes qui auront à faire appliquer le nouveau processus :
 - le responsable du département : il est la MOA du projet,
 - les responsables du bureau d'études informatique et du service Produits : leurs équipes sont impactées par le livrable du projet,
- **la direction qualité** (représentée par la directrice qualité) : est en charge du SMQ et constitue une partie intéressée dans le sens où le livrable du projet impacte le SMQ de l'entreprise. La mise en application du nouveau processus nécessite son approbation,
- **le chef de projet** (fonction que j'occupe) : a la fonction de AMO et à la charge de planifier, exécuter et suivre le projet d'amélioration.

4.1.2.2. Management du contenu du projet

4.1.2.2.1. Définir le contenu

Le but de cette partie est de s'assurer que le projet comprend tous les travaux nécessaires à sa réalisation et uniquement ces travaux. Pour cela le PMBOK préconise de déterminer la Structure de Découpage du Projet (SDP) (ou en anglais WBS pour Work Breakdown Structure). La SDP a pour but d'aider à organiser le projet, à établir la planification initiale et le budget prévisionnel.

La SDP est une décomposition hiérarchique, axée sur les tâches et activités, du travail que l'équipe de projet doit exécuter pour atteindre les objectifs du projet et produire les livrables voulus (définition du Project Management Institute). La SDP est constituée d'éléments. Chaque élément correspond à une tâche ou à un ensemble de tâches du projet.

Pour déterminer le SDP, je suis parti des exigences soumises au projet (Tableau 10 page 53) et des contraintes (chapitre 2.4) qui lui sont imposées. Les livrables du projet sont :

- un processus de test amélioré (exigences identifiées dans le Tableau 10 par TPI),
- un processus de gestion des exigences (exigences identifiées Tableau 10 par REQ).

Concrètement les livrables se matérialiseront par des documents expliquant les deux processus ainsi que des modèles (Word, Excel,...) et des outils permettant de mettre en œuvre les processus de manière identique par les équipes projets du département Produits (utilisateurs finaux du système produit par mon projet).

4.1.2.2.2. Limites du projet

La norme IEEE 830, couvre en fait deux processus : le processus de « gestion des exigences » et le processus de « définition des exigences ». Les éléments relatifs à la définition des exigences de cette norme sont hors périmètre de mon projet.

4.1.2.3. Structure de découpage du projet (SDP)

Les contraintes (cf. chapitre 2.4.3 : visibilité, facilité d'utilisation, adaptivité et évolutivité) imposées m'ont amené à organiser le projet selon une méthode de développement de type itérative et incrémentale. Ce choix a pour principal avantage de supprimer « l'effet tunnel³ ». Cette méthode vise à segmenter le travail sous forme d'itérations créant chacune un incrément du processus. La planification et l'évaluation du projet sont alors plus faciles car les itérations sont de taille plus aisément gérable. Cette méthode permet aussi l'expérimentation et la validation des processus plus tôt lors de chaque itération.

³ Effet tunnel : laps de temps écoulé entre la formulation des besoins d'utilisateurs concernant un futur système, et l'aboutissement dans sa conception à l'issue duquel ces premiers peuvent évaluer le travail effectué sans avoir eu de droit de regard durant sa période de développement.

J'ai proposé de découper mon projet en une succession de six projets pilotes (PP) et quatre projets de validation (PV) (Figure 30). Le choix d'en faire six vient en réponse à la demande du responsable du département Produits d'impliquer une majeure partie du personnel du département dans le projet.

Un PP et un PV sont des projets de développement d'un produit logiciel. Les projets désignés comme pilote ont comme objectif de définir la méthodologie par identification et capitalisation de bonnes pratiques qui constitueront les nouveaux processus. Chacun des 6 projets pilotes doit réaliser une partie des exigences.

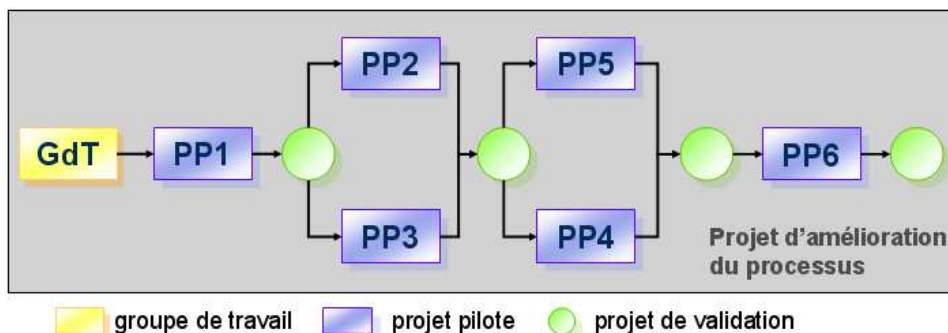


Figure 30 : ordonnancement des diverses phases du projet d'amélioration

La définition de la politique de test (exigence TPI0 - Tableau 10) est fondamentale au nouveau processus de test. Le projet va s'appuyer dessus. Elle doit être définie rapidement et en début de projet. J'ai proposé de la faire définir par un groupe de travail (identifié en jaune dans la Figure 30) au travers de 4 réunions.

Le Tableau 14 présente la structure de découpage du projet sous une forme macroscopique. Celle-ci sera amenée à être affinée par l'ajout d'activités ou tâches nécessaires à la gestion des autres domaines préconisés par le PMBOK.

Activités		Objectif
1 groupe de travail		Définir la politique de test
	4 réunions	Définir la politique de test grâce à du brainstorming
	Rédaction du document	Rédiger la politique de test
6 projets pilotes		Réaliser une itération du processus
	Rédaction des spécifications	Rédiger les spécifications à partir d'un lot d'exigences
	Concevoir et réaliser les solutions	Concevoir et réaliser une itération du processus correspondant aux spécifications
	Rédaction de la documentation	Rédiger les documents d'utilisation du processus
4 projets de validation		Valider une ou plusieurs itérations du processus
	Présenter les objectifs	Présenter le nouveau processus à l'équipe projet en charge de les appliquer
	Appliquer les nouveaux processus	Mise en œuvre du nouveau processus au sein d'un projet de développement
	Évaluer l'application des processus	Évaluer le projet de développement grâce à TPI et CMMI
Gestion du projet		Réaliser le suivi et la maîtrise du projet Réaliser la communication

Tableau 14 : structure de découpage du projet

4.1.2.3.1. Suivi et maîtrise du contenu

L'objectif de cette activité est de suivre et de maîtriser les modifications apportées au contenu du projet. Cette activité est à la charge du chef de projet.

Afin de pouvoir créer les 6 lots d'exigences destinées à chaque projet pilote et également améliorer la visibilité sur l'avancement de mon projet et sur les livrables associés, les exigences du Tableau 10 ont été détaillées en 89 exigences de niveau inférieur. Ces dernières ont été définies sur la base des suggestions d'amélioration du modèle TPI et de la norme IEEE 830. L'ensemble des exigences est visible en annexe 7.3 Matrice de traçabilité des exigences (page 118). Cette matrice de traçabilité est utilisée pour effectuer la gestion des exigences tout au long du projet. Son contenu est validé en début de projet par le comité de pilotage. Toute modification de son contenu (ajout, modification ou retrait d'exigence) nécessite une nouvelle approbation du comité de pilotage.

4.1.2.4. Management du délai

Le projet n'est pas soumis à une contrainte de délai forte. Toutefois pour éviter une dérive du projet lié à l'absence d'objectif temporel, nous avons convenu, (le responsable du département et moi-même) d'une durée d'un an (fin du projet au plus tard Décembre 2010). Cet objectif permet de conditionner des choix lors de la phase de planification et les décisions lors de la phase de réalisation. Dans l'optique de tenir cet objectif, il est nécessaire :

- d'effectuer une estimation de la durée,
- d'élaborer un planning,
- de définir le suivi et la maîtrise de ce planning.

4.1.2.4.1. Estimation des délais

La durée du projet est déterminée à partir de la taille du projet et du séquençement des activités qui le compose.

La taille d'un projet, exprimée en homme/jours, est donnée par la somme des charges nécessaires à sa réalisation [18]. La taille du projet d'amélioration est basée sur la durée nécessaire à la réalisation des 89 exigences. Sans données historiques ni avis d'expert, j'ai fait l'hypothèse que 20 heures par exigences sont nécessaires pour leurs réalisations, ce qui correspond à 237 jours/homme.

Ces 237 jours/homme correspondent à 100% de la charge nécessaire à la réalisation du projet. À cela il faut ajouter du temps nécessaire à la gestion du projet. En me basant sur le vade-mecum du chef de projet [19], j'ai prévu l'équivalent de 25% du temps de réalisation soit 63 jours. Enfin, un temps de 27 jours est également alloué à l'intégration du chef de projet adjoint afin de réduire le risque R2 – indisponibilité du chef de projet (cf. chapitre 4.1.2.6). Cela donne une taille du projet à 327 jours/homme.

Le Tableau 15 présente la répartition du temps alloué par activité (en jours).

Description de l'activité	Durée de l'activité	
	En jours	En %
Définition de la politique de test	28	11%
1 ^{er} projet pilote	32	14%
2 ^{ème} projet pilote	31	14%
3 ^{ème} projet pilote	31	12%
4 ^{ème} projet pilote	28	12%
5 ^{ème} projet pilote	28	12%
6 ^{ème} projet pilote	27	12%
1 ^{er} projet de validation	8	3%
2 ^{ème} projet de validation	8	3%
3 ^{ème} projet de validation	8	3%
4 ^{ème} projet de validation	8	3%
Total	237	100%
Gestion du projet	63	26%
Intégration du chef de projet adjoint	27	11%
Total	327	

Tableau 15 : répartition des charges par activité

4.1.2.4.2. Élaboration d'un planning

La Figure 31 montre les principales phases du projet de manière macroscopique. C'est durant la phase nommée « RÉALISATION » que sont exécutées les activités liées à la définition de la politique de tests, les projets pilotes et de validation. La phase nommée « SUIVI ET MAITRISE » est dédiée à l'activité de gestion de projet.

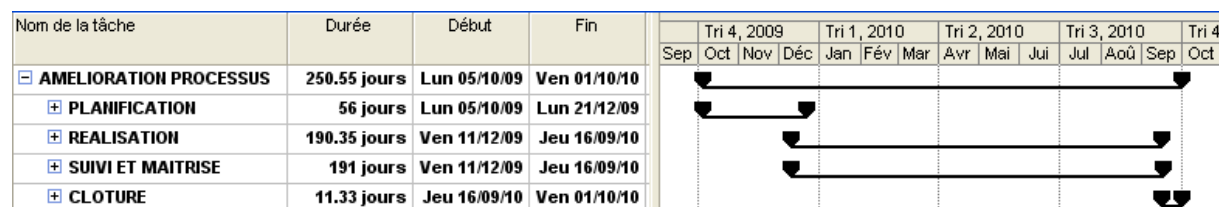


Figure 31 : Planning macroscopique du projet d'amélioration

La principale difficulté pour tenir ce délai réside en l'identification de dix projets de développement nécessaires à la réalisation des projets pilotes et de validation. Ce point est géré en tant que risque projet (chapitre 4.1.2.6 page 70).

Pour limiter ce risque, la liste des projets de développement logiciel prévus sur 2010 a été étudiée. Cette étude a permis de mettre en évidence un panel d'environ 50 projets de développements, sur un total de 70, d'une durée variant d'un à trois mois. Tous les 50 projets ne seront pas lancés en 2010, mais cela permet de valider l'hypothèse qu'il y a suffisamment de projet pour alimenter les PP et PV.

4.1.2.4.3. Maîtrise du planning

Le mode d'organisation du projet par projets pilotes (PP) et de validation (PV) ne permet pas de déterminer avec précision une date de fin. Cette date de fin est fortement dépendante de l'avancement de ces derniers. La fin de la phase « REALISATION », prévue au 16 septembre 2010, correspond à une fin au plus tôt avec une durée moyenne des projets de développement de deux mois.

Pour maîtriser ce délai et tenir une fin au plus tard en décembre 2010, une identification des projets de développement aptes à tenir le rôle de PP ou PV est fait au plus tôt lors des réunions du comité de pilotage du projet d'amélioration.

Le planning est défini grâce à l'outil Microsoft Project. En fin de phase de planification, après l'avoir validé avec le comité de pilotage, le planning est étiqueté comme planning initial permettant le suivi des évolutions. Il est mis à jour au moins une fois par mois par le chef de projet ou son adjoint.

4.1.2.5. Management du coût

Le management des coûts permet de s'assurer que le projet peut être réalisé en respectant le budget approuvé. Pour cela, il est nécessaire :

- d'estimer les coûts,
- de suivre et maîtriser les coûts.

4.1.2.5.1. Estimation des coûts

L'estimation du coût est relativement simple dans le cas de ce projet, puisqu'il n'y a pas d'investissement matériel. Le coût correspond principalement à la somme des coûts horaires des ressources humaines du projet. Le coût horaire est forfaitaire indépendamment du profil de la ressource et se monte à 87,39 €/h (donnée comptable). Le coût total du projet est de 215 K€ sur une base de 7,5 heures par jour.

$$\text{Coût total} = \text{taille du projet (en jrs/h)} \times \text{heures travaillées par jour} \times \text{coût horaire brut}$$

À ce budget est ajouté une provision de 7 K€ en vue des formations CMMI, TPI ou autres si nécessaire.

4.1.2.5.2. Suivi et maîtrise du coût

Suivre et maîtriser les coûts vise à influencer les facteurs générateurs d'écart de coûts et maîtriser les modifications du budget du projet.

Éviter les écarts de coûts, ou du moins les minimiser, revient à s'assurer du respect du temps passé à effectuer une activité par rapport au temps prévu. Un suivi hebdomadaire est réalisé par le chef de projet. Le choix d'effectuer un suivi hebdomadaire vient du fait que le personnel a obligation de pointer leurs heures au minimum une fois par semaine.

4.1.2.6. Management des risques

Le but de ce domaine de la gestion de projet est d'identifier les problèmes pouvant avoir une action défavorable sur le projet et de définir les actions préventives et correctives.

Le management des risques est réalisé au début du projet lors d'une phase d'analyse des risques. Durant cette phase, une identification des risques projet est faite. Ensuite la gravité de chaque risque est évaluée selon la potentialité d'apparition (sur une échelle de 1 à 5) et l'impact sur le projet sur une échelle de 1 à 5). La gravité d'un risque correspond au produit de la potentialité par l'impact.

$$\text{gravité} = \text{potentialité} \times \text{impact}$$

La phase d'analyse est réalisée par le chef de projet. J'ai pris la décision de proposer des actions préventives lorsque la gravité est supérieure ou égale à 10. L'action préventive proposée vise à réduire la gravité en agissant sur la potentialité ou l'impact. Cette analyse est ensuite soumise au responsable du département Produits pour effectuer une révision. En cas d'accord, les actions sont déclenchées, dans le cas contraire l'analyse est réajustée conjointement entre le chef de projet et le responsable du département jusqu'à aboutir à un accord.

4.1.2.6.1. Gestion des risques

Le tableau contenant l'ensemble des risques identifiés et leur évaluation est visible en annexe 7.4 Tableau des risques projet (page 120). Les risques majeurs, pour lesquels une action préventive a été mise en place sont au nombre de trois.

- **Risque R2 - indisponibilité du chef de projet**

Un départ ou une absence prolongée du chef de projet peut entraîner un arrêt du projet. Dans le cas de projets de développement de logiciel, il y a d'autres personnes au sein du département ayant suffisamment de connaissances et compétences du développement pour reprendre le projet en cours. Dans le cas du projet d'amélioration de processus, les connaissances (les modèles d'amélioration et leur mise en application, les pratiques de test,...) ne sont détenues que par le chef de projet. L'indisponibilité du chef de projet entraînerait une interruption voire un arrêt du projet. Pour réduire ce risque, une action de prévention est mise en place en accord avec le responsable du département Produits. Cette action consiste en la désignation d'un chef de projet adjoint dans le but d'effectuer un transfert de connaissances et de compétences. Le coût de l'action est estimé à 17 K€ principalement lié au temps (25 jours) nécessaire au transfert de compétences et à la formation.

- **Risque R1 - manque de compétence en conduite du changement**

Le projet est très orienté méthodologie. Il nécessite de la part du chef de projet des compétences en conduite du changement. Une des activités qui favorise le changement est la communication sur les améliorations que le projet apporte. L'action préventive mise en place vise à aider le chef de projet sur la communication sur le projet. Une personne du marketing est détachée ponctuellement en assistance. Le coût de cette action est estimé à 3,5 K€ correspondant à 5 jours d'intervention pour la mise en place d'un plan de communication.

- **Risque R3 - manque de motivation du groupe chargé de définir la politique de test**

La politique de test est un élément clef du projet d'amélioration (cf. chapitre 2.3.2). La non atteinte de l'objectif par le groupe de travail a un impact sur la suite du projet. Une des principales raisons de la non atteinte de l'objectif est liée au manque de motivation des personnes constituant le groupe. Deux actions limitent ce risque, d'une part par une sélection des personnes participants et d'autre part, une intervention du responsable du département lors de la 1^{ère} réunion afin de sensibiliser les membres du groupe à l'importance de la démarche. Le coût de l'action est estimé à 150€ (1 heure de responsable du département Produits). Cette action sera réitérée durant le projet, car un an c'est long et les gens oublient vite.

4.1.2.6.2. Suivi et maîtrise des risques

Tout au long du projet de nouveaux risques apparaissent ou les risques préalablement identifiés évoluent. Afin de suivre et de maîtriser les risques, j'ai mis en place un tableau de bord (liste de risques gérés à l'aide d'un tableau Excel disponible en annexe 7.4). Ce tableau des risques est mis à jour au moins une fois par mois ou dès l'apparition de nouveaux risques. Lorsqu'un risque atteint une gravité de 10, des actions préventives ou correctives sont soumises au comité de pilotage lors des réunions d'avancement projet dans l'optique de leur mise en œuvre.

4.1.2.7. Management de la qualité

Afin de s'assurer que le projet répond aux besoins définis au départ, un ensemble d'activités sont à mettre en œuvre tout au long du projet pour effectuer l'assurance et le contrôle de la qualité.

