

# Projet

## Combat pour la Vie – Stratégie de test

Référence : Test-CPLV-0C  
Fournisseur  
Date : 10/01/2019  
Version/Édition : 0C  
État : Préliminaire

Type de diffusion : Restreinte  
Autre référence :

### SIGNATURES

	Fonction	Nom	Date	Visa
Auteur	Binôme	Martins Thomas Duret Romain	03/01/2019	
Validé par				
Vérifié par				
Approuvé par				

### AUTORISATIONS CLIENT

	Fonction	Nom	Date	Visa
Approuvé par				
Approuvé par				

**Diffusion Restreinte**

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## Historique des révisions

Date	Description et justification de la modification	Pages / Chapitre	Edition / Révision
03/01/2019	Création	Toutes	0A
08/01/2019	Corrections	Toutes	0B
10/01/2019	Corrections retour Mr Yves Jehanno	Toutes	0C

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## Liste de diffusion interne

Nom	Fonction	Action	Date	Nb exemplaire(s)

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## Table des matières

Historique des révisions.....	2
Liste de diffusion interne .....	3
Table des matières .....	4
1 Introduction.....	6
1.1 Objectif du document .....	6
1.2 Présentation du logiciel.....	6
1.3 Hypothèses et risques .....	7
1.4 Glossaire et abréviations.....	7
2 Organisation de la validation .....	9
2.1 Rôles et responsabilités.....	9
2.2 Documents de tests .....	10
2.3 Environnement de test .....	11
2.3.1 Environnements matériel et logiciel du site central.....	11
3 Ressources .....	12
3.1 Ressources humaines .....	12
3.2 Ressources matérielles.....	13
4 Stratégie de test.....	14
4.1 Stratégie de validation .....	14
4.1.1 Principe .....	14
4.1.2 Organisation de la stratégie .....	14
4.1.3 Objectif de test.....	15
4.1.3.1 Validation des règles de gestion .....	15
4.1.3.2 Validation des règles de calcul .....	15
4.1.4 Niveaux de criticité.....	16
4.1.5 Critères d'arrêt des tests .....	16
4.1.5.1 Normal.....	16
4.1.5.2 Anormal.....	16
4.2 Circuit des modifications.....	18
5 Description des tests .....	19
5.1 Le cycle de développement .....	19
5.2 Validation des projets .....	19
5.2.1 Dossier de tests unitaire .....	19
5.2.2 Dossier de tests d'intégration.....	19
5.2.3 Dossier de tests de validation .....	20
5.2.3.1 Tests de charges .....	20
5.2.3.2 Simulation des utilisateurs .....	20
5.3 Cas des projets en langage objets .....	20
5.3.1 Stratégie issue de l'UML .....	21

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

6	Garantir l'iso périmètre.....	22
7	Choix d'outil.....	23

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

# 1 Introduction

## 1.1 Objectif du document

Ce document a pour objectif de décrire la méthodologie Fournisseur concernant le test et la validation d'application. Il aborde successivement l'organisation, la planification des tests et la préparation des campagnes de tests à effectuer.

Il aborde les points suivants:

- l'organisation du projet de test validation, rôles et responsabilités des intervenants dans le processus de test,
- la communication entre les intervenants responsables des tâches à effectuer ainsi que leur planification,
- les activités de préparation et de conduite des tests,
- la stratégie de test de validation appliquée au projet,
- les outils et les environnements dans lesquels les tests doivent s'exécuter,
- les niveaux de criticité et les critères d'arrêt des tests,
- le cycle de prise en compte des anomalies rencontrées dans l'exécution des tests ainsi que leur traitement,
- le mode d'archivage des éléments issus des étapes de test.

La définition de ces activités doit permettre d'automatiser les campagnes de tests afin d'obtenir les éléments indispensables à l'exécution des tests de non régression lors des phases d'évolutions et de corrections de l'application.

Ce document est applicable sur les modules entrant dans le cadre du développement de l'application à développer.

## 1.2 Présentation du logiciel

Pour tester efficacement une application, il est important de comprendre ses fonctionnalités. C'est pourquoi, une présentation de l'application est imposée dans le plan de test.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

### 1.3 Hypothèses et risques

Chaque application comporte des hypothèses sur la fiabilité et sur le risque de glissement du projet. Il s'agit de définir l'ensemble des incidents pouvant survenir pendant la phase de test.

Afin de contrôler ces risques, une procédure de traitement des anomalies est mise en place permettant la prise en compte immédiate du problème et la détermination d'une solution optimisée. On peut également réduire ces problèmes en utilisant les maquettes d'écran pour fiabiliser les différents scripts de tests.

### 1.4 Glossaire et abréviations.

#### Abréviations

DTV : Dossier de Test Validation  
 PTV : Plan de Test Validation  
 T&V : Test et Validation

#### Glossaire

##### **Campagne de Test :**

Activité qui consiste à dérouler un ensemble de jeux de test. Un dossier de test est produit à l'issue d'une campagne.

##### **Cas de Test :**

Déclinaison d'un test précisant les valeurs utilisées pour les variables du test ainsi que les résultats attendus.

##### **Dossier de Test :**

Ensemble documentaire qui contient la description des scénarios et cas de tests, ainsi que l'exécution des jeux de test. Le dossier de test est le reflet d'une campagne de test.

##### **Jeux de Test :**

Ensemble de scénarios et cas de tests permettant de tester un produit logiciel. L'enchaînement des cas et scénarios de tests est relatif à une stratégie de test précisée dans le plan de test.

##### **Plan de Test (ou Protocole de Recette) :**

Document décrivant le déroulement d'un jeu de test : Stratégie de test, critères d'arrêt, planification.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

### **Scénarios de Tests :**

Ensemble de cas de tests cohérents permettant de traiter un objectif fonctionnel.

### **Test Fonctionnel :**

Test (vu de l'utilisateur) du bon fonctionnement d'un produit logiciel, d'une fonctionnalité ou d'une fonction de base. Vérification par rapport aux spécifications.

### **Test de Non Régression :**

Vérification qu'une nouvelle version du produit logiciel fonctionne sans dégradation (technique, fonctionnelle, performance) par rapport à la version précédente.

### **Test Unitaire :**

Vérification qu'un module, objet, ou autres composants d'un produit logiciel est conforme à ses spécifications (techniques et fonctionnelles). Basé sur des tests fonctionnels (vue de l'extérieur) et structurels (vue de l'intérieur).

### **Test ISO périmètre**

Vérification que la nouvelle application développée réalise et couvre à l'identique l'ensemble des fonctionnalités présente dans l'ancienne application

### **Test de Validation :**

Vérification que le produit logiciel est cohérent et complet par rapport aux spécifications fonctionnelles.



Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 2 Organisation de la validation

### 2.1 Rôles et responsabilités

On définit l'organisation humaine nécessaire pour la mise en place des tests du logiciel. Les responsabilités associées aux différents intervenants du projet par rapport aux activités de tests, sont définies dans le tableau suivant :

*Exemple:*

Responsabilités des Activités	Chef de projet Romain	Chef de projet Thomas
<b><u>Organisation:</u></b>		
Définition de l'organisation des tests	X	X
Réception des développements et contrôle des Tests Unitaires		
Livraison des lots		
<b><u>Spécification:</u></b>		
Définition des Scénarios de test		X
Définition des cas de test		X
Définition des données de test		
<b><u>Conception:</u></b>		
Création du Référentiel de Données (Base)		
Paramétrage environnement Application	X	
Paramétrage environnement outil de test		
Livraison maquettes écrans		
Création des scripts de test		
<b><u>Exécution:</u></b>		
Lancement des jeux d'essai		
<b><u>Analyser:</u></b>		
Analyse résultat		
Traitement des Anomalies		

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

<b>Responsabilités des Activités</b>	<b>Chef de projet Romain</b>	<b>Chef de projet Thomas</b>
Suivi des Anomalies		

Le responsable de projet est le garant des livraisons entre l'équipe de développement et l'équipe de validation

## 2.2 Documents de tests

Les documents suivants sont générés par l'équipe de validation et constituent le référentiel documentaire de test. Ils sont à nouveau appliqués dans les tests de non-régression lors de livraison de nouvelles versions de l'application.

Ces documents sont identifiés par :

- Plan de Test de Validation
- Dossier de Test de Validation.

Les documents sont édités à l'aide des outils bureautiques en vigueur chez Fournisseur.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

Les données de test générées par les activités sont identifiées par :

- copie de tous les fichiers générés par les outils : scénarios, scripts de test, fichiers de données.
- rapport de couverture lors de l'exécution des tests (avec ou sans outil).

Tous ces éléments doivent pouvoir être identifiés et liés à chaque version du logiciel. Un rapport de validation regroupe et lie toutes ces informations.