L'ensemble de ces activités forme le processus de management de la qualité. Les activités que j'ai proposées pour assurer la qualité sont :

- **établir et maîtriser une matrice de traçabilité** : permet d'identifier les exigences et de tracer les livrables associés au fur et à mesure que ces derniers sont disponibles,
- **réaliser les revues de passage de phase** : cette activité est déjà une obligation du SMQ,
- **vérifier et valider les livrables** : permet de s'assurer que le nouveau processus répond au besoin,
- **gérer en configuration** : permet de gérer tous les produits du projet d'amélioration (documents, outils, ...) et d'identifier leur contenu,
- **gérer les anomalies** : vise à définir la manière de suivre et maîtriser les anomalies détectées durant le projet.

La gestion des anomalies est réalisée grâce à un outil (TRAC) utilisé dans le cadre des projets de développement. Une anomalie, dans le cas du projet d'amélioration, est un problème lié à la mise en œuvre du processus (dysfonctionnement des outils support, défaut dans un document, pratique inapplicable, ...)

Le traitement des anomalies (décision d'une action de correction et assignation à un responsable) sera effectué sous la responsabilité du chef de projet. Un état d'avancement du traitement des anomalies sera effectué au moins une fois par mois et transmis au comité de pilotage.

4.1.2.8. Management des ressources humaines

Le management des ressources humaines vise à définir la composition de l'équipe projet ainsi que le rôle et responsabilités attribués aux personnes de l'équipe.

4.1.2.8.1. Mise à disposition des ressources

Les équipes des projets pilote, de définition de la politique de test, etc. sont constituées pour la majeure partie de personnes formant des groupes de travail. La constitution de ces groupes de travail n'est, au début du projet, connue que sous la forme de profil de personne. La mise à disposition des ressources (identification des personnes physiques) est déterminée au lancement du groupe de travail par le responsable du département Produits en accord avec les responsables de service.

L'équipe projet d'amélioration est, quant à elle, constituée de personnes désignées au lancement et pour toute la durée du projet. Pour autant, les personnes de l'équipe ainsi que le chef de projet ne sont pas affectées à plein temps sur ce projet.

4.1.2.8.2. Composition des équipes, rôles et responsabilités

La composition des diverses équipes est présentée par la Figure 32. Les rôles et responsabilités sont définis dans les Tableau 16 et Tableau 17.

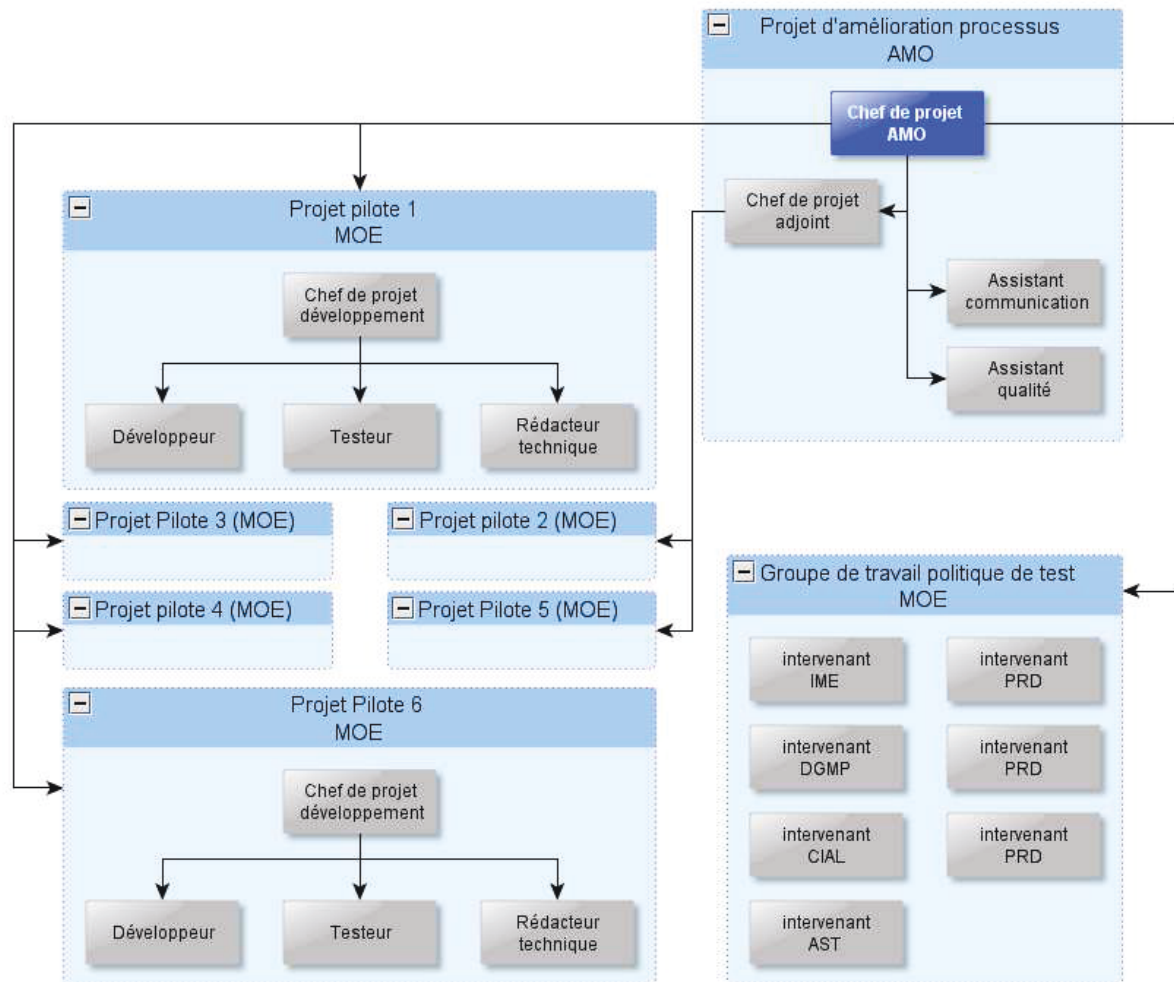


Figure 32 : organigramme des ressources du projet

L'équipe projet d'amélioration de processus

Rôle	Responsabilité
chef de projet AMO	est en charge : - de la planification, - du suivi et de la maîtrise du projet, - de la communication au niveau département Produits comité de pilotage et direction générale, - de la rédaction des spécifications, - de l'organisation et le conseil des - groupes de travail politique de test, - des projets pilote 1 ; 3 et 5, - de l'évaluation des projets de validation, - de l'organisation des revues,

Rôle	Responsabilité
chef de projet adjoint	- remplace le chef de projet sur toutes les activités lorsque ce dernier n'est pas disponible, - est en charge de l'organisation et le conseil des projets pilotes 2 et 4,
Assistant communication	- participe à la définition du plan de communication, - est en charge de la communication du projet au niveau entreprise,
Assistant qualité	- participe à la création et à la formalisation du nouveau processus en vue de son intégration au SMQ, - conseille le chef de projet sur la réalisation des évaluations, - participe à l'évaluation de projet de validation,

Tableau 16 : rôles et responsabilités des ressources permanentes de l'équipe projet

Les besoins en ressources pour le projet sont liés au type d'activité à réaliser :

Équipe	Responsabilité	Rôle / Profil de la ressource
Groupe de travail PdT	en tant que MOE, a la charge de définir la politique de test	metteur en service
		chargé d'essai moteur
		commercial
		responsables produits
		rédacteur technique
		assistant produits
		développeur
Projets pilotes	en tant que MOE, a la charge de définir et mettre en œuvre les solutions répondant aux lots d'exigences correspondant à une itération du processus	chef de projet développement logiciel
		testeur
		rédacteur technique
		développeur

Tableau 17 : rôles et responsabilités des ressources par activité

4.1.2.9. Management de la communication

Un des facteurs impactant le bon déroulement d'un projet est le niveau d'informations des parties prenantes et de l'équipe projet. La collecte des informations et leurs diffusions correspond au management de la communication.

Les personnes visées par la communication sont les collaborateurs du projet d'amélioration (équipe PP et PV compris), le comité de pilotage, la direction générale.

Le projet d'amélioration de processus a également un impact sur la manière de travailler des personnes du département Produits mais également sur la qualité des produits logiciels. Ces derniers sont utilisés à divers niveaux par un grand nombre de collaborateurs de D2T et par les clients de D2T. Toutes ces parties prenantes sont à prendre en compte dans la gestion de la communication.

Améliorer le processus de test pour atteindre le niveau 1 de TPI

Le Tableau 18 regroupe l'ensemble des actions de communication à réaliser au cours du projet d'amélioration de processus.

Destinataire	Type de comm.	Responsable	Périodicité	Type d'information
Comité de pilotage	Reporting projet	Chef de projet AMO	Mensuel	Coût Délai Qualité Problèmes rencontrés
Direction	Lancement	Chef de projet AMO	En début projet	Présentation globale du projet
	Bilan	Chef de projet AMO	A la fin du projet	Présentation du nouveau processus et apport à l'entreprise
Les collaborateurs du département Produits	Conduite du changement	Chef de projet AMO	Au lancement du projet	Présentation et sensibilisation sur les objectifs d'amélioration du processus
		Chef de projet AMO et un participant du groupe	À l'issue du groupe de travail	Présentation de la politique de test
		Chef de projet AMO et un participant du PP	À l'issue de chaque PP	Présentation des améliorations apportées au processus
		Un participant du PV	A l'issue de chaque PV	
Les collaborateurs de D2T	Qualité des produits	Assistant communication	Semestriel	Présentation des objectifs et actions du projet

Tableau 18 : liste des actions de communication

4.2. Réalisation

Ce chapitre est dédié à la présentation de l'organisation des 3 principales activités de la phase de réalisation du projet, à savoir : (1) la définition de la politique de test à l'aide d'un groupe de travail avec production d'un document de mise en oeuvre ; (2) la vérification du nouveau processus et (3) la validation du nouveau processus. Ces deux dernières activités ont été effectuées par des projets existants chacun ayant la charge d'appliquer un sous-ensemble des directives préconisées.

Nous allons donc présenter l'activité initiale de définition puis les activités de vérification et de validation.

4.2.1. L'activité définition de la politique de test

La définition d'une politique de test doit permettre une organisation des tests cohérente au sein de tous les projets de développement. Elle doit être les fondations du nouveau processus. Il faut pour cela qu'elle soit définie avant de débiter les projets pilotes.

Lorsqu'une politique de test existe dans une organisation, elle est généralement définie par la direction. Elle est alors une orientation organisationnelle. Elle peut aussi être définie par des experts (consultant, qualitatif, ...), elle a dans ce cas une orientation opérationnelle. C'est ce second choix qui a été retenu.

N'ayant pas d'expert dans ce domaine en interne et ne pouvant pas faire intervenir d'expert externe (coût financier trop important), j'ai proposé de constituer un groupe de travail pluridisciplinaire composé de personnes provenant du département Produits mais également des collaborateurs D2T liés de par leur activité aux produits logiciels de D2T.

4.2.1.1. Constitution du groupe de travail

La constitution du groupe de travail est réalisée en deux étapes.

La première étape consiste à identifier les profils métier des participants et de justifier l'intérêt de leur participation. Ceci afin d'identifier les personnes physiques sur la base de leur expertise et leurs intérêts à participer à ce type d'activité. J'ai réalisé cette première étape dans le but d'obtenir une première présélection.

La seconde étape, consiste à faire approuver une liste définitive par le comité de pilotage. Cette seconde étape est réalisée au cours d'une réunion qui aboutit au déclenchement de l'activité « définir une politique de test ».

4.2.1.2. Organisation de l'activité

L'activité est réalisée sous la forme de quatre réunions d'une durée de deux heures chacune. Afin de réaliser l'activité dans un délai raisonnable (1 mois prévu initialement), l'ensemble des réunions a été planifié dès le début de l'activité. La Figure 33 présente le planning.

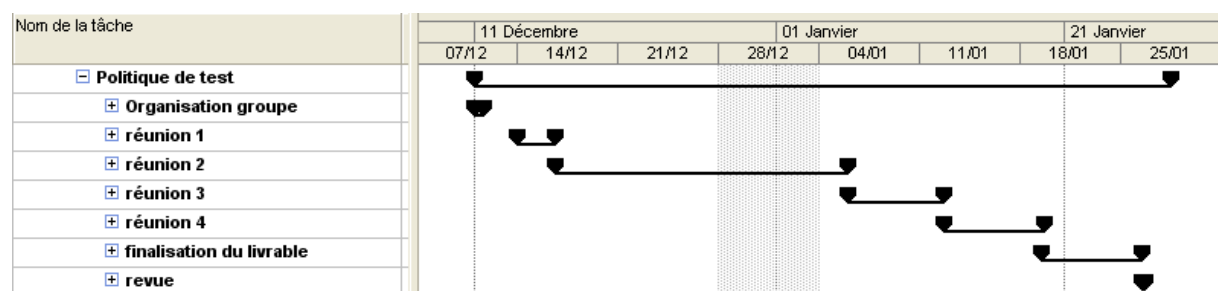


Figure 33 : planning de l'activité "définir une politique de test"

Une réunion est composée de plusieurs tâches :

- **la préparation** : réaliser par le chef de projet et consistant à définir l'organisation de la réunion (ordre du jour, supports, ...),
- **la tenue de la réunion** : d'une durée limitée de deux heures,
- **la rédaction du compte-rendu** : réalisé par un des participants du groupe,
- **la réalisation d'un travail hors réunion** : en vue de la réunion suivante (déclenchée selon l'évolution de la réunion).

Les objectifs de chaque réunion sont :

- **1^{ère} réunion - présentation générale** : expliquer le choix des participants, ce qui est attendu du groupe, l'organisation,
- **2^{ème} réunion** : réalisation d'un brainstorming⁴ afin de faire émerger des propositions,
- **3^{ème} et 4^{ème} réunions** : traitement des propositions issues du brainstorming dans le but de définir les niveaux de test.

Le résultat obtenu à l'issue des quatre réunions est la mise en œuvre d'une échelle à 5 niveaux allant de « très faible » à « très bon ». Chaque niveau définit un ensemble de critères à remplir et d'objectif à atteindre formant les éléments de base d'une stratégie de test. Cette échelle de test est décrite dans sa totalité dans le chapitre 3.2 Le document « Politique de test ».

La rédaction et la mise en forme du document décrivant la politique de test sont réalisées par l'équipe projet d'amélioration (chef de projet et assistant qualité).

L'élément central de la politique de test est le tableau de synthèse (Tableau 6 page 33). Il est utilisé lors de la présentation de cette dernière à l'ensemble des collaborateurs du département Produits.

4.2.1.3. Clôture et bilan

L'activité est clôturée par un bilan présenté au comité de pilotage. Le bilan vise à analyser les écarts entre le prévu et le réalisé, en termes de coût, délai et qualité. Il doit également permettre de capitaliser un retour d'expérience sur le sujet.

Le coût : initialement estimé à 18 k€ s'est élevé à l'issue de l'activité à 15 k€. Ce coût est inférieur à celui estimé, du fait que peu de reprises ont été nécessaires à la suite de la présentation de la première version du document aux membres du groupe et au comité de pilotage.

Le délai : une livraison du document initialement prévue pour mi-janvier 2010, a eu lieu fin janvier. Ce retard est dû à une mauvaise prise en compte des congés de fin d'année.

La qualité : l'objectif est atteint. Une politique de test est définie. Elle contient les éléments nécessaires à la définition d'objectifs de test communs à tous les projets de développement logiciels du département Produits. La politique a été acceptée par le comité de pilotage.

4.2.2. Activité intégration de la politique de test (Accompagnement au changement)

Pour mettre en place la politique de test, il ne suffit pas d'informer les chefs de projet de son existence et de leur imposer son application. Mettre en pratique cette politique est, pour eux, un changement dans leurs habitudes de travail.

Pour les accompagner dans ce changement, il était nécessaire de leur apporter une aide et ainsi faciliter l'acceptation de cette contrainte. Cette aide a été réalisée au travers de deux actions :

- **l'accompagnement à la définition d'une stratégie de test** : au cours des projets pilotes, un membre de l'équipe du projet d'amélioration (moi ou le chef de projet adjoint) intervient en

⁴ Brainstorming : outil de créativité libre et ordonnée qui permet de rechercher en groupe et en toute liberté un maximum d'idées sur un sujet donné ou d'inventer des solutions pour résoudre un problème.

support auprès de l'équipe du projet pilote afin de l'aider à définir la stratégie de test répondant à la politique et ainsi former une majorité du personnel du département Produits à son utilisation,

- **la création d'un guide d'utilisation** : avec le temps le personnel se renouvelle et les effets de la formation s'estompent, j'ai donc proposé de créer un guide d'utilisation de la politique de test.

Dans la suite de cette section sont fournies les explications sur la manière d'utiliser la politique de test pour définir une stratégie de test. Le point de départ de cette démarche est l'identification du niveau de test que le projet vise à atteindre. Ce niveau est imposé par le management du département ou le chef de projet. Dans ce dernier cas, le guide apporte également un aide au chef de projet pour justifier son choix. Ce guide explique les différents éléments qui composent l'échelle de niveau de test (Tableau 6 page 33) afin de mieux l'appréhender.

Pour mémoire, les 5 niveaux de tests, détaillés dans la section 4.2.2.5, vont de « très faible » à « très bon ». Au début du projet, le chef de projet définit le niveau de tests requis lors du développement (si cela n'a pas été fait préalablement). Le choix du niveau de tests lui permet ainsi de définir ses besoins en ressources et l'organisation à mettre en œuvre pour atteindre ce niveau. Un niveau de tests se caractérise par 4 critères chacun détaillé dans une section et qui sont :

- le **référentiel des exigences** disponible (§4.2.2.1),
- la **formalisation des tests** : le formalisme de test à mettre en œuvre (§4.2.2.2),
- la **couverture** : le taux de couverture par niveau de priorité (§4.2.2.3),
- les **testeurs** : correspond aux ressources impliquées (§4.2.2.4).

4.2.2.1. Référentiel des exigences

Cette échelle de niveau de tests se base sur 2 principes sans lesquels la mise en œuvre des niveaux de test est difficilement applicable.

Le premier principe est que la liste des exigences est définie à la fin de la phase de spécification et que chaque exigence est identifiée de façon unique. Cette liste des exigences identifiées est appelée «référentiel des exigences ».

Le second est que les fonctions à réaliser répondent à un besoin et que ces fonctions sont exprimées sous forme d'exigences. La description des exigences peut se faire par 2 types de description :

Description pauvre : la description de l'exigence est succincte. Par exemple : corriger l'anomalie n°2458.

Description riche : La description de l'exigence remplit 6 critères, à savoir :

- clairement énoncée,
- correctement énoncée,
- complète,
- non contradictoire,
- approprié à mettre en œuvre,
- testable.

4.2.2.2. La formalisation des types de test

Dans cette partie sont définies les diverses manières de formaliser un type de test. L'objectif étant de définir sous quelle forme sont conservées les traces de la réalisation des tests.

Le Tableau 19 présente les trois types de formalisation pouvant être utilisés.

Formalisme	Description
informel	Le test est effectué avec pour seule trace de l'exécution l'information sur la réussite ou l'échec du test. Aucune trace sur la manière dont le test a été réalisé, ni les résultats qui ont été obtenus à l'issue du test. Cette manière de formaliser un test ne peut être mis en œuvre que sur des exigences de faible priorité pour le produit : • en cas de manque de temps pour mettre en œuvre un formalisme de niveau plus élevé (semi formel ou formel), • par un testeur qui a une expertise du produit à tester (ex. le responsable produits).
semi formel	Lors de la réalisation d'un test, une trace des résultats d'exécution est conservée. Des informations sur la manière dont ont été réalisés les tests peuvent faire partie des traces du test. Les traces ne permettent toutefois pas une ré-exécution du test à l'identique lors d'une campagne de test ultérieure ou par une personne autre que celle qui l'a effectué la première fois. Cette manière de formaliser un test doit être considéré, par toute personne amenée à faire du test, comme étant le minimum requis pour tester une exigence.
formel	Dans ce type de formalisation, lors de l'exécution du test ou préalablement à l'exécution, un script de test est créé. Les résultats d'exécution du script sont utilisés comme trace de test. L'objectif dans ce cas est de pouvoir exécuter ultérieurement le test de manière identique, suite à l'évolution du produit et/ou par une personne ayant peu ou pas d'expertise du produit à tester. Cette manière de formaliser un test doit être mise en œuvre : • dans le cas de test d'exigences critiques du produit, • par une personne ayant une bonne expertise du produit (lors de la création du script). Un script de test est un document qui décrit : • l'état initial du système testé, • la séquence d'opérations à effectuer, • l'état final, après exécution de la séquence d'opérations, • le ou les jeux de données du test (ensemble de valeurs pour chacune des variables mises en œuvre lors de la séquence d'opération). Les informations fournies doivent être suffisamment claires et précises afin de permettre à une personne non experte du système testé d'exécuter le script.

Tableau 19 : les divers types de formalisation de test

Les types de tests, décrits dans le Tableau 20, sont ceux qui sont les plus communément utilisés lors du test des produits logiciel D2T. L'objectif recherché est de définir les types prioritaires à mettre en œuvre dans une campagne de test. La définition de ces types de test ne limite en rien l'utilisation d'autres types, en complément, qui se justifieraient dans un contexte particulier.

Dans le cas où d'autres types de test, que ceux décrits dans le Tableau 20, sont utilisés, ceux-ci sont définis au niveau du plan de test.

Type de test	Objectif
capacité fonctionnelle	Ce test a pour but de contrôler qu'une nouvelle exigence a été réalisée conformément aux demandes faites dans la phase de spécification.
non régression	Le but de ce test est de contrôler, après modification d'un logiciel, qu'une exigence, préalablement testée lors d'une version antérieure, a encore le comportement tel qu'il a été demandé à l'origine. Ces tests sont effectués quand le logiciel ou son environnement est modifié.
compatibilité ascendante	La compatibilité ascendante est la faculté, pour un logiciel, d'être compatible avec une de ses anciennes versions, ou une ancienne version des technologies qu'il emploie. Ce test a pour but de contrôler qu'une exigence / fonction / un logiciel, après modification, est encore en mesure de traiter des données ou des technologies qu'il était possible de gérer avec l'ancienne version.
documentation	Ce test a pour but de s'assurer que la documentation est cohérente avec le fonctionnement du logiciel.
recette	Le test de recette est un test fonctionnel réalisé en présence du client pour prouver que le logiciel répond aux objectifs contractuels définis à l'origine du projet.

Tableau 20 : les types de tests communément utilisés chez D2T

4.2.2.3. Couverture des exigences

La couverture des exigences définit un objectif à atteindre en termes de nombre d'exigences contrôlées par des tests. Les exigences sont classées par priorité (avec au minimum 2 types de priorité : élevée et basse). Un objectif de couverture est défini pour chaque type de priorité. Cet objectif de couverture se nomme taux de couverture des exigences par les tests et est exprimé en pourcentage.

Taux de couverture : rapport entre le nombre d'exigences contrôlées par un ou plusieurs tests et l'ensemble des exigences du projet.

$$\text{taux_de_couverture}(\%) = \frac{\sum \text{exigences_testées}}{\sum \text{exigences}}$$

4.2.2.4. Les ressources

La colonne « Testeur » indique les ressources impliquées dans la réalisation des tests. Elles sont de trois types :

- **BEI** : les testeurs de ce groupe sont des personnes du Bureau d'Étude Informatique (développeur, architecte, ...)
- **PRD** : les testeurs de ce groupe sont toutes des personnes issues du service Produits (responsable produits, assistant produits, chef de projet logiciel, rédacteur technique...).
- **Client** : les testeurs de type client peuvent être internes (metteur en service, opérateur DGMP, chargé d'essai DGMP, ...) ou externes.

4.2.2.5. Les niveaux de tests

Le niveau est défini par un terme (très faible, faible, moyen, bon, très bon) qui exprime une qualité des tests visée et par la suite obtenue.

4.2.2.5.1. Très faible

Ce niveau de tests peut être requis lorsque le test n'est pas techniquement réalisable en interne (matériel ou environnement uniquement disponible chez le client).

Référentiel des exigences

Le référentiel des exigences est pauvre. Ce cas peut se présenter lorsqu'une anomalie bloquante est identifiée nécessitant une correction urgente et que le référentiel des exigences n'a pas été créé lors du développement des versions précédentes du produit.

Formalisation des tests

Le produit est livré au client, sous dérogation, sans test préalable ou peu de tests informels en interne.

Couverture

Priorité	élevée	basse
Taux de couverture	≥ 0% des exigences	≥ 0% des exigences

Ressources

Lorsque des tests sont réalisés en interne, ils sont pris en charge par le Bureau d'Étude Informatique. Dans le cas de tests externes le client en a la charge.

4.2.2.5.2. Faible

Ce niveau de tests peut être requis lorsque le délai de réalisation dédié aux tests est trop court pour les réaliser de manière plus formalisée.

Référentiel des exigences

Le référentiel des exigences est pauvre. Le référentiel des exigences des versions antérieures du produit peut ne pas exister ou est d'un formalisme pauvre. Une documentation utilisateur de la version précédente existe. En l'absence de spécification de la version antérieure la documentation utilisateur peut être considérée comme palliatif à l'absence de spécification et est utilisé pour la création du référentiel des exigences.

Formalisation des tests

Les tests sont réalisés de manière informelle. Le but principal est de vérifier que les nouvelles exigences réalisées sont opérationnelles et correspondent aux attentes du client (interne ou externe). Certaines exigences peuvent ne pas être testées sur demande du chef de projet.

Couverture

Priorité	élevée	basse
Niveau de couverture	≥ 50% des exigences	≥ 0% des exigences

Ressources

Les tests sont réalisés par des ressources du Bureau d'Étude Informatique et/ou du service Produits ou client lorsque l'exigence ne peut être testée en interne.

4.2.2.5.3. Moyen

Ce niveau correspond au niveau minimum attendu en situation normale lors d'un développement de produit. Tout produit déclaré comme stratégique doit obtenir au minimum ce niveau de tests.

Référentiel des exigences

Le référentiel des exigences est riche. Une documentation utilisateur de la version précédente existe. La spécification est maintenue à jour tout au long du développement.

Formalisation des tests

Les tests sont réalisés de manière semi formelle. Le but est de vérifier que les exigences sont opérationnelles et correspondent aux attentes du client (interne ou externe). L'objectif est de couvrir un maximum des exigences réalisées (anciennes et nouvelles) pour avoir confiance en la version courante. Pour atteindre l'objectif de couverture des exigences, certaines exigences peuvent être testées de manière informelle sur accord du chef de projet voire ne pas être testées, sur demande du chef de projet, dans le cas des exigences de priorité basse.

Couverture

Priorité	élevée	basse
Niveau de couverture	≥ 80% des exigences	≥ 0% des exigences

Ressources

Les tests sont réalisés par des ressources du Bureau d'Étude Informatique et/ou du service Produits ou client lorsque l'exigence ne peut être testée en interne.

4.2.2.5.4. Bon

Ce niveau est requis pour tout produit désigné comme stratégique et ayant atteint le niveau de tests moyen lors de la précédente version.

Référentiel des exigences

Le référentiel des exigences est riche. Le niveau de priorité des exigences est défini dans le référentiel des exigences. Une documentation utilisateur de la version précédente existe. La spécification est maintenue à jour tout au long du développement.

Formalisation des tests

La majorité des exigences est testée et réalisée de manière semi formelle. Certaines exigences sont testées de manière formelle. Le choix de ces dernières est du ressort du chef de projet et correspond à des exigences critiques du produit. Certaines exigences de priorité basse peuvent être testées de manière informelle sur accord du chef de projet ou ne pas être testées sur demande du chef de projet.

Couverture

Priorité	élevée	basse
Niveau de couverture	= 100% des exigences	≥ 50% des exigences

Ressources

Les tests sont réalisés par des ressources du Bureau d'Étude Informatique et/ou du service Produits ou client lorsque l'exigence ne peut être testée en interne.

4.2.2.5.5. Très bon

L'objectif de ce niveau est de réduire les coûts d'exécution des tests formels. Le niveau de tests de la version précédente du produit est déjà de niveau bon ou supérieur.

Référentiel des exigences

Le référentiel des exigences est riche. Le niveau de priorité des exigences est défini dans le référentiel des exigences. Une documentation utilisateur de la version précédente existe. La spécification est maintenue à jour tout au long du développement.

Formalisation des tests

L'objectif est de couvrir la totalité des exigences de priorité élevée par des tests formels. Cet objectif est défini en début de projet. Certains tests réalisés de manière formelle sont automatisés. Sur accord

du chef de projet, certaines exigences à priorité élevée peuvent être testées par des tests semi formelles. Les exigences de priorité basse sont majoritairement couvertes par des tests semi formels ou informels (sur accord du chef de projet). Certaines exigences de priorité basse peuvent ne pas être testées sur demande du chef de projet.

Couverture

Priorité	élevée	basse
Niveau de couverture	= 100% des exigences	≥ 80% des exigences

Ressources

Les tests sont réalisés par des ressources du Bureau d'Étude Informatique et/ou du service Produits ou client lorsque l'exigence ne peut être testée en interne.

4.2.3. Définition du nouveau processus

La définition du nouveau processus est réalisée grâce à des Projets Pilotes (PP). Ce sont des projets de développement de logiciels désignés pour identifier des pratiques et des outils qui constituent la partie opérationnelle du nouveau processus. Chaque PP se voit soumettre une partie des exigences et doit proposer des solutions puis démontrer leur capacité à être mise en œuvre. Pour pouvoir réaliser cela, il est nécessaire de répartir l'ensemble des exigences par lots. Ce lotissement est réalisé sous la responsabilité du chef de projet et validé par le comité de pilotage.

4.2.3.1. Lotissement et objectif des lots

La création de lots d'exigences est réalisée en se basant sur le fait que certaines exigences doivent être satisfaites avant d'autres. Par exemple, pour pouvoir adapter les tests aux risques produits il faut au préalable avoir déterminé les risques produits. Il est donc nécessaire de mettre en œuvre la détermination des risques dans un lot précédent celui mettant en œuvre l'adaptation des tests aux risques. De plus, étant donné que certains PP se déroulent en parallèle (PP2/PP3 et PP4/PP5) il est nécessaire de prendre en compte le niveau d'interaction entre les exigences.

J'ai défini six lots d'exigences. Le résultat de la répartition des exigences issue du travail de lotissement est présenté de manière synthétique dans le Tableau 21 et de manière détaillée dans la matrice de traçabilité (Annexe 7.3). Le Tableau 21 permet de constater que la répartition des exigences par lots n'est pas uniforme. Cette répartition vise à atteindre un objectif. Chaque PP a ainsi un objectif précis à atteindre. Par exemple, le PP2 doit mettre en place des pratiques visant à gérer les risques. J'ai, de fait, assigné au lot du PP2 toutes les exigences liées aux risques tels que : TPI1.1 - identifier les risques en fonction des exigences, REQM1.1.3.5 - définir une priorité selon les risques... Chacun de ces objectifs est décrit dans la suite de cette section.

Projet pilote	PP1	PP2	PP3	PP4	PP5	PP6	Total
Nombres d'exigences à réaliser	23	8	14	13	20	11	89

Tableau 21 : répartition des exigences par projet pilote

Objectif de PP1 : initier la démarche

Poser les bases des processus de test et de gestion des exigences. Concrètement cela se traduit par l'obtention d'une méthodologie qui vise à aider les futurs projets de développement à être en mesure d'établir:

- la liste des exigences à réaliser,
- la liste des tests à réaliser,
- le niveau de formalisme mis en œuvre pour chaque test,
- les liens entre les exigences et le test.

Objectif de PP2 : gérer les risques

La gestion des risques se traduit par l'obtention d'une méthodologie qui vise à aider les futurs projets de développement pour :

- définir les types de risques possibles,
- prioriser les fonctions à réaliser suivant les risques identifiés,
- définir une stratégie de tests à réaliser basée sur les risques,
- établir les liens entre les risques des exigences et les risques au niveau des tests.

Objectif de PP3 : gérer la criticité et le besoin client et faire montée en compétence

L'objectif de ce PP est d'une part, de définir une méthodologie permettant de prioriser les fonctions selon une vision client et d'autre part, débiter la montée en compétence dans le domaine du test des personnes amenées à faire du test.

Cela se traduit concrètement par :

- une évolution de la ou des méthodes et outils mis au point lors du PP1 afin de prendre en compte la provenance des exigences. Les exigences d'un logiciel ne proviennent pas toutes du même client (par exemple elles peuvent provenir d'un client externe et interne (marketing, direction,...),
- une montée en compétence en matière de méthode de test avec des formations sur les techniques de test.

Objectif de PP4 : gérer les modifications

Les exigences soumises à un projet sont amenées à évoluer au cours de ce projet (ajout de nouvelles exigences, annulation ou report d'exigences initialement prévues, etc.). Ces modifications ont un impact sur le travail des divers intervenants du projet et doivent, de fait, être gérées. Cet objectif se traduit principalement par la mise en place de moyens permettant de :

- documenter les modifications,
- maintenir la traçabilité entre les exigences et les éléments qui leurs sont liés.

Objectif de PP5 : piloter le projet par les exigences et les tests

L'adaptation du pilotage projet se traduit par l'obtention d'une méthodologie qui vise à aider les futurs projets de développement pour :

- planifier, piloter et contrôler les processus de test et de gestion des exigences,
- rédiger un plan de test,
- contrôler la compréhension et l'engagement des participants sur les exigences,
- adapter le projet et sa mise en place suivant le budget et les ressources disponibles,
- faire un bilan et un retour d'expérience sur les exigences et les tests réalisés.

Objectif de PP6 : reproduire

À ce stade le nouveau processus est défini. Ce dernier PP vise à préparer au déploiement du processus. Les objectifs du PP6 sont donc de :

- pérenniser les deux processus,
- gérer la ré-exécution des tests,
- gérer l'infrastructure.

Ce projet pilote a la particularité de nécessiter la participation du management du département Produits.

4.2.3.2. Organisation

Les six projets pilotes sont organisés de manière identique. Cette organisation se décompose en quatre phases dont l'ordonnancement est présenté par un extrait du planning (Figure 34). Ces phases sont :

- **la spécification** : consiste à regrouper l'ensemble des exigences soumises au PP dans un document,
- **la réalisation** : consiste en l'identification de solutions et leur mise en pratique pour s'assurer de leur faisabilité,
- **la documentation** : consiste à rédiger la documentation relative aux solutions retenues,
- **la clôture** : consiste à faire le bilan du PP.

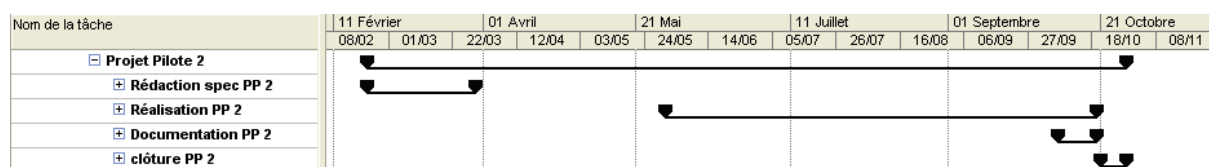


Figure 34 : planning du projet pilote 2

4.2.3.3. Spécification

La phase de spécification a pour but d'extraire les exigences, relatives au PP concerné, de la matrice de traçabilité et de les regrouper dans un document. La spécification permet ainsi aux personnes intervenant dans le PP d'identifier le périmètre et les objectifs sur lesquels ils ont à travailler.

Les spécifications sont rédigées par le chef du projet d'amélioration et le chef de projet adjoint. Elles sont rédigées le plus tôt possible pour qu'elles soient validées lors d'une revue. La révision des spécifications est faite en deux étapes :

- la première étape vise à faire approuver le document par le comité de pilotage. Dès lors que le document est approuvé, le travail de recherche d'un projet de développement apte à servir de pilote peut commencer,
- La seconde étape consiste à faire vérifier le document par l'équipe PP afin de s'assurer que son contenu est compris et qu'ils acceptent de travailler sur le sujet. Leur accord sur la spécification permet d'avoir un engagement de leur part et de s'assurer de leur motivation, indispensable à l'atteinte de l'objectif.