## 2.3 Environnement de test

On définit l'environnement de test devant être mis en œuvre : matériel et logiciel. La description précise de l'environnement est spécifiée pour l'ensemble de la campagne de test. En cas d'utilisation d'outil de test on spécifie les caractéristiques minimales nécessaires pour l'utilisation de l'outil.

### 2.3.1 Environnements matériel et logiciel du site central

On établit le tableau de description du site central.

Ordinateur	Version
Type :	Tour PC ou PC portable
Mémoire centrale :	8 Go
Disques :	500 Go ou 1 To
Système d'exploitation :	Windows 10 ou Ubuntu 18
<b>Logiciels de communications :</b>	
Réseau	Local
Protocole de communication	Aucun
<b>Environnement logiciel pour l'application :</b>	
- Eclipse	
<b>Application(s) partie serveur</b>	Aucun

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

### 3 Ressources

#### 3.1 Ressources humaines

Les ressources nécessaires à la définition et à l'application des tests de validation sont listées dans le tableau suivant :

Equipe	Profil	Nombre
Chef de projet	Binôme	2
Equipe de Développement	Binôme	2

Ces profils sont utilisés suivant la planification des phases de tests et en fonction de l'importance du projet.

Le tableau suivant indique pour chaque phase du projet les tâches allouées à chaque profil :

Tâche	Responsabilité	Effort (Jour)	Observations
Préparer le Plan de Test	Chefs de projet	A définir	
Spécification des Tests - Fonctions de gestion - Fonctions d'édition - fonctions interfaces	Chefs de projet	A définir	1 écran, n états, n traitements, 1 interfaces / jour sur la base de n écrans de niveau critique.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

L'effort alloué à la résolution des problèmes rencontrés lors de l'exécution des scénarios comprend :

- L'analyse du problème par l'équipe de validation
- La détermination des modifications à apporter par le chef de projet.

### 3.2 Ressources matérielles

Les ressources matérielles nécessaires à l'accomplissement des tâches sont définies dans le tableau suivant :

Tâche	Matériel	Observations
Préparer le Plan de Test	Ressources prestataire	Moyens bureautiques mis en oeuvre chez le prestataire
Spécification des Tests	Outils bureautiques Spécifications fonctionnelles détaillées	Dernière version des documents

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 4 Stratégie de test

### 4.1 Stratégie de validation

La stratégie de validation s'effectue en trois étapes :

- tests unitaires,
- tests d'intégration,
- tests de validation fonctionnelle.

#### 4.1.1 Principe

L'objectif des tests de validation est d'assurer la conformité fonctionnelle de chaque module et de ses interfaces par rapport aux spécifications fonctionnelles et techniques. Il s'agit de parcourir l'ensemble des chemins du module en utilisant des jeux de valeurs valides et invalides. Pour cela on applique des séries de scénarios correspondant aux modèles des traitements.

Les scénarios doivent décliner les cas de test suivant la criticité de la fonctionnalité ou de la règle de gestion mise en œuvre.

#### 4.1.2 Organisation de la stratégie

La stratégie de validation mise en place doit suivre les phases du développement (par lot), pour cette raison l'organisation est découpée en trois niveaux :

- 1er niveau : création des scénarios associés à chaque lot,
- 2ème niveau : intégration au fur et à mesure des lots,
- 3ème niveau : validation globale des lots.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

#### 4.1.3 Objectif de test

Les objectifs de test associés à la validation sont identifiés comme suit :

##### **Domaine de l'application**

Les tests couvrent les objectifs fonctionnels : vérification de la conformité de traitement des fonctions. Ce domaine fonctionnel ne couvre que les fonctions critiques (voir chapitre suivant)

Les cas de test seront définis comme suit :

- Les fonctions définies comme critiques sont testées pour le cas nominal de traitement.
- Un cas d'erreur est appliqué aux règles de gestion mises en œuvre dans ces fonctions.
- Suivant la criticité des règles de gestion (définie par CP), plusieurs cas de test sont appliqués.

##### **Domaine graphique**

Il n'y a pas d'objectif graphique, c'est à dire que les objets graphiques sont testés dans la mesure où ils sont utilisés dans l'enchaînement des écrans :

Pas de contrôle systématique sur chaque objet graphique sauf lorsque le client l'impose.

Pas de contrôle du respect des règles d'ergonomie s'il n'existe pas de charte d'ergonomie transmise par le client.

##### 4.1.3.1 Validation des règles de gestion

La stratégie de tests de conformité fonctionnelle appliquée sur l'application doit valider l'ensemble des règles de gestion définies pour l'application.