Le choix du projet retenu comme pilote est réalisé selon plusieurs critères tel que :

- **le nombre de personnes constituant l'équipe projet** : au minimum 3 personnes,
- **la taille du projet** : entre 2 et 6 hommes/mois,
- **le type de développement** : un produit logiciel.

Dès lors que le projet pilote est désigné, la phase de réalisation peut commencer.

4.2.3.4. Réalisation

La réalisation consiste à définir des pratiques et des outils répondant aux exigences soumises au PP. Il faut pour cela, dans un premier temps, identifier des solutions et choisir la plus adaptée pour ensuite s'assurer qu'elle est réellement adaptée.

L'identification des solutions et le choix retenu sont réalisés aux cours de réunions de type brainstorming selon le même schéma organisationnel que pour la définition de la PdT (cf. 4.2.1.2). Les réunions, d'une durée de deux heures et en moyenne au nombre de six, aboutissent à la rédaction d'un document décrivant les solutions retenues.

Cette étape peut être vue comme une phase de conception lors de laquelle l'évolution du processus de test et gestion des exigences est définie. Les nouveaux composants ajoutés au processus sont des pratiques et des outils pour le test ou la gestion des exigences. Par exemple, dans le cas du PP3 une des exigences soumises au PP était de définir une pratique permettant de documenter la technique de construction des cas de test (cf. annexe 7.3 – exigence TPI5.1). La conception a abouti à une proposition de maquette d'une fiche de description de test.

L'évolution issue de la phase de conception est soumise au comité de pilotage pour vérification et validation. Le comité de pilotage s'assure que cette évolution est compatible avec les travaux réalisés par le projet en charge de l'amélioration du processus de gestion de projet et plus généralement avec le SMQ.

4.2.3.5. Vérification

Dès lors que l'évolution est approuvée par le comité de pilotage, la phase de vérification peut débuter. Elle vise à s'assurer que l'évolution est adaptée, en la mettant en œuvre au cours du développement du logiciel ou de l'ajuster dans le cas contraire. Pour reprendre l'exemple précédent, l'équipe du PP rédige des fiches de description de test afin de finaliser le modèle.

4.2.3.6. Documentation

La phase de documentation est réalisée en parallèle de la réalisation. Elle est faite par l'assistant qualité et le chef de projet ou le chef de projet adjoint.

La fiche de description du processus est reprise pour intégrer les évolutions, les documents et modèles sont mis en conformité avec le standard de la société. Des modes d'emploi sont rédigés à destination des futurs utilisateurs.

La fin de phase de documentation et de réalisation se termine conjointement par une nouvelle présentation des processus au comité de pilotage pour approbation.

4.2.3.7. Clôture

À l'issue des phases de réalisation et documentation, le chef de projet réalise le bilan du projet pilote en vue de sa clôture. Le bilan porte sur l'atteinte des objectifs de coût, délai et qualité et l'analyse des écarts. Cette section présente le bilan des trois premiers projets pilotes et illustre celui-ci de manière chiffrée par le Tableau 22.

	durée (en mois)		date de fin	coût (en k€)		écart (en %)	
	prévue	réelle	réelle	prévu	réel	délai	Coût
PP1	4.3	5.1	01/07/2010	20.97	26.96	18%	29%
PP2	5.6	8.3	29/10/2010	20.32	16.27	47%	-20%
PP3	5.6	8.2	29/10/2010	20.32	23.97	45%	18%

Tableau 22 : bilan coût et délai des projets pilotes 1 à 3

Afin de réaliser le projet d'amélioration sur une période d'un an, la durée des projets pilotes a été estimée de 4 à 5 mois. Un recouvrement entre la fin du PP précédent et le début du suivant devant permettre d'atteindre cet objectif. À l'issue des trois premiers PP, il s'est avéré que cette hypothèse ne se vérifie pas. J'ai alerté très tôt le comité de pilotage d'un risque important de dérive du délai. La décision a été prise de privilégier le respect du coût et de la qualité au détriment du délai.

Le coût total à l'issue des trois premiers projets pilotes est supérieur de 9% au coût estimé. L'écart est pour la majeure partie imputable à la gestion des projets (des retards sont également imputables à des reprises de conception lors de la phase de réalisation). Le coût de cette dernière est plus important du fait d'une durée des projets plus longue, augmentant implicitement la charge de ce poste. Cette dérive du coût a entraîné la décision de réorganiser le projet. Le détail de la réorganisation du projet est présenté dans le chapitre 4.3.

Améliorer le processus de test pour atteindre le niveau 1 de TPI

En termes de qualité, la couverture des exigences par la réalisation est de 98% avec les exigences TPI16.1 annulée et REQM1.1 reportée. Cette couverture doit toutefois être confirmée par les projets de validation.

4.2.4. La vérification

La vérification vise à s'assurer que le produit du projet a été bien fait [3]. Pour ce faire, chaque équipe projet pilote effectue la vérification de la partie (les outils et leur documentation) qu'elle a implémentée (cf. § 4.2.3.5). L'équipe projet pilote doit ainsi démontrer, par une mise en application au sein même de leur projet de développement, que les solutions qu'elle a proposées sont efficaces.

Toutefois ce niveau de vérification peut être considéré, dans le domaine du test, comme du test unitaire, puisqu'il ne contrôle que ce que le projet pilote a implémenté. Il est également nécessaire de réaliser de la vérification plus globale. En effet, à l'issue de chaque projet pilote nous obtenons une nouvelle version du processus de test. Il faut ainsi également s'assurer que chaque version du processus est bien faite. Par « bien faite » il faut comprendre qu'il me faut démontrer que la version est conforme à l'attendu.

Pour ce faire, j'ai créé un outil, la **matrice de traçabilité**, afin d'identifier ce que chaque version du processus doit contenir en termes d'exigences et de tracer l'élément (une partie d'un outil par exemple) qui répond à cette exigence. J'ai également défini une **méthode pour gérer en configuration** les versions du processus. Avec cette matrice et cette méthode, j'ai été en mesure de réaliser la vérification et de démontrer de manière formelle l'atteinte des objectifs au cours des **revues projet**.

4.2.4.1. La matrice de traçabilité

Pour effectuer le suivi et être en mesure de maîtriser l'ensemble de ces 89 exigences, j'ai créé une matrice de traçabilité sous Excel présentée partiellement à la Figure 35 et est consultable dans sa totalité en annexe 7.3 page 118.

1	2	3	4	A	C	F	I	J	N	R	V	Z	AD
11	Identificateur	Type	n°	Exigences	Résultat livré	pp1	pp2	pp3	pp4	pp5	pp6		
14	TPI	1		Établir et maintenir une stratégie de test		1	1	1	1				
15	TPI	1		Établir et maintenir une stratégie pour les tests en aval		1	1	1	1	1	1		
22	TPI	2		Établir et maintenir un modèle de cycle de vie		1			1	1	1	1	
23	TPI	2		Planification, Description, Exécution		1			1	1	1	1	
24	TPI	2.1		Rédiger un plan de test		1					1	1	
25											1		
26	TPI	2.1.1		Définir les critères qui conditionnent le lancement des tests	chapitre 6.1 dans le plan de test et documenté dans le	1							
27	TPI	2.1.2		Déterminer le référentiel de tests	Chapitre 4 du plan de test	1							
28	TPI	2.1.2		Établir la liste des tests	Colonne B du référentiel	1							
29	TPI	2.1.2		Identifier de manière unique	Colonne A du référentiel	1							
30	TPI	2.1.2		Traçable	doc_processus.doc	1							
31	TPI	2.1.3		Déterminer la stratégie de test	Définir le niveau de test attendu	1							
32	TPI	2.1.4		Mettre en place l'organisation							1		
33	TPI	2.1.5		Mettre en place les livrables de test								1	
34	TPI	2.1.6		Définir l'infrastructure et les outils									1
35	TPI	2.1.7		Mettre en place le management							1		
36	TPI	2.1.8		Planifier	modèle_Plan_test_projet_v0.1.doc	1							
37	TPI	2.3		Établir une description des tests					1				
42	TPI	2.4		Exécution des tests		1						1	
47	TPI	2.5		Établir un bilan des tests		1					1	1	
48	TPI	2.5.1		Procéder à une évaluation du logiciel testé							1		
49	TPI	2.5.2		Procéder à une évaluation du processus de tests							1		
50	TPI	2.5.3		Formuler des propositions									1
51	TPI	2.5.4		Rendre compte	modèle_Bilan_test_projet_v0.2.doc	1							
52	TPI	5		Établir des techniques de spécification de test				1					
53	TPI	5		Techniques informelles				1					
54	TPI	5.1		Documenter les techniques de construction de cas de test				1					
55								1					

Figure 35 : matrice de traçabilité du contenu du projet

Cette matrice est un référentiel des exigences. J'ai préféré utiliser le terme de matrice de traçabilité plutôt que référentiel des exigences pour ne pas influencer, par la suite, les solutions proposées par les

équipes des projets pilotes par le choix de solution que j'ai personnellement fait pour mon projet. Chacune des 89 exigences y est définie par :

- **un identificateur** (colonnes A et C) : permet de tracer l'exigence tout au long du projet et de faciliter la communication (il est plus aisé et moins ambigu de dire : « la solution de TPI2.1.8 est ... » que « la solution de l'exigence : établir et maintenir un modèle de cycle de vie des tests est ... »). Cet identificateur est construit afin de mettre en évidence la provenance de l'exigence : TPI pour les exigences liées au processus de test et REQM pour celles liées au processus de gestion des exigences,
- **un projet pilote** (colonnes J, N, R, V, Z, AD) : indique le projet pilote durant lequel l'exigence est traitée. Un « 1 » dans une des cases d'un des 6 Projets Pilotes (PP1, PP2, ..., PP6) indique que l'exigence est réalisée au cours du projet pilote correspondant,
- **un livrable associé** (colonne I) : permet de s'assurer du traitement de l'exigence et aide à la vérification et à la validation.

4.2.4.2. Méthode de gestion de configuration

Le livrable du projet est un processus. Celui-ci est constitué d'un ensemble de documents et d'outils correspondant aux composants du processus. Ces derniers suivent un cycle de vie (créer, valider, obsolète, ...) lié à leurs modifications itératives au cours des projets pilotes et de validation. Il est important de pouvoir identifier tout au long et à l'issue du projet quel ensemble cohérent de composants forme une version du processus. Pour exemple, le projet pilote 1 produit la version 1.0 du processus, les projets pilotes 2 et 3 produisent la version 2.0 du processus, etc. L'activité qui permet d'identifier cet ensemble cohérent est la gestion de configuration.

Il n'existe pas chez D2T d'outil dédié à la gestion de configuration de ce type de composant. Pour identifier les versions de processus, j'ai défini une règle de gestion associée à un mécanisme d'étiquetage des composants. Cette règle se base sur un identifiant à deux nombres (par exemple : politique_de_test_V2.3.doc). Cette règle s'applique également à l'ensemble des documents produits au cours des PP et PV (ex : les spécifications, les documents de conception, de validation,...)

L'étiquetage est constitué de deux nombres formant un numéro de version v.r. Le Tableau 23 présente la signification de l'étiquetage.

v	r	Version	désignation
>0	=0	majeure	Composant ou document approuvé lors d'une revue (ex : politique_de_test_V2.0.doc).
>0	>0	mineure	Composant ou document en cours de réalisation. (ex : politique_de_test_V2.1.doc).

Tableau 23 : gestion de configuration - règle d'étiquetage

Un processus est en version majeure lorsque tous ses composants sont en version majeure. Un processus doit être en version majeure pour être utilisé par un projet de validation ou pour être déployé à tous les projets de développement.

4.2.4.3. Les revues

Le processus de développement applicable au sein du département Produits impose la mise en œuvre des phases suivantes : spécification, conception, réalisation, documentation et validation. Le passage d'une phase à la suivante est décidé au cours d'une revue (cf. chapitre 2.2.2.4).

Une revue évalue les produits de la phase précédant la revue pour déterminer s'ils sont aptes à permettre de réaliser les activités de la phase succédant la revue.

L'application de ce processus dans le cadre du projet d'amélioration est présentée par la Figure 36.

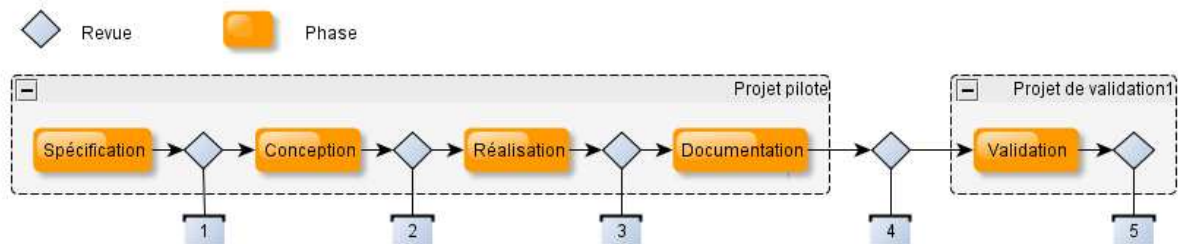


Figure 36 : application du processus de développement D2T au projet d'amélioration

Les phases de spécification, conception, réalisation et documentation sont effectuées lors des projets pilotes et la validation lors des projets de validation. Les revues sont indiquées par un losange bleu. Les revues 1 à 3 sont des revues standards. La revue 4 fait office de revue de documentation et de clôture du projet pilote et conditionne le démarrage du projet de validation. La revue 5 cumule les fonctions de revue de validation et de clôture du projet de validation.

Les revues chez D2T se réalisent sous forme de réunion formelle nécessitant la présence de plusieurs personnes (au moins 4). Les participants échangent sur le contenu du document et proposent des évolutions ou corrections. Ce genre de réunion peut fréquemment durer 2 à 3 heures. Vu le nombre important de revues liées au nombre de projets pilotes et de validation, ce mode de fonctionnement risque d'entraîner un surcoût (avec au minimum 28 revues, 4 participants et deux heures, le coût s'élève à plus de 19 000€).

Pour limiter ce coût, j'ai proposé un processus d'évaluation visant à réduire le temps des réunions. Cette procédure est basée sur un constat, celui que la revue est fréquemment placée immédiatement à l'issue de la rédaction des documents soumis à révision. Les participants n'ont alors pas suffisamment de temps pour en prendre connaissance. C'est ce manque de connaissance qui entraîne généralement un allongement de la durée de la revue. La procédure présentée en Figure 37 introduit une étape de lecture avec un délai suffisant (de minimum 10 jours) avant la tenue de la revue. Les participants sont ainsi en mesure de prendre connaissance du document, de signaler les problèmes et de les traiter avant la revue. Cette procédure doit minimiser la détection des problèmes (voir de faux problèmes) lors de la revue et de diminuer la durée de la revue.

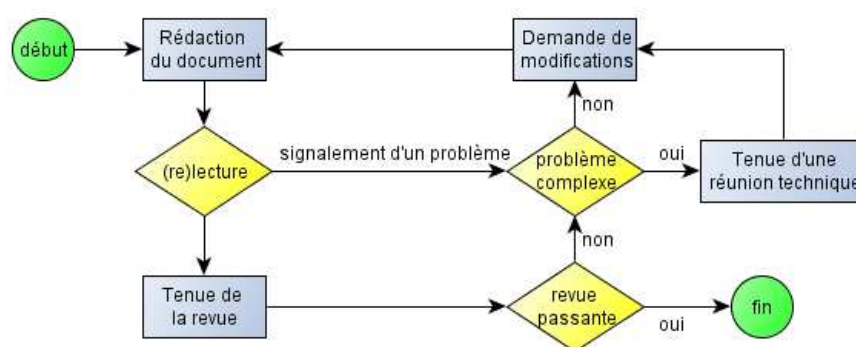


Figure 37 : procédure d'évaluation

4.2.5. La validation

L'objectif de cette activité vise à s'assurer que le nouveau processus répond au besoin et ainsi en évaluer l'efficacité. Pour ce faire, les itérations de processus issues des projets pilotes sont mises en pratique au cours de projets de développement désignés comme projet de validation. Le but étant de

s'assurer que le processus et surtout les outils et documents associés ont été faits pour l'usage attendu et pour les utilisateurs supposés (en somme déterminer l'efficacité) et dans le cas contraire d'identifier les anomalies. L'efficacité du processus est évaluée selon TPI pour les exigences de test tel que présentée au sous-chapitre 2.2.4 et CMMI pour la partie gestion des exigences.

Le principe de l'organisation des projets de validation (PV) consiste à réaliser une évaluation d'un projet de développement de logiciel (le projet de validation) au cours duquel le nouveau processus aura été appliqué. Les PV ont pour but :

- d'évaluer le niveau de qualité du processus amélioré (le processus est-il suffisamment clair pour être appliqué ? Est-il adapté à la situation ?),
- de mesurer la progression par rapport à l'état initial.

La réalisation de PV n'a pu être effectuée par manque de projets de développement. C'est une des raisons pour lesquelles une réorganisation du projet d'amélioration a été décidée. Cette dernière est présentée dans le chapitre 4.3.

4.2.5.1. Organisation

La Figure 38 présente le planning prévisionnel d'un PV. Celui-ci est prévisionnel, car sa durée est fortement dépendante de la durée du projet de développement désigné comme PV (représentée par la tâche « exécution »). Un PV est composé de quatre tâches :

- **le lancement** : identifier un projet apte à réaliser la validation du processus,
- **l'exécution** : appliquer le nouveau processus au cours du développement du logiciel,
- **l'évaluation** : préparer l'équipe et l'évaluation, collecter et analyser les preuves, réaliser le bilan,
- **la clôture** : analyser l'écart par rapport au coût/délai/qualité.

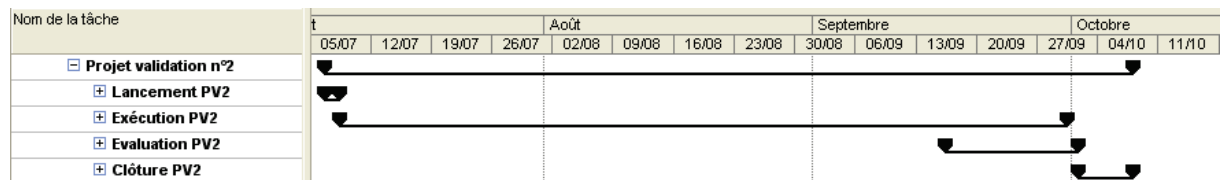


Figure 38 : planning prévisionnel d'un projet de validation

4.2.5.2. Lancement et exécution

Un PV est initié par une phase de lancement qui consiste à sélectionner le projet de développement désigné comme projet de validation. La sélection s'effectue sur un seul critère qui est la taille du projet. Le projet recherché doit avoir une taille de 2 à 6 hommes/mois, afin d'obtenir une durée compatible avec le délai du projet d'amélioration (un PV d'une taille trop importante n'apporterait des résultats d'évaluation qu'après la fin du projet d'amélioration donc trop tard pour exploiter les résultats au cours des PP qui lui succèdent).

Dès lors que le PV est identifié et désigné, une réunion d'engagement est réalisée avec l'équipe du PV. Cette réunion consiste à expliquer l'objectif et à informer sur l'emplacement de stockage des éléments relatifs au nouveau processus (documents processus, modèles et outils). Elle permet également d'obtenir l'engagement et de déterminer la motivation de l'équipe sur la mise en application du nouveau processus.

À l'issue de la phase de lancement, le PV passe en phase d'exécution. Cette dernière consiste en un déroulement normal du projet, à savoir développer le logiciel, sans intervention d'une personne du projet d'amélioration. Un budget de 3300€ (correspondant à 5 jours de travail) est alloué au budget initial du projet pour couvrir le coût nécessaire à la mise en œuvre du processus (appliquer un nouveau

processus nécessite d'en prendre connaissance, de l'assimiler pour ensuite pouvoir l'appliquer). Durant cette phase, lorsque l'équipe rencontre des problèmes liés au nouveau processus, elle signale le problème en tant qu'incident dans la base de gestion des anomalies TRAC. Ainsi tracé, cela évite d'oublier de les traiter (correction ou action de reprise lors des PP ultérieurs) avant la clôture du PV.

4.2.5.3. Évaluation

La phase d'évaluation a lieu lorsque l'exécution arrive à terme. Elle débute par une étape de préparation qui vise à définir la composition de l'équipe et à déterminer la date de réalisation de l'évaluation. La phase se termine par un bilan.

Pour réaliser l'évaluation, il est nécessaire de définir une méthode qui permettra de reproduire l'évaluation de manière identique à chaque PV et ainsi d'être en mesure de comparer les résultats obtenus.

Pour définir la méthode, je me suis inspiré de la méthode SCAMPI pour « Standard CMMI Method for Process Improvement ». SCAMPI est la méthode d'évaluation de processus CMMI développée par le Software Engineering Institute (SEI). La méthode SCAMPI n'est pas appliquée de manière stricte. D'une part car elle est contraignante et donc difficilement applicable dans le contexte de mon projet, même dans sa version la plus légère (il existe trois classes d'évaluation A, B et C avec la classe A étant la plus complète et utilisée lors des certifications officielles). D'autre part, je n'utilise que les indicateurs de réalisation de pratique du domaine de processus REQM pour l'évaluation de la partie « gestion des exigences » du nouveau processus. Pour la partie « gestion des tests », les indicateurs de réalisation sont ceux proposés par le modèle d'évaluation TPI (les points de contrôle cf. 4.1.1.2.4).

La méthode d'évaluation peut se résumer en cinq points :

- constituer une équipe d'évaluation,
- identifier le périmètre de l'évaluation,
- collecter les preuves objectives,
- évaluer le niveau de mise en œuvre du processus,
- fournir les résultats de l'évaluation (le bilan).

Le Tableau 24 présente la composition de l'équipe d'évaluation constituée d'au minimum deux personnes : le chef d'équipe et un membre. Le rôle de chef d'équipe est tenu par l'assistant qualité, sa pratique en conduite d'audits ISO fait qu'elle était la plus à même d'occuper ce rôle. L'équipe est complétée par un seul membre, rôle que j'occupe ou qu'occupe le chef de projet adjoint. Nous participons à l'évaluation pour apporter l'expertise des pratiques intégrées au nouveau processus.

Rôle	Fonction	Responsabilité
Chef d'équipe	Assistant qualité	Supervise l'organisation de l'évaluation Pilote l'évaluation Réalise le bilan
Membres de l'équipe d'évaluation	Chef du projet d'amélioration	Participe à l'évaluation
	Chef de projet Adjoint	Participe à l'évaluation

Tableau 24 : composition de l'équipe d'évaluation

Le périmètre de l'évaluation est identique pour tous les projets de validation. Par identique il faut comprendre qu'à chaque PV toutes les pratiques attendues seront évaluées, sachant que lors des premiers PV certaines n'auront pas encore été définies dans le processus. Les pratiques attendues correspondant aux 7 secteurs-clefs du niveau 1 de TPI (Tableau 2 page 22) et 5 objectifs spécifiques du domaine REQM de CMMI (Tableau 3 page 24). L'intérêt est de pouvoir comparer les résultats de chaque évaluation et ainsi déterminer une progression.

Les pratiques sont évaluées sur la base d'indicateurs de réalisations réparties de manière suivante :

Améliorer le processus de test pour atteindre le niveau 1 de TPI

- 36 indicateurs pour les 7 secteurs-clef issus du modèle TPI,
- 7 indicateurs, pour les 6 objectifs spécifiques du domaine REQM, issus de la méthode SCAMPI [22],
- 2 indicateurs, communs aux deux parties et correspondant à leur institutionnalisation, issus de la méthode SCAMPI [22].

La collecte des preuves objectives débute lorsque le projet de développement est terminé. Les preuves objectives sont déduites des indicateurs de réalisation de pratique. Ces dernières correspondent aux traces laissées lors de la mise en œuvre d'une pratique. Une fiche de description de test constitue, par exemple, une trace de la pratique « Établir des techniques informelles de description de test ». L'équipe du PV transmet les preuves à l'équipe d'évaluation. Certaines preuves pouvant être fournies de manière orale, il est prévu de réaliser des entretiens avec certaines personnes de l'équipe du PV.

L'évaluation consiste à déterminer, par l'analyse des preuves, un niveau d'atteinte de chaque pratique. Ces niveaux sont décrits dans le Tableau 25. La collecte et l'évaluation sont réalisées sur une période d'une journée.

Niveau		Signification
Rouge		L'objet de la pratique n'est pas couvert ou est mal couvert par l'ensemble des pratiques mises en œuvre.
Jaune		L'objet de la pratique est jugé partiellement couvert par l'ensemble des pratiques mises en œuvre, quelques lacunes ou problèmes ont été identifiés.
Vert		L'objet de la pratique est jugé couvert de façon adéquate par l'ensemble des pratiques mises en œuvre.
Non encore réalisée		Le stade auquel se trouve le processus ne permet pas encore la mise en œuvre de la pratique.

Tableau 25 : critères d'évaluation des pratiques

A l'issue de l'analyse des preuves, un bilan est dressé sous la forme d'un tableau récapitulatif tel que présenté dans l'exemple du Tableau 26.

Pratiques	Type de preuves	Niveau atteint
TPI15	Établir du reporting des anomalies	Une base TRAC est utilisée pour tracer les constats identifiés au cours du développement 
REQM	Gérer les exigences	Un référentiel des exigences est créé pour tracer les exigences du produit à développer 
...	...	...

Tableau 26 : exemple de bilan d'évaluation

4.2.5.4. Clôture

Lors de la clôture du PV une analyse des écarts du coût et du délai estimé/réel est présentée au comité de pilotage. Le chef de projet d'amélioration présente également le bilan de l'évaluation et propose un plan d'action lorsque des pratiques sont de niveau jaune ou rouge ainsi qu'un délai de réalisation des actions. Si les pratiques sont toutes au niveau vert ou « non encore réalisées » ou que le plan d'action et le délai de réalisation proposés pour la correction des pratiques au niveau rouge et jaune convient au comité de pilotage, le projet pilote est clôturé. Les résultats de l'évaluation sont communiqués au sein du département Produits.

4.3. Réorganisation du projet et déploiement

Le manque de nouveaux projets de développement pour alimenter le projet d'amélioration a nécessité de revoir son organisation. Cette section présente les deux propositions de réorganisations envisagées dont l'une consiste à poursuivre la définition du processus grâce à un groupe de travail constitué de collaborateurs du département Produits et non plus par des projets pilotes et la seconde consistant à interrompre la définition du processus et de le mettre en application en l'état. La seconde solution a été retenue.

4.3.1. Situation du projet en octobre 2010

En octobre 2010, lors du bilan projet, à l'issue des projets pilotes 2 et 3, la situation est la suivante :

- **il y a un manque de projets de développement** : pour alimenter les projets pilotes et surtout les projets de validation rendant impossible l'évaluation du nouveau processus tel que prévu initialement, ce qui impacte la qualité (pas d'évaluation) du projet d'amélioration,
- **la durée des projets pilotes** : plus importante que prévue (8 mois en moyenne au lieu de 4 à 5 mois) impactant directement le coût (temps nécessaire à la gestion de projet plus important).

Les Tableaux 27 à Tableau 29 indiquent le niveau auquel le projet a mis en œuvre les pratiques de gestion de tests issues du modèle TPI et de gestion des exigences issues du domaine REQM du modèle CMMI. Chacun des trois tableaux comporte une colonne à droite, nommée « état », indiquant le niveau de mise en œuvre. Une pratique est caractérisée par trois états possibles :

- **introduit** : la pratique est partiellement mise en œuvre dans le nouveau processus,
- **réalisé** : la pratique est totalement mise en œuvre,
- **non réalisé** : la pratique n'est pas encore définie dans le processus.

n°SC	Objectif de la pratique du secteur-clef	État
1	Établir et maintenir une stratégie de test pour les tests en aval	Introduit
2	Établir et maintenir un modèle de cycle de vie des tests	Introduit
5	Établir des techniques informelles de description de test	Réalisé
11	Établir l'engagement et la motivation	Introduit
15	Établir du reporting des anomalies	Réalisé
16	Gérer les anomalies (en interne)	Réalisé
18	Gérer le processus de test	Introduit

Tableau 27: les apports liés au modèle TPI

n°	Objectif de la pratique	État
1	Définir une politique de test	Réalisé
2	Réaliser un bilan des tests	Réalisé

Tableau 28 : les apports non liés au modèle TPI

n°	Objectif de la pratique du domaine REQM	État
1	Obtenir une compréhension des exigences	Introduit
2	Obtenir l'engagement sur les exigences	Non réalisé
3	Gérer les modifications aux exigences	Non réalisé
4	Maintenir la traçabilité bidirectionnelle des exigences	Introduit
5	Identifier les incohérences entre les produits du projet et les exigences	Non réalisé

Tableau 29 : les apports liés au domaine REQM du modèle CMMI

À l'issue du 3^{ème} projet pilote 43 exigences sur un total de 89 prévues ont été réalisées. Les pratiques proposées pour répondre à ces 43 exigences ont été validées par le comité de pilotage pour, par la suite, être vérifié au sein même de projet de développement et ainsi démontrer leur efficacité.

Le manque de projets de développement est un risque qui avait été identifié au lancement du projet. Il avait été jugé peu probable aux vues du nombre de projets prévus sur 2010. Le niveau de probabilité de ce risque a été revu à la hausse dès la fin du premier trimestre 2010. Le comité de pilotage l'a jugé acceptable dès lors qu'il n'impactait que le délai du projet (le délai étant la variable d'ajustement du projet cf. chapitre 2.4 Les exigences projet).

En juin 2010, le manque de projet de développement a entraîné la décision, sur accord du comité de pilotage, de supprimer le premier projet de validation afin de privilégier les projets pilotes 2 et 3. En octobre 2010, le manque de nouveau projet est tel qu'il n'est plus possible de démarrer les projets pilotes 4 et 5, ni le projet de validation 2.

Conscient de l'inefficacité de l'organisation du projet, j'ai soumis au comité de pilotage deux propositions de réorganisation du projet.

4.3.2. Les propositions de réorganisation

4.3.2.1. Proposition 1

Cette première proposition de réorganisation (Figure 39) vise à ne plus utiliser les projets de développement que pour alimenter les projets de validation. La définition du processus n'est plus réalisée par des projets pilotes, mais par un groupe de travail. Celui-ci est constitué de personnes (entre 5 à 7) sélectionnées parmi les collaborateurs du département Produits et ayant montrées leur motivation à participer au projet d'amélioration.

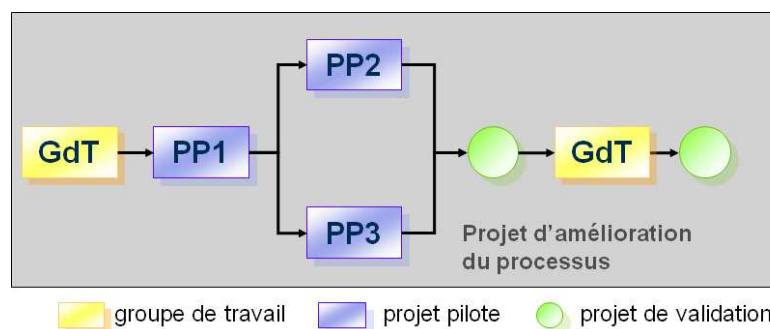


Figure 39 : proposition de réorganisation 1

Le groupe de travail est identifié en jaune sur la Figure 39 entre les deux projets de validation. Dans cette organisation, ce groupe de travail ne pourra démarrer au plus tôt qu'au deuxième trimestre 2011. Il sera précédé d'un projet de validation qui doit permettre de déterminer l'ensemble des exigences (les nouvelles ainsi que celles à reprendre) que le groupe doit traiter. Ce dernier doit ensuite terminer de définir le processus, lequel est ensuite de nouveau évalué par un projet de validation.

L'avantage de cette organisation est qu'elle ne requiert plus que deux projets de développement et permet d'obtenir un processus couvrant l'ensemble des exigences. L'inconvénient est qu'il nécessite un allongement de la durée, sur l'année 2011 voire même à début de 2012, avant le déploiement du processus (2 x 8 mois pour les projets de validation et 2 mois pour finaliser la définition du processus soit environ 18 mois).

4.3.2.2. Proposition 2

La seconde proposition de réorganisation s'appuie sur une des contraintes imposées au projet à savoir l'évolutivité (cf. chapitre 2.4.3 - page 25). Cette dernière a pour effet d'obliger chaque projet pilote à produire un processus exploitable et applicable.

Partant du constat qu'à l'issue des projets pilotes 2 et 3 :

- nous disposons d'un processus applicable,
- qu'il est difficile de déterminer une échéance de sortie des autres itérations du processus,
- qu'il faut évaluer ce nouveau processus avant de réaliser une nouvelle itération.

J'ai proposé comme seconde réorganisation :

- d'interrompre les évolutions du processus,
- de mettre le processus obtenu (désigné comme version 1.0 sur la Figure 40) en application sur l'ensemble des nouveaux projets de développements venant à être lancés à partir de novembre 2010 et ce durant toute l'année 2011.



Figure 40 : proposition de réorganisation 2

Cette période de « mise en pratique » du nouveau processus est accompagné par du support réalisé par l'équipe projet d'amélioration (principalement le chef de projet et le chef de projet adjoint). À l'issue de cette période, une évaluation est réalisée sur le même principe que pour les projets de validation.

L'avantage de cette proposition est d'éviter un essoufflement du projet d'amélioration et une démotivation de son équipe du fait de ne pas voir une fin du projet. L'évaluation est possible sur plusieurs projets (au moins deux voire trois) donc plus juste. À la suite de l'évaluation un nouveau projet d'amélioration peut être lancé avec un nouvel objectif tenant compte des résultats de l'évaluation.

4.3.3. Planification de la nouvelle organisation

La proposition retenue par le comité de pilotage est celle que j'ai soutenue (la proposition 2). Ce choix a été réalisé du fait :

- de son avantage à maîtriser le projet (il vaut mieux mettre en application le processus en l'état que de voir le projet « tirer en longueur »),
- d'un changement majeur qui est mon départ du projet, à la demande de la Direction, pour mener une mission de 18 mois dans une autre filiale du groupe IFPEN.

En collaboration avec le chef de projet adjoint, qui reprend le poste de chef de projet après mon départ, j'ai défini un plan de cette nouvelle organisation et je l'ai faite validée par le comité de pilotage. Cette nouvelle organisation est considérée comme un projet de « déploiement » avec un contenu et un budget.

4.3.3.1. Le contenu

Le but du projet étant de déployer le nouveau processus de test afin qu'il soit appliqué au cours des projets de développement, j'ai proposé une organisation en trois phases, qui sont :

- la mise en application,
- l'accompagnement,
- la maintenance.

Mise en application

La mise en application du processus de test est une phase relativement simple, puisqu'elle consiste en deux actions visant à permettre aux équipes projet de développement à appliquer le nouveau processus de test. D'une part, une action consiste en l'intégration des modèles et guides, réalisés par le projet d'amélioration, dans le système de management de la qualité (SMQ). Cette action est une formalité puisque l'ensemble des éléments a préalablement été validé par la Direction Qualité. D'autre part, en de la communication auprès des équipes projets du département Produits sur la mise en application du nouveau processus de test.

Accompagnement

L'accompagnement est réalisé par le chef de projet et consiste à intervenir auprès des équipes projet de développement, devant appliquer le processus de test, pour leur apporter du support. La phase d'accompagnement débute dès que la phase mise en application est terminée. Le but n'est pas de faire à leur place mais de les conseiller sur la manière de faire ou d'apporter des explications sur l'utilisation des modèles.

Le chef de projet déploiement juge de la pertinence de l'intervention. Par exemple, s'il constate que l'équipe projet de développement n'a pas préalablement fait l'effort de consulter les guides d'utilisation, il peut décider que son intervention n'est pas justifiée.

Maintenance

Comme tout système, un processus et les éléments qui le constituent comportent des anomalies. C'est au cours de la phase d'accompagnement que ces anomalies sont détectées. Il est alors nécessaire de réaliser des corrections. Ces corrections sont réalisées lors de phase de maintenance. Elle est réalisée par le chef de projet avec l'aide de l'assistant Qualité.