Pour chaque scénario, on prendra :

- des cas de tests correspondant au fonctionnement nominal (au moins 1 cas par classe d'équivalence),

Pour chacune de ces opérations les tests sur les saisies valides sont déclinés selon les cas de tests suivant :

- saisie avec les valeurs proposées dans les listes,
- des cas de tests aux limites de la plage de validité (au moins 1 cas par limite),

Ces tests devront être effectués dans chacun des contextes d'utilisations possibles (profils utilisateurs).

##### 4.1.3.2 Validation des règles de calcul

Toutes les règles de calcul seront testées avec des valeurs supérieures, inférieures et limites. L'ensemble des règles de calcul liées aux taux de TVA, aux montants multiples, aux sommes... doit être recensé et spécifié pour pouvoir être validé.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

#### 4.1.4 Niveaux de criticité

Trois niveaux de criticité sont définis : **Critique**, **Courant** et **Annexe**. Dans le cadre des tests de validation l'affectation des niveaux de criticité aux fonctions est choisie selon la fréquence d'utilisation des fonctionnalités et leur impact sur la base de données.

##### Critique

Ce niveau correspond aux fonctionnalités mettant à jour la base de données (les fonctions les plus critiques étant celles qui effectuent des mises à jour irréversibles) ou aux fonctionnalités majeures de l'application (ex : routage de flux, interface, fonction dont le dysfonctionnement bloque l'application, etc..)

##### Courant

Ce niveau correspond aux fonctionnalités les plus utilisées en interrogation ou visualisation.

##### Annexe

Ce niveau correspond aux fonctionnalités n'ayant pas d'impact sur le traitement des données et aux fonctionnalités peu utilisées. Ce niveau regroupe toutes les éditions, certaines fonctions de consultation, des fonctions d'extraction... "

#### 4.1.5 Critères d'arrêt des tests

##### 4.1.5.1 Normal

Le **critère d'arrêt normal** des tests (au niveau de chaque campagne) est défini par le respect des conditions suivantes :

- tous les cas de tests ont été déroulés,
- tous les résultats obtenus sont analysés et enregistrés,
- toutes les fiches d'anomalies sont soldées.

##### 4.1.5.2 Anormal

Le **critère d'arrêt anormal** d'une campagne intervient dans les cas suivants :

- ressources nécessaires à la campagne insuffisantes,
- déroulement non conforme,
- nombre d'anomalies détectées dans l'exécution trop important.

Le premier cas est contrôlé lors de la préparation de l'exécution des tests (ex: base de données non à jour, réseau non disponible,...)



Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

Les deux autres cas sont traités directement dans les scénarios de test par le contrôle et l'enregistrement des fichiers.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b> <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 4.2 Circuit des modifications

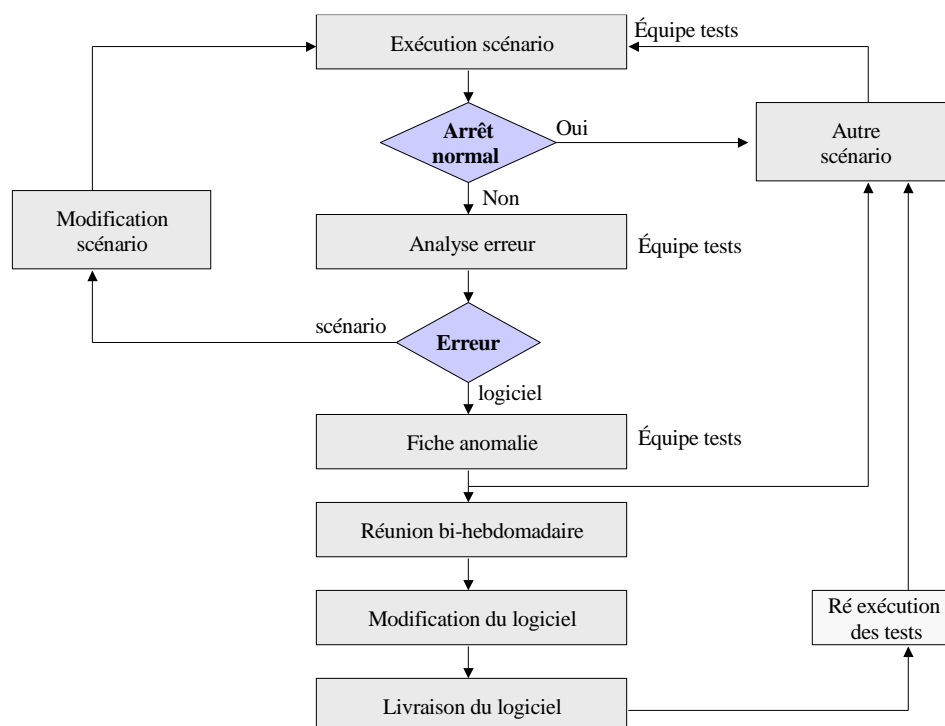
Le circuit des modifications peut intervenir lors de l'exécution de chaque phase d'activité de test :

- spécification,
- conception,
- exécution,
- analyse.