Pour ce faire, une base de signalement d'anomalies est créée avec l'outil TRAC. Lorsqu'une équipe projet de développement détecte une anomalie (par exemple, une partie d'un guide d'utilisation qui n'est pas clair ou une partie d'un modèle ne peut pas être mise en pratique) elle l'a signale dans la base.

Une analyse des anomalies signalées dans la base est faite au moins une fois par trimestre (plus si le chef de projet l'estime nécessaire) avec une première analyse en Janvier 2011. Cette analyse vise à déterminer si les anomalies sont avérées et de leur coût de corrections. Deux cas se présentent alors, selon que l'ensemble des corrections nécessite :

- **moins d'une journée de travail** : les modifications sont faites,
- **plus d'une journée** : une proposition de correction est soumise préalablement au responsable du département et à la Direction Qualité, sur un accord favorable les corrections sont faites, dans le cas contraire elles sont reportées pour être incluses dans les objectifs d'un nouveau projet d'amélioration.

Lorsque les corrections sont réalisées, les éléments modifiés du processus sont intégrés dans le SMQ et une information sur les modifications est faite aux équipes.

4.3.3.2. Coût et délai

Le coût

Le coût global est estimé entre 18.5 à 21.1 K€. Le détail de ce coût est présenté dans le Tableau 30. Ce coût est uniquement lié à du temps de travail, avec plus de 60% du coût dédié à la phase d'accompagnement dont l'effort exprimé en équivalent jours/homme se situe entre 18 à 20 jrs/h. Environ 2 K€ est attribué à la gestion de projet pour effectuer le suivi et la clôture du projet.

Phases	Coût en K€	Équivalent en jrs/h
Mise en application et accompagnement	11.9 à 13.2	18 à 20
Maintenance	3.3 à 4.0	5 à 6
Évaluation	2.0	3
Gestion de projet	1.3 à 2.0	2 à 3
Coût global	18.5 à 21.1	28 à 32

Tableau 30 : coût du projet de déploiement

Le délai

Le projet de déploiement se déroule sur une période allant de début Novembre 2010 à fin Décembre 2011. La Figure 41 présente l'ordonnancement des diverses phases et tâches du projet.

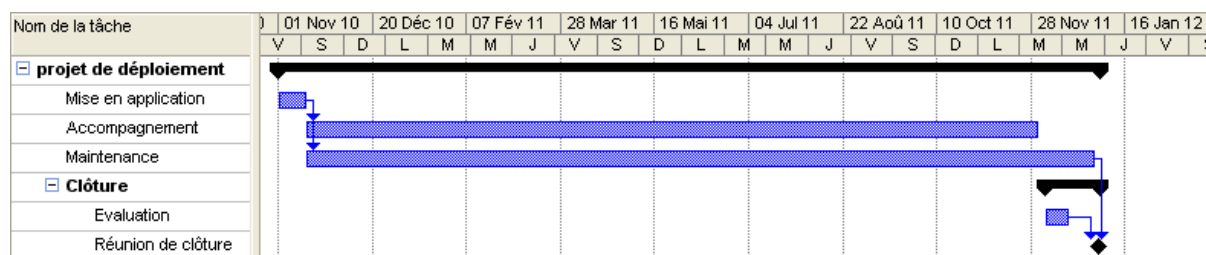


Figure 41: planning du projet de déploiement

Suivi et maîtrise

La taille de l'équipe projet de déploiement étant réduite au chef de projet et ponctuellement à l'assistant qualité, le suivi et la maîtrise du projet est réduit au plus simple. Un suivi des temps passés est effectué par activité et un reporting mensuel est réalisé par le chef de projet indiquant les points marquants du mois et le budget consommé.

4.3.3.3. Clôture

Un des objectifs du projet de déploiement consiste à évaluer le nouveau processus. La clôture du projet s'effectue sur la base d'un rapport d'évaluation. L'évaluation du processus de test est faite selon le mode opératoire décrit dans le chapitre 4.2.5.3 page 91. Le projet est clôturé sur présentation des conclusions de l'évaluation et de l'état de la base TRAC (toutes les anomalies devant être traitées) au responsable du département et à la Direction Qualité.

Conclusion de cette partie du document

Dans cette partie a été présentée l'organisation du projet d'amélioration selon des préconisations du PMBOK. Celles-ci sont de créer une structure de découpage du projet (SDP) basée sur le contenu du projet et enrichie suite au traitement d'autres domaines de management telle la gestion des risques, du coût, des ressources humaines,... Cette SDP est ensuite utilisée pour réaliser le planning qui sert à la conduite du projet.

Pour répondre aux contraintes de visibilité et évolutivité imposées au projet, ce dernier a été découpé en une succession d'étapes visant à définir (les projets pilotes) puis à valider (les projets de validation) des incréments du nouveau processus.

Ces étapes sont mises en œuvre au cours de projet de développement de logiciel. Ce mode de fonctionnement du projet d'amélioration s'est avéré inefficace. Une nouvelle organisation a été proposée et acceptée par le comité de pilotage. Cette nouvelle organisation vise à mettre en application le nouveau processus. Celui-ci ne couvre pas l'ensemble des attentes initialement prévues mais a suffisamment évolué pour permettre une amélioration de la qualité des produits. Ce nouveau processus sera évalué à l'issue d'une année de mise en pratique.

5. Bilan

Objectif de cette partie du document

Au moment, novembre 2010, où je quitte le projet, celui-ci n'est pas terminé. La décision de ne pas réaliser les trois derniers projets pilotes et de mettre en application, durant toute l'année 2011, le processus issu des trois premiers projets pilotes, ne permet pas d'effectuer un bilan définitif du projet. L'atteinte des objectifs du projet en termes qualitatifs ne sera, par exemple, réellement possible qu'à la suite de l'évaluation réalisée à l'issue de la phase « mise en pratique ». Le bilan qui est fait dans ce chapitre correspond à l'état du projet au moment de la mise en application de la nouvelle organisation.

Ce bilan est réalisé selon quatre visions. La première est purement « comptable » et s'appuie sur des éléments chiffrés. Le chapitre 5.1 est ainsi consacré à l'analyse du niveau d'atteinte du coût, délai, fonctionnalités et qualité par le projet.

Le chapitre 5.2 met l'accent sur les apports du projet au département Produits.

Le chapitre 5.3 est destiné au retour d'expérience pour capitaliser mon expérience acquise lors de ce projet et en faire profiter toute autre personne amenée à réaliser des travaux similaires.

Enfin dans la dernière partie, je présenterai ce que ce projet m'a apporté sur le plan personnel.

5.1. Bilan projet

Le but de ce chapitre est de faire le bilan du projet en termes de coût, délai, fonctionnalités et qualité. Ce bilan est réalisé suivant deux états : avant et après révision. Par révision il faut comprendre un changement de périmètre du projet dû d'une part à des modifications des exigences (ajout, suppression, suspension) en cours de projet et d'autre part, suite à la réorganisation décidée en novembre 2010.

Le coût initial du projet avait été estimé à 222 k€. Ce coût correspond à la réalisation d'une politique de test par un groupe de travail, des six projets pilotes, des quatre projets de validation ainsi que la gestion du projet et l'intégration du chef de projet adjoint. Après révision, ce coût est estimé à 119 k€ et comprend la définition de la politique de test par un groupe de travail, de trois projets pilotes, ainsi qu'une partie du budget relatif à la gestion du projet et l'intégralité du budget d'intégration du chef de projet adjoint. Ce coût ne prend pas en compte le budget nécessaire au support et à la maintenance au cours de la phase de mise en application ni de l'évaluation prévue à l'issue.

Le coût réel du projet (à la clôture de PP 2 et 3 en novembre 2010) est de 141 k€, soit supérieur de 18% à la prévision. Cet écart est principalement dû à un surcroît de travail en suivi et maîtrise du projet d'amélioration du fait que les PP ont duré plus longtemps que prévu (cf. chapitre 4.2.3.7). Le Tableau 31 présente un état synthétique du bilan du coût projet.

	prévu	réel	Écart
Avant révision	222 k€	141 k€	--
Après révision	119 k€	141 k€	+18%

Tableau 31 : bilan coût (en k€)

Le délai est la variable d'ajustement du projet. Il n'y avait pas de réelle attente en termes de respect du délai. La durée initialement prévue pour la réalisation (débutant par la définition de la politique de test au 1^{er} Décembre 2009 et se terminant par la clôture du projet à l'issue du 4^{ème} projet de validation) était fixée à fin 2010 soit 12 à 13 mois.

En novembre 2010, le bilan (cf. Tableau 32) met en évidence que 11 mois ont été nécessaires pour définir la politique de test et réaliser trois projets pilotes. Une clôture du projet est prévue, à la suite de la révision, au début de l'année 2012 à l'issue de l'évaluation. La durée après révision est de 26 mois.

	prévu	réalisé
Avant révision	12 mois	11 mois
Après révision	26	--

Tableau 32 : bilan délai (en mois)

Le bilan du projet en termes de couverture des exigences est présenté dans le Tableau 33.

	prévues	réalisées	couverture
Avant révision	89	43	49%
Après révision	43	43	100%

Tableau 33 : bilan de la couverture des exigences fonctionnelles (en nombre d'exigences)

Initialement le projet comportait 89 exigences (cf. chapitre 4.1.2.3.1). Ce nombre a été réduit à 43 après la révision de la manière suivante :

- une exigence du projet pilote 1 a été supprimée car estimée ne pas être du ressort du projet,
- une exigence du projet pilote 3 a été reportée,
- les exigences des projets pilotes 4 à 6 ont été reportées sur un nouveau projet prévu à la suite de celui-ci.

La couverture des exigences avant révision est de 49% et de 100% après révision. Cette couverture n'est toutefois relative qu'à la réalisation, elle reste à confirmer au travers de l'évaluation prévue en début 2012.

Les attentes en termes de qualité du nouveau processus étaient de trois types (cf. chapitre 2.4.3) :

- **adaptivité** : la politique de test, avec ses niveaux de test, apporte la flexibilité et l'adaptabilité au processus. Le choix du niveau (faible, moyen, bon,...) au début du projet de développement permet de le contraindre plus ou moins selon la situation,
- **conformité réglementaire** : l'application du nouveau processus ne remet pas en cause l'ancien. Toutes les nouvelles tâches (planifier et concevoir les tests, faire le bilan, créer le référentiel des exigences, ...) ne font qu'enrichir l'ancien processus de développement. Durant la période de mise en application sur 2011, les projets de développement peuvent appliquer le nouveau processus sans risquer de créer des non-conformités,
- **facilité d'utilisation** : la définition du processus par les collaborateurs rend implicite la facilité d'utilisation, mais cela reste tout de même à mesurer par l'évaluation prévue à la suite de la « mise en pratique ».

5.2. Apports du projet à l'entreprise

Dans cette partie sont présentés les divers apports du projet. Bien que toutes les exigences initialement prévues n'aient pu être couvertes, le projet a toutefois permis d'introduire les sept secteurs-clefs de TPI et une majeure partie (2 sur 5) des pratiques REQM.

À ce stade, il est difficile de mesurer l'impact du nouveau processus et des pratiques associées sur l'amélioration de la qualité des produits logiciel de D2T. L'ensemble des collaborateurs du département Produits doit se l'approprier. Néanmoins au-delà des évolutions apportées au processus de développement, le projet a surtout permis une prise de conscience, par les collaborateurs du département Produits, que tester des logiciels n'est pas une activité empirique.

La définition d'un objectif de test d'un projet de développement est passé de « il faut que ça fonctionne » ou « il faut tester » vers un état de vision plus détaillée de couverture des exigences fonctionnelles et la possibilité d'évaluer un écart entre ce qui était prévu au début et ce qui est réellement réalisé à l'issue du projet de développement. Le chef de projet a également la visibilité sur l'avancement des tests grâce au reporting. **Il est possible d'en conclure une élévation du niveau d'indépendance de la réalisation des tests par une élévation du paramètre financier.** Un budget pour les tests est à présent défini, justifié et suivi au cours du projet.

La mise en place de fiche de description de test permet de capitaliser et de maîtriser l'effort de test de non-régression. Lorsque le projet de développement est terminé, les fiches sont conservées pour être utilisées lors de la maintenance ou des évolutions du logiciel pour permettre de réaliser les tests de non-régression. Les fiches étant très détaillées permettent, entre autres, de faire faire une partie des tests par des personnes avec peu de compétences sur le produit sous test et apportent ainsi une flexibilité dans l'utilisation des ressources humaines.

La principale avancée réside dans la compréhension du concept de couverture des exigences par les tests et par le développement. Avant le projet, les logiciels étaient développés sur un périmètre fonctionnel « vague ». Les testeurs indiquaient quelles exigences fonctionnelles avaient été testées mais pas celles qui ne l'étaient pas. Le choix des tests réalisés restait à la discrétion de celui qui les

réalisait. À présent, grâce au référentiel des exigences, il est possible d'avoir une couverture des exigences fonctionnelles par les tests. Ce concept de couverture a tellement plu que les développeurs ont demandé d'avoir l'équivalent au niveau développement des exigences dans le référentiel.

Il reste néanmoins une difficulté au niveau de la gestion des exigences. Cette difficulté est de déterminer le bon niveau de granularité pour gérer les exigences. Une granularité trop fine rend difficile voire impossible le maintien à jour du référentiel tout au long du projet. Un découpage trop macroscopique des exigences fonctionnelles est inefficace car vague.

Bien sûr, cette prise de conscience n'est pas générale, il y a et il y aura toujours une partie des collaborateurs qui estimeront que ces pratiques ne sont rien d'autres que de la « paperasse » inutile et qui fait perdre du temps. Mais du moins l'équipe de management du département Produits ainsi que quelques collaborateurs sont convaincus de l'utilité des améliorations apportées et c'est le niveau nécessaire pour pérenniser ce processus lors de la phase de mise en pratique qui a lieu sur l'année 2011.

5.3. Retour d'expérience

Une analyse du déroulement du projet est présentée dans ce chapitre afin de capitaliser mon expérience acquise et en faire profiter toute autre personne amenée à réaliser des travaux similaires. Il y est présenté dans un premier temps ce qui a bien fonctionné au niveau du projet pour, dans un second temps, proposer des pistes d'amélioration.

5.3.1. Ce qui a bien fonctionné

Les pratiques qui ont eu un effet positif sur le projet sont :

- la méthode de définition du nouveau processus,
- la communication,
- le suivi du projet grâce à la matrice de traçabilité.

La méthode de définition du processus par les projets pilotes est directement inspirée des « process group » basé sur la capitalisation des compétences des collaborateurs [30]. Ce mode de fonctionnement est efficace en termes d'émergence de solutions « pratiques ». De plus, le fait de devoir appliquer les pratiques proposées, au cours du pilote, oblige l'équipe à rester sur des solutions simples et pragmatiques.

Cette méthode est également une aide à la conduite du changement car les processus et procédures sont souvent ressentis par ceux qui doivent les appliquer, comme étant définis par des personnes ne connaissant pas le métier (le développement de logiciels). Dans le cas présent, les procédures, modèles, directives ont été définis par ceux qui devront les appliquer (les développeurs, chefs de projet, responsables et assistants produits).

La communication, toute la littérature et les experts du domaine de l'amélioration de processus sont unanimes sur le fait qu'elle tient une part importante dans la réussite du projet. Il faut très tôt commencer à expliquer, présenter, rassurer et convaincre. Je ne présenterai pas toutes les actions de communication mises en œuvre au cours du projet. Je désire en présenter deux qui ont été appréciées.

La première a été de créer un logo permettant d'identifier tous les éléments (principalement les documents : rapports, courriels d'information, support de présentation, ...) liés au projet. Par exemple, dans une présentation générale des activités du département Produits, il était ainsi aisé d'identifier les actions réalisées par mon projet.

La seconde était de faire participer les personnes des équipes des projets pilotes à la présentation des évolutions. Cela a évité que les améliorations ne soient trop associées uniquement à l'équipe projet d'amélioration.

Enfin la dernière pratique qui, selon moi, a été bénéfique au projet est l'utilisation de la matrice de traçabilité. Le comité de pilotage a apprécié la qualité de la visibilité que cet outil a apportée au niveau

du suivi des exigences tout au long du projet. Cet outil ne remplace pas un planning mais est complémentaire. Il est évident que sans la matrice, j'aurais découvert tardivement que certaines exigences n'avaient pas été traitées. Elle m'a également servi de démonstrateur du concept de gestion des exigences appliqué à mon projet.

5.3.2. Ce qui aurait pu être amélioré

Avec le recul, que j'ai à présent, je suis également conscient des améliorations qu'il faudrait apporter si d'ailleurs ce type de projet devait être réitéré. Ces améliorations porteraient sur deux aspects du projet :

- la communication,
- l'utilisation d'un outil de gestion des exigences.

Je me suis rendu compte que la communication doit être différente selon le public visé. Ainsi la direction et le management intermédiaire désirent connaître les bases sur lesquelles reposent les choix d'amélioration et sont, de fait, rassurés de savoir qu'ils proviennent de modèles reconnus tel TPI et CMMI. Les opérationnels, au contraire, sont réticents à l'idée des concepts provenant de modèles d'amélioration. Cela accentue leurs craintes de devoir réaliser des tâches qu'ils considèrent inutiles. Il est préférable d'éviter d'exprimer les objectifs du projet d'amélioration en termes de conformité par rapport à un modèle. Il est ainsi préférable de dire « on va éviter de travailler sur des exigences vagues » plutôt que « Obtenir une compréhension des exigences ».

Avant que le projet ne débute, j'avais évalué plusieurs outils de gestion des exigences, XQual, Salomé, TestLink, pour n'en citer que quelques-uns disponibles en open source. Ces évaluations ont eu lieu dans le cadre de mes activités de test au sein de projets de développement. Au lancement du projet d'amélioration, je me suis refusé à introduire d'emblée ce type d'outil pour centrer le travail sur la méthode et non sur l'outil. À présent, je reste persuadé qu'il ne fallait pas introduire d'outil, mais qu'il est bon de montrer que ces outils existent car les opérationnels sont inquiets lorsqu'ils voient la lourdeur de la gestion des exigences grâce à un tableur type Excel ; mais sont rassurés de savoir qu'après une phase de compréhension de la méthode avec le tableur ils auront la possibilité de passer sur un outil plus efficace.

5.4. Bilan personnel

Cette section du bilan est destinée à la présentation de ce que le projet m'a apporté personnellement.

Cela fait, à présent, plus de dix ans que j'occupe la fonction d'assistant produits avec comme principale tâche le test des logiciels de l'entreprise. J'ai commencé sans avoir la moindre notion de génie logiciel. Au début, je me suis appuyé sur mes connaissances en tant qu'ancien utilisateur. Mais cela s'est avéré rapidement insuffisant et pas assez efficace. J'ai donc décidé de me former et c'est le CNAM qui m'a permis d'acquérir les connaissances nécessaires à rendre mon travail de test plus efficace. J'ai par la suite entrepris de me faire certifier ISTQB pour montrer que testeur est un métier à part entière et non une tâche annexe. Le projet m'a permis de communiquer et de transmettre une partie de mes connaissances et de mon expérience acquises tout au long de ces dix années à mes collègues du département Produits.

Ce projet m'a également permis d'acquérir des compétences en pilotage de groupes de travail. C'est une activité complexe car elle consiste à atteindre un objectif par l'intermédiaire du travail des personnes formant le groupe. Le CNAM m'a fourni, lors de mon cursus, une boîte à outils pour ce genre de situation. Ces outils s'appellent : QQCOCQP⁵, SMART⁶, brainstorming⁷, mindmapping⁸. Je ne prétends pas avoir acquis la maîtrise de ces outils mais leur mise en œuvre, au cours du projet, m'a permis de mieux appréhender leurs avantages et inconvénients.

Enfin le dernier point que je veux mettre en lumière dans ce bilan personnel, est qu'il n'est nullement nécessaire de « réinventer la roue ». J'entends par là, que tout ce dont on a besoin pour améliorer les méthodes de travail a été capitalisé par une multitude d'experts dans des recueils tels CMMI, IEEE, PMBOK, TPI, TMap, pour ne citer que ceux que j'ai exploités dans le cadre de mon projet. Ma formation au CNAM, m'a permis de démarrer la mise en application des trois premiers. Par la suite l'acquisition de quelques bons livres et surtout l'adhésion à des communautés actives aide beaucoup. Pour ma part je suis activement celle de CMMI-en français [30] créé par Richard Basque auteur de plusieurs ouvrages sur CMMI et celle du Comité Français du Test Logiciel [25].

Conclusion de cette partie du document

Au terme de 11 mois, un nouveau processus a été défini couvrant 49% des exigences soumises initialement au projet. Le coût réel de la réalisation s'élève à 141 k€.

Bien qu'uniquement une partie des exigences ait été implémentée dans le nouveau processus, celui-ci permet toutefois l'introduction de l'ensemble des pratiques des sept secteurs-clefs du niveau 1 du modèle TPI. Une politique de test existe et permet une gestion homogène des activités de tests sur l'ensemble des projets de développement de logiciels de D2T. Cela permet une élévation du niveau d'indépendance dans la réalisation des tests. En ce qui concerne la gestion des exigences, la mise en place du concept de référentiel permet de tracer les exigences tout au long des projets de développement voire durant toute la vie du logiciel.

Les retours d'expérience à retenir sont :

- l'importance de la communication dans ce type de projet,
- l'efficacité de la définition du processus par des groupes d'utilisateurs.

Enfin, sur le plan personnel, ce projet m'a permis d'acquérir des compétences en animation de groupes de travail, en gestion de projet et de mettre en pratique l'utilisation de modèles d'amélioration.

⁵ QQCOCQP : correspond aux premières lettres de Qui, Quoi, Comment, Où, Combien, Quand, Pourquoi.

⁶ SMART : correspond aux premières lettres de Spécifique, Mesurable, Acceptable, Réaliste, Temporel (déterminé dans le Temps) et est une méthode visant à formuler des objectifs de manière claire et précise.

⁷ Brainstorming : est une technique de résolution créative de problème conçue en 1935 par Alex Osborn.

⁸ Mindmapping : technique consistant à représenter sous forme de liens sémantiques différentes idées ou de liens hiérarchiques entre différents concepts.

6. Conclusion générale

Le département Produits, de la société D2T, définit et développe divers systèmes destinés à la gestion des bancs d'essai de moteur thermique. En 2009, la direction générale de D2T a exprimé le besoin de mener des actions dans le but d'améliorer la qualité des produits. Une de ces actions consiste à mieux vérifier et valider les produits, ce qui revient à améliorer le processus de test.

Pour mener à bien cette action, un projet a été confié à une équipe dans laquelle j'occupe la fonction de chef de projet AMO. Le coût du projet est estimé à 222 K€ et sa durée à un an. L'amélioration du processus consiste à introduire les sept secteurs-clefs du niveau 1 du modèle TPI. Une amélioration de la gestion des exigences est également incluse au périmètre du projet.

Le nouveau processus, à l'issue du projet, se concrétise par un ensemble d'activités (de pratiques) intégrées au processus de développement déjà en place. Ces activités visent à :

- identifier et caractériser les exigences du logiciel à développer,
- planifier les tests,
- concevoir les tests et les formaliser,
- capitaliser l'expérience à travers un bilan.

Pour aider à mettre en œuvre ce processus, un ensemble d'outils est disponible. Ces outils se présentent sous la forme de fichiers Excel ou Word. L'un d'entre eux est le référentiel, celui-ci est utilisé pour l'identification des exigences et la planification des tests. D'autres outils, telle la fiche de description de tests, permettent de formaliser l'exécution des tests et anticiper leur conception.

Ce projet s'appuie sur un ensemble de modèles, normes et méthodes, à savoir :

- **TPI** : utilisé pour évaluer et identifier les opportunités d'amélioration du processus de test,
- **TMap** : utilisé comme référence pour mettre en œuvre les améliorations du processus de test,
- **CMMI** : utilisé pour évaluer et identifier les opportunités d'amélioration du processus de gestion des exigences,
- **IEEE 830** : utilisé comme référence pour mettre en œuvre les améliorations du processus de gestion des exigences,
- **PMBOK** : utilisé comme guide pour organiser, suivre et maîtriser le projet.

L'organisation du projet est basée sur une succession de groupes de travail (les projets pilotes) qui visent à définir le nouveau processus de manière itérative et incrémentale. Des projets de développement de logiciel (les projets de validation) sont utilisés pour valider les itérations du processus. Ce choix d'organisation vise principalement à aider à la conduite du changement en faisant participer les futurs utilisateurs par une capitalisation de leurs pratiques.

A l'issue de 11 mois, le nouveau processus est déployé sur les projets venant à être lancés. Ce nouveau processus couvre 50% des exigences initialement prévues. Néanmoins, il permet l'introduction et la mise en œuvre de tous les secteurs-clefs du niveau 1 de TPI.

Une des principales difficultés dans ce type de projet est liée à la dimension humaine. Les pratiques à mettre en place sont, à ce niveau de maturité, très basique voire « simpliste » encore faut-il qu'elles soient acceptées, à la suite du déploiement, par ceux qui doivent les appliquer. Le problème à résoudre est à 20% de type technique et à 80% de type humain.

Un processus n'est jamais parfait, il doit être adapté à la situation. Lorsque la situation évolue il est nécessaire de faire évoluer le processus. Par ce projet, la roue de Deming a avancée.

7. Annexes

7.1. Les secteurs-clef TPI : description détaillée

Le tableau ci-dessous présente une description détaillée des vingt secteurs-clef que comporte le modèle TPI. Ce tableau est extrait du livre « Test Process Improvement » [8].

	Secteur-clef	Description
1	Stratégie de test	La stratégie de test doit se concentrer sur la manière de détecter les anomalies les plus importantes le plus tôt possible et au moindre coût. La stratégie de test précise de quelle manière on couvre les besoins et on maîtrise les risques. La qualité globale de la stratégie de test est d'autant meilleure que chaque étape de test possède sa propre stratégie adaptée à son objectif et bien différenciée de la stratégie des autres étapes de test.
2	Modèle de cycle de vie	Le processus de test est composé de plusieurs phases : la planification, la préparation, les spécifications, l'exécution et le bilan. Chaque phase possède ses propres activités définies par un objectif, des données en entrée, un processus, des données en sortie, des dépendances, des techniques et des outils, des moyens logistiques, de la documentation, etc. L'intérêt d'un modèle de cycle de vie est d'augmenter la maîtrise du processus de test grâce à la planification et au suivi des différentes activités.
3	Début du processus de test	Bien que l'exécution des tests commence normalement après la réalisation du logiciel, le processus de test peut et doit commencer plus tôt. Commencer plus tôt dans le cycle de vie d'un projet signifie trouver les anomalies plus tôt et plus facilement, voire prévenir les erreurs. Cela permet de réduire au minimum le temps pendant lequel les tests se trouvent sur le chemin critique du projet.
4	Estimation et planification	La planification et l'estimation des tests déterminent à quel moment se déroulent les activités, et les ressources nécessaires à leur accomplissement.
5	Techniques de spécification de test	Une technique de spécification de test est un moyen standard de déduire des cas de test à partir des informations adéquates. L'utilisation de techniques de test permet de connaître la qualité et la profondeur des tests, et améliore la réutilisabilité des tests.
6	Techniques de test statique	Il y a des choses qu'on ne peut tester de manière dynamique, c'est-à-dire en faisant tourner les programmes. On appelle tests statiques les revues de produits (sans exécution de programme) ou les relevés de mesures qui permettent d'atteindre un certain niveau de qualité. A cet effet, on utilise des listes de contrôle.

	Secteur-clef	Description
7	Métriques	On appelle métriques des observations mesurables des caractéristiques d'un produit ou d'un processus. Dans le cas d'un processus de test, la mesure de l'avancement du processus et de la qualité du logiciel testé est très importante. On s'en sert pour contrôler le processus de test, objectiver le jugement porté sur un logiciel testé et permettre des comparaisons entre des logiciels ou des processus différents : pourquoi tel logiciel fonctionne-t-il avec beaucoup moins d'incidents que tel autre ? pourquoi ce processus est-il plus efficace que tel autre ? Quand on veut améliorer un processus de test, les métriques se révèlent d'une grande utilité parce qu'elles permettent de mesurer les conséquences des actions d'amélioration en comparant les données objectives des situations a priori et a posteriori.
8	Outillage des tests	L'outillage des tests revêt plusieurs aspects et vise plusieurs objectifs : réduction des charges de test, réduction des délais de test, amélioration de la couverture des tests, amélioration de la flexibilité des tests, meilleure appréciation de l'état du processus de test, meilleure motivation des testeurs par la mécanisation des tâches ingrates ou répétitives.
9	Environnement de test	L'exécution des tests se déroule dans un environnement de test. Cet environnement est généralement composé des éléments suivants : des machines, des logiciels, des moyens de communication, des utilitaires pour alimenter et utiliser des bases de données et des fichiers, des procédures. L'environnement doit permettre de savoir dans quelle mesure le logiciel testé répond aux besoins. L'environnement a une grande importance sur la qualité, la durée et le coût du processus de test. Les principales caractéristiques d'un environnement sont la responsabilité, la gestion, la disponibilité, la représentativité et la flexibilité.
10	Environnement de travail	L'équipe de test a besoin d'espace, de bureaux, de chaises, de postes de travail et de moyens techniques (traitement de texte, imprimantes, téléphone, etc.). Une bonne organisation de l'environnement de travail a un effet très favorable sur la motivation de l'équipe de test, sur la communication interne et externe, et sur l'efficacité du travail.
11	Engagement et motivation	L'engagement et la motivation des personnes impliquées dans le test sont des conditions essentielles au bon fonctionnement d'un processus de test. Les personnes impliquées ne sont pas seulement les testeurs mais aussi, par exemple, l'encadrement du projet et la ligne hiérarchique. Ceux-ci sont particulièrement importants pour créer de bonnes conditions. Cela signifie que le processus de test est doté d'un délai, d'un budget et de ressources suffisants, en qualité comme en quantité, pour atteindre les objectifs fixés.
12	Fonctions de test et formation	La qualité de la composition d'une équipe de test est très importante. Une équipe de test est un mélange de métiers, fonctions, savoir-faire et compétences. Au delà du savoir-faire particulier en matière de test, la connaissance fonctionnelle, la maîtrise de l'organisation et des connaissances informatiques générales sont nécessaires, ainsi que des compétences relationnelles. Pour acquérir tout cela, il faut de la formation.

	Secteur-clef	Description
13	Périmètre méthodologique	Chaque processus de test recourt à une certaine méthode de travail comprenant des activités, des procédures, des règles, des techniques, etc. Quand les méthodes de travail diffèrent d'un projet à l'autre ou que la méthode est tellement générique qu'elle doit être redéfinie à chaque fois, il en résulte une perte d'efficacité du processus de test. L'objectif de l'organisation est d'utiliser une méthode à la fois suffisamment générique pour être applicable en toute situation et suffisamment détaillée pour qu'on n'ait pas besoin de la repenser à chaque fois.
14	Communication	Dans un processus de test, la communication doit s'établir aussi bien au sein de l'équipe de test qu'avec les autres acteurs concernés, tels les développeurs, les utilisateurs, le client... La communication n'est pas seulement nécessaire à la fluidité du processus de test, mais aussi pour rendre compte de l'avancement et de la qualité des travaux en cours.
15	Reporting	Le test n'est pas tant le fait de trouver des anomalies que de déterminer le niveau de qualité du logiciel testé. Le reporting doit être construit de manière à donner au client un avis argumenté sur la qualité du logiciel testé et même sur le processus de développement.
16	Gestion des anomalies	Bien que la gestion des anomalies incombe plus au projet lui-même qu'aux testeurs, ce sont néanmoins les testeurs qui sont les plus impliqués dans cette tâche. Une bonne gestion des anomalies doit non seulement permettre de suivre le cycle de traitement d'une anomalie mais aussi de suivre les évolutions du logiciel vers la qualité via les anomalies trouvées, cela afin d'émettre un avis autorisé sur la qualité.
17	Gestion des dossiers de test	Les produits du test doivent être « maintenables » et réutilisables, autant dire qu'il faut les administrer. En marge des produits des tests eux-mêmes, tels que les plans de test, les spécifications, les bases et les fichiers de données, il est important que les produits des étapes précédentes (conception fonctionnelle, réalisation) soient également bien administrés, car le processus de test peut se trouver interrompu si on livre de mauvaises versions de programmes, etc. Si les testeurs demandent qu'une administration des versions soit mise en place pour ces produits, cela ne pourra qu'avoir une influence bénéfique pour le projet et la « testabilité » du logiciel s'en trouvera accrue.
18	Gestion du processus de test	Pour administrer chaque processus et chaque activité, les 4 étapes du cercle de Deming sont essentielles : planification, exécution, vérification et action. L'administration du processus est d'une importance vitale pour la réalisation d'un test optimal dans un processus de test souvent turbulent.
19	Évaluation	L'évaluation désigne la revue des produits intermédiaires tels que l'expression de besoins et la conception fonctionnelle. L'évaluation présente l'avantage de permettre la détection des anomalies beaucoup plus tôt dans le processus de développement, ce qui réduit singulièrement les coûts de correction des anomalies. De plus, il est plus facile de mettre en œuvre une évaluation, car cela ne requiert aucun moyen technique (pas de mise en place d'un environnement ni de programme à faire tourner).

Améliorer le processus de test pour atteindre le niveau 1 de TPI

	Secteur-clef	Description
20	Tests en amont	Les tests en amont sont uniquement effectués par les développeurs. Les plus connus des tests en amont sont les tests unitaires et les tests d'intégration. Comme pour l'évaluation, les tests en amont permettent la détection des anomalies beaucoup plus tôt dans le processus de développement que les tests en aval. Les tests en amont sont très efficaces car ils requièrent peu de communication et parce que le détecteur de l'anomalie est aussi celui qui l'a produite et qui la corrige.

7.2. Les modèles de document

Le modèle de bilan des tests

BILAN DES TESTS				 D2T POWERTRAIN ENGINEERING			
Affaire N° :		0		Chef de projet :		0	
Objet Affaire :		0		Version :		0	
Famille :		0		Lot :		0	
Equipe projet :		0					

Couverture et écarts

formalisme	nb prévu	nb réalisé
Informel	0	0
Formel	0	0
non testé	0	0
semi-formel	0	0
non testable	0	0

Ecart des tests			
	niveau	nombre	temps
Souhaité :	Bon	0	0
Obtenu :		0	0

Couverture prévue



Couverture réalisée



REMARKS

01/01/2012

02/01/2012

03/01/2012

04/01/2012

05/01/2012

Raisons des écarts

	BILAN DES TESTS	 POWERTEAM ENGINEERING
--	------------------------	--

Anomalies

Nombre détectées :

		Sévérité			
		haute	normale	basse	aucune
Type	Bloquant				
	Non bloquant				
	Amélioration				
	Total				

Nombre Corrigées :	
Nombre non corrigées :	

Justification de la non correction des anomalies

Id anomalie	Description anomalie	Type	Justification de la raison de non correction

	BILAN DES TESTS	 POWERTRAIN AND SERVICES
--	------------------------	--

Budget

Coût du test (tps test / tps projet)

	Coût projet (en hrs)	Coût phase de test (en hrs)	$ratio = \frac{hrs_phase_de_test}{hrs_projet}$
Prévu			
Réalisé			
Ecart			

Ecart par rapport au plan

Phase	Durée prévue (en hrs)	Durée réalisée (en hrs)	Ecart (en %)
Analyse des spécifications			
Rédaction du plan de test			
Exécution des tests			
Préparation de la plateforme de test			
Confirmation correction constat			
Bilan			

Cette liste ne se veut pas exhaustive et est uniquement fournie à titre d'exemple.

Raisons des écarts

	BILAN DES TESTS	 ROWESTEIN ENGINEERING
--	------------------------	--

Retour d'experience

Problèmes rencontrés

--

Remarques

--

Le modèle de référentiel

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7		Affaire N° :		Chargé de projet :		Équipe projet :	
8		Objet Affaire :		Version :			
9		Famille :		Lot :			
10							
11		Qui	commentaires et modifications				Quand
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	

Le modèle de fiche de description de test

XX XXX XXXX,X
Le 08/04/2010

Fiche de description de test



D2T
POWERTRAIN ENGINEERING

Familie :		Produit :	
Fonction testée :			
Référence :			

Historique

N° version	Date	Commentaires	Auteur

Cette fiche de description de test est composée de : (indiquer le nombre de variantes contenu dans la FDT)

Nombre de scénarii nominaux :
Nombre de variantes :

Description du test :
Indiquer dans cette section l'objectif du test et un rappel succinct de la fonction testée.

Données de test :
Indiquer dans cette section les données utilisées pour la réalisation de ce test (fichier, valeur numérique ou chaîne de caractères,...).

Ressources intervenants :
Indiquer dans cette section toutes les ressources (matériel, logiciel, humaine) qui interviennent dans le test.

Pré conditions :
Indiquer dans cette section les opérations qu'il est nécessaire de réaliser avant de pouvoir exécuter le scénario.

Scénario :

Étape	Action intervenant	Réaction attendue du système (optionnel)

Post-conditions (optionnel) :
Indiquer dans cette section les opérations qu'il est nécessaire de réaliser à l'issue de l'exécution du scénario. Par exemple, lorsqu'un fichier de donnée est modifié lors du scénario, indiquer les opérations à réaliser afin de restaurer les données à l'état initial.

Résultats attendus
Indiquer dans cette section le ou les résultat(s) attendu(s), à l'issue du scénario, qui permet de déclarer que le test est passant ou non.

Rédigé par groupe de travail PRD
Vérifié par

Page 1/1



Fiche de reporting

XX XXX XXXX
Le 08 xxx, 2010

Fiche de reporting de test



PROJET

Projet :		N° affaire :	
Chef de projet :		Testeur :	
Version logiciel :			

Date du reporting :	
---------------------	--

REPORTING DES TESTS

État de l'activité de test :			
Nombre de tests réalisés :		Budget consommé :	
Nombre de tests total :		Budget total :	

Test	Fonction	Commentaire et/ou observation	Résultats (OK/NOK)	N° constat

Commentaires :



Rédigé par : groupe de travail PRD
Vérifié par :

Page 1/1



7.3. Matrice de traçabilité des exigences

Identificateur	Type	n°	K	Exigences	Résultat livré	PP1	PP2	PP3	PP4	PP5	PP6
TPI	1			Etablir et maintenir une stratégie de test			1		1		1
TPI	1			Etablir et maintenir une stratégie pour les tests en aval			1		1		1
TPI	1.1			Identifier les risques en fonction des exigences			1				
TPI	1.2			Adapter les tests en fonction du risque			1				
TPI	1.3			Choisir la technique en fonction du risque					1		
TPI	1.4			Identifier le type de réexécution en cas de retest							1
TPI	2			Etablir et maintenir un modèle de cycle de vie		1			1	1	1
TPI	2			Planification, Description, Exécution		1			1	1	1
TPI	2.1			Rédiger un plan de test		1				1	1
TPI	2.1.1			Définir les critères qui conditionnent le lancement des tests	chapitre 5.1 dans le plan de test et documenté dans le	1					
TPI	2.1.2			Déterminer le référentiel de tests	Chapitre 4 du plan de test	1					
TPI	2.1.2.1			Etablir la liste des tests	Colonne B du référentiel	1					
TPI	2.1.2.2			Identifier de manière unique	Colonne A du référentiel	1					
TPI	2.1.2.3			Traçable	doc_processus.doc	1					
TPI	2.1.3			Déterminer la stratégie de test	Définir le niveau de test attendu	1					
TPI	2.1.4			Mettre en place l'organisation						1	
TPI	2.1.5			Mettre en place les livrables de test						1	
TPI	2.1.6			Définir l'infrastructure et les outils							1
TPI	2.1.7			Mettre en place le management						1	
TPI	2.1.8			Planifier	modèle Plan test projet v0.1.doc	1					
TPI	2.3			Etablir une description des tests					1		1
TPI	2.3.1			Concevoir les cas de test							1
TPI	2.3.2			Concevoir les scripts de test					1		
TPI	2.3.3			Préciser le mode de prise en charge du logiciel à tester et de							1
TPI	2.3.4			Effectuer des tests d'infrastructure							1
TPI	2.4			Exécution des tests		1					1
TPI	2.4.1			Prendre en charge le logiciel à tester et l'infrastructure							1
TPI	2.4.2			Mettre en place les bases de données initiales							1
TPI	2.4.3			Exécuter les tests	1.modèle FDT numTest nomPrd numVersion.doc	1					
TPI	2.4.4			Réexécuter les tests							1
TPI	2.5			Etablir un bilan des tests		1				1	1
TPI	2.5.1			Procéder à une évaluation du logiciel testé						1	
TPI	2.5.2			Procéder à une évaluation du processus de tests						1	
TPI	2.5.3			Formuler des propositions							1
TPI	2.5.4			Rendre compte	modèle Bilan test projet v0.2.doc	1					
TPI	5			Etablir des techniques de spécification de test				1			
TPI	5			Techniques informelles				1			
TPI	5.1			Documenter les techniques de construction de cas de test				1			
TPI	5.1.1			Déterminer la situation initiale				1			
TPI	5.1.2			Définir le processus de changement = les actions de tests à				1			
TPI	5.1.3			Identifier le résultat attendu				1			
TPI	11			Etablir l'engagement et la motivation		1		1		1	1
TPI	11			Affectation de budget et de temps		1		1		1	1
TPI	11.1			Le personnel de tests considère les tests comme une chose	Echelle de niveaux de test V2.0.pdf	1					
TPI	11.2			Affecter aux tests des ressources disponibles et un budget	1. Profils de testeurs V0.2.doc	1					
TPI	11.3			Piloter les tests par le budget et les ressources						1	
TPI	11.4			Faire monter l'équipe de test en compétences				1			
TPI	11.5			Pour la plupart des membres de l'équipe de test, les tâches de							1
TPI	11.6			Les testeurs entretiennent de bonnes relations avec les autres							1
TPI	15			Etablir du reporting		1					
TPI	15			Anomalies		1					
TPI	15.1			Faire des compte-rendu périodique	modèle Fiche reporting projet jlmmaaaa v0.1.doc	1					
TPI	16			Gérer les anomalies		1	1				
TPI	16			Gérer en interne les anomalies		1	1				
TPI	16.1			Gérer les étapes de recherche d'anomalies			1				
TPI	16.2			Enregistrer les anomalies	1. retiré du projet voir	1					
TPI	18			Gérer le processus de test						1	
TPI	18			Planification et exécution						1	
TPI	18.1			Dresser un plan de test						1	
TPI	18.1.1			Préciser toutes les tâches à effectuer						1	
TPI	18.1.2			Indiquer la période sur laquelle chaque tâche s'étend						1	
TPI	18.1.3			Préciser les ressources requises (personnes, matériel...) pour						1	
TPI	18.1.4			Préciser les produits résultants de la tâche						1	

Améliorer le processus de test pour atteindre le niveau 1 de TPI

Identificateur	Type	n°	R	Exigences	Résultat livré	P1	P2	P3	P4	P5	P6
REQM	1			Obtenir une compréhension des exigences		1	1	1	1	1	
REQM	1.1			Etablir des critères pour distinguer des fournisseurs		1	1	1		1	
REQM	1.1.1			Etablir des critères pour l'évaluation des exigences.							
REQM	1.1.2			Etablir des critères pour l'acceptation des exigences.							
REQM	1.1.3			Etablir un référentiel des exigences comprise et accepté par	modèle_Ref_exig_projet_v0.1.xls	1					
REQM	1.1.3.1			découpage de la spec en fonction	Colonne B du référentiel	1					
REQM	1.1.3.2			définir un identificateur unique	Colonne A du référentiel	1					
REQM	1.1.3.3			définir une priorité selon les besoins client				1			
REQM	1.1.3.4			définir une criticité				1			
REQM	1.1.3.5			définir une priorité selon les risques			1				
REQM	1.1.3.6			identifier les sources desquelles proviennent les exigences				1			
REQM	1.1.3.8			placer les exigences sous contrôle (référentiel des exigences)						1	
REQM	1.1.3.9			documenter les exigences non techniques affectant et						1	
REQM	1.2			Etablir des critères objectifs pour l'évaluation et l'acceptation		1	1	1	1	1	
REQM	1.2.1			Enoncer les critères clairement / exacte	Doc_processus.doc	1					
REQM	1.2.2			Enoncer les critères correctement / non ambiguë	Doc_processus.doc	1			1		
REQM	1.2.2.1			Définir la capacité du système à résoudre un problème ou					1		
REQM	1.2.2.2			Définir les contraintes limitatives relatives à la capacité					1		
REQM	1.2.3			Obtenir un énoncé complet					1		
REQM	1.2.4			Enoncer les critères de manière non contradictoires /	Doc_processus.doc	1					
REQM	1.2.5			identifier de façon unique /cohérente	doc_processus.doc §2.3.3 et 2.3.4	1					
REQM	1.2.6			Appropriés à mettre en œuvre			1	1			
REQM	1.2.7			Définir des critères vérifiables (testables)				1			
REQM	1.2.8			Définir des critères traçables		1					
REQM	1.3			Analyser les exigences pour vérifier qu'elles correspondent						1	
REQM	1.4			Obtenir une compréhension partagée avec les fournisseurs		1					
REQM	1.4.1			Prévoir un délai de lecture pour la spec	doc_processus.doc chapitre délai de lecture	1					
REQM	1.4.2			Résoudre les problèmes de compréhension, par les parties	méthode de gestion des commentaire avant revue voir	1					
REQM	2			Obtenir l'engagement sur les exigences						1	
REQM	2.1			Evaluer l'impact des exigences sur les engagements						1	
REQM	2.2			Négocier et enregistrer les engagements.						1	
REQM	3			Gérer les modifications aux exigences					1		
REQM	3.1			Documenter toutes les exigences et toutes les modifications					1		
REQM	3.2			Maintenir l'historique des modifications aux exigences, avec					1		
REQM	3.3			Evaluer l'impact des modifications aux exigences du point de					1		
REQM	3.4			Mettre les données sur les exigences et les modifications à la					1		
REQM	4			Maintenir la traçabilité bidirectionnelle des exigences		1			1	1	
REQM	4.1			Maintenir la traçabilité des exigences pour assurer que la							
REQM	4.2			Maintenir la traçabilité entre une exigence et ses exigences		1			1	1	
REQM	4.3			Générer une matrice de traçabilité		1			1		
REQM	4.3.1			Suivre les exigences	1. Onglet gestion des fonctions	1					
REQM	4.3.2			Mesurer l'état de chacune des exigences allouées	1. Modèle de tabl. Fonct. - Onglet gestion des fonctions				1		
REQM	4.3.3			Mesurer les changements des exigences allouées					1		
REQM	4.3.4			Tracer les exigences	1. Onglet gestion des fonctions	1					
REQM	5			Institutionnaliser le processus							1
REQM	5.1	6		Etablir et maintenir une directive organisationnelle traitant de la							1
REQM	5.2	6		Etablir et maintenir le plan pour la mise en œuvre du							1
REQM	5.3			Fournir les ressources adéquates pour la mise en œuvre du							1
REQM	5.4			Assigner les responsabilités et l'autorité pour la mise en							1
REQM	5.5			Former selon les besoins les personnes qui mettent en œuvre							1
REQM	5.6			Gérer en configuration, au niveau approprié, les produits de							1
REQM	5.7			Identifier et impliquer, tel que planifié, les parties prenantes							1
REQM	5.8			Suivre et contrôler le processus vis-à-vis son plan de mise en							1
REQM	5.9			Evaluer de manière objective la conformité du processus à sa							1
REQM	5.10			Passer en revue, avec la hiérarchie, les activités, le statut et							1

7.4. Tableau des risques projet

Id.	Intitulés des risques	Conséquences	P	I	G	Actions			
						Préventif	Coûts (€)	Corrective	Coûts(€)
Gestion de projet									
R1	Conduite du changement	Non application du processus	1	5	5	coatching	3500		
R2	indisponibilité du chef de projet	arrêt du projet	1	5	5	CdP adjoint	17000		
Groupe de travail									
R3	manque de motivation du groupe	Pas de solution ou solution inappropriée	2	4	8	Intervention resp. PRD	100		
Projet Pilote									
R4	Pas de projet de développement approprié	retard projet	2	2	4				
R5	L'équipe n'est pas motivée	solution inappropriée	2	4	8				
R6	Mauvais choix de répartition des exigences	solution inappropriée	3	2	6				
Projets de Validation									
R7	Pas de projet de développement approprié	retard projet	2	2	4				
R8	L'équipe n'est pas motivée	évaluation impossible	2	4	8				

8. Glossaire

AMO	Assistance à la Maîtrise d’Ouvrage
BEI :	Bureau d’Étude Informatique est le service du département Produits qui est en charge du développement des logiciels
Brainstorming:	Outil de créativité libre et ordonnée qui permet de rechercher en groupe et en toute liberté un maximum d'idées sur un sujet donné ou d'inventer des solutions pour résoudre un problème
Cas de test:	ensemble des jeux de données (valeurs en entrée et sortie), des scénarii et des résultats attendus, utilisé pour la réalisation d'un test
CFTL:	Comité Français des Tests Logiciels
CMMI:	Capability Maturity Model Integration
ECU	Engine Control Unit ou unité de contrôle du moteur est un calculateur qui contrôle le fonctionnement du moteur
EDI:	Environnement de Développement Intégré
IEME:	Ingénierie des Équipements des Moyens d’Essais
Effet tunnel:	laps de temps écoulé entre la formulation des besoins d'utilisateurs concernant un futur système, et l'aboutissement dans sa conception à l'issue duquel ces premiers peuvent évaluer le travail effectué sans avoir eu de droit de regard durant sa période de développement.
Élicitation des exigences	activité consistant à rassembler, traiter et suivre les besoins servant de base à la définition du produit [7]
IGMP:	Ingénierie des Groupes MotoPropulseurs
Indicateurs de réalisation de pratique:	traces laissées lors de la mise en œuvre d’une pratique
ISO 12207:	norme définie par l’Organisation Internationale de Normalisation qui recense et décrit les processus couramment mis en œuvre dans les projets informatiques
ISTQB:	International Software Testing Qualification Board
MOA:	maîtrise d’ouvrage
MOE:	maîtrise d’œuvre
Parties prenantes:	toute personne ou organisation ayant un intérêt au projet
PdT:	politique de test
PMBOK:	Project Management Body of Knowledge
PMI:	Project Management Institute
PP:	projet pilote
PRD	Service Produits faisant office de MOA au sein du département Produits
PV:	Projet de validation

Améliorer le processus de test pour atteindre le niveau 1 de TPI

REQM:	pour REQuirements Management est un domaine de processus de CMMI dont l'intention est de gérer les exigences des produits et composants de produits du projet [21].
Risque produit	est un risque qui porte sur les exigences fonctionnelles du produit
RP:	responsables produits
SC:	secteurs-clef du modèle TPI
SCAMPI:	Standard CMMI Method for Process Improvement
SMQ:	Système de Management de la Qualité
Test de confirmation:	test qui exécute des cas de test qui ont été en échec la dernière fois qu'ils furent exécutés, de façon à vérifier le succès des actions de corrections
Test en aval:	tests réalisés lors de la branche montante du cycle de vie de développement en V,
TPI:	Test Process Improvement
V&V	Vérification et Validation

9. Références

9.1. Bibliographie

- [1] ISO/IEC 12207-2008 Technologies de l'information - Processus du cycle de vie du logiciel - 2008
- [2] Industrialiser le test fonctionnel – B. LEGEARD ; F. BOUQUET ; N. PICKAERT – Éditions DUNOD – 2009
- [3] Le test des logiciels – S.XANTHAKIS ; P.REGNIER ; C.KARAOULIOS – Éditions Hermes – 2000
- [4] Le génie logiciel – J. PRINTZ – Éditions PUF – 1995
- [5] IEEE 830 - Recommended Practice for Software Requirements Specifications - Éditions IEEE-1998
- [6] TMap Next – T. KOOMEN ; L. VAN DER AALST ; B. BROEKMAN ; M. VROOM – Éditions Uitgeverij Tutein Nolthenius – 2008
- [7] Processus d'ingénierie du logiciel - C. PINET - Éditions Pearson Education – 2002
- [8] Test process Improvement A practical step-by-step guide to structured testing - T. KOOMEN ; M. POL - Éditions Addison-Wesley - 1999
- [9] The art of software testing (second edition) - G.J. MYERS – Editions John Wiley & sons – 2004
- [10] Systematic Software Testing – R. D. CRAIG ; S. P. JASKIEL - Éditions Artech House – 2002
- [11] Qualimétrie des systèmes complexes – T. FORSE – Les éditions d'organisation – 1989
- [12] IEEE 610.12 Standard Glossary of Software Engineering Terminology - IEEE - 1990
- [13] Standard Glossary of terms used in software Testing V2.1 – ISTQB – 2010
- [14] Pratique de tests logiciels – J.F. PRADAT-PEYRE; J.PRINTZ - Editions DUNOD -2009
- [15] Syllabus Niveau Fondation - ISTQB – 2010
- [16] IEEE 829 Standard for Software and System Test Documentation - IEEE – 2008
- [17] Corpus de connaissances en management de projet - Project Management Institute – 2004
- [18] Maîtrise d'oeuvre des projets informatiques – A. DURAND – Editions DUNOD - 2004
- [19] Vade-mecum du chef de projet – PRINTZ CONSEIL – 1999
- [20] Rédiger des cas d'utilisation efficaces – A COCKBURN – 2001
- [21] Un itinéraire fléché vers CMMI – R. Basque – Edition DUNOD 2nd éd. -2006
- [22] Évaluer avec CMMI étape par étape – M. Lamnabhi – Édition Afnor 2008

- [23] CMMI for Development Version 1.2 Août 2006
- [24] IEEE 1012 - Standard for Software Verification and Validation – IEEE - 1998

9.2. Webographie

- [25] <http://www.cftl.net/>
- [26] <http://www.tmmifoundation.org/>
- [27] <http://www.chillarege.com/articles/odc-concept>
- [28] <http://eng.tmap.net/>
- [29] <http://www.rbc-us.com/software-quality-services/consulting>
- [30] <http://fr.groups.yahoo.com/group/CMMI-en-francais/>