Il permet la prise en compte des anomalies rencontrées dans la phase de test impliquée :

- anomalie de spécification,
- anomalie de conception,
- anomalie d'exécution.

Le cycle de prise en compte est représenté par le schéma suivant :



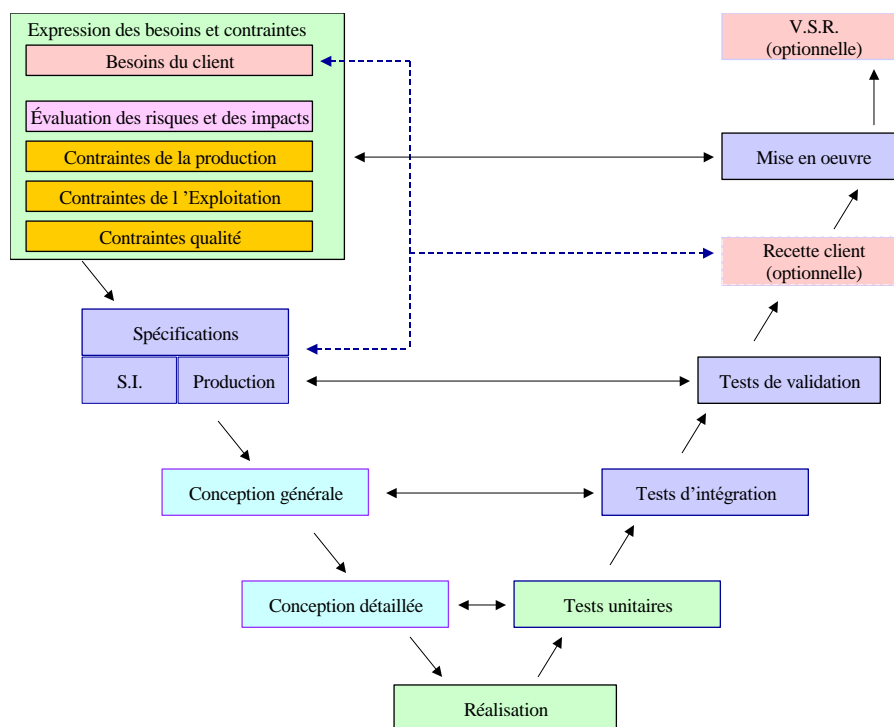
Le suivi de ces anomalies sera traité par le chef de projet de l'équipe de tests en collaboration avec le chef de projet de développement.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 5 Description des tests

### 5.1 Le cycle de développement

Le cycle de développement logiciel (cf. Plan de développement) en vigueur chez le prestataire est le suivant :



### 5.2 Validation des projets

#### 5.2.1 Dossier de tests unitaire

Le dossier de tests unitaires est initialisé dans la phase de conception détaillée et finalisé avant la phase de tests unitaires. Il contient l'ensemble des tests par module. En fonction de la complexité de l'application, on peut établir un dossier par lot fonctionnel. L'objectif est de valider la qualité du code et les performances de chacun des modules développés.

#### 5.2.2 Dossier de tests d'intégration

Le dossier de tests d'intégration est initialisé dans la phase de conception générale et finalisé avant la phase de tests d'intégration. Il contient l'ensemble des tests garantissant l'intégration des modules entre eux puis dans l'environnement définitif.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

### 5.2.3 Dossier de tests de validation

Le dossier de tests de validation est initialisé dans les phases de spécification et finalisé avant la phase de validation. On effectuera dans un premier temps les tests de stabilité pour vérifier les performances globales de l'application. Dans un second, on effectuera les tests fonctionnels correspondant aux besoins du client.

#### 5.2.3.1 Tests de charges

Les tests de charge sont le complément des tests fonctionnels. Ils ont pour objectifs la validation des applications et des serveurs dans un contexte Multi-Utilisateurs. La méthodologie Fournisseur se déroule en plusieurs étapes :

- Identification de l'architecture matérielle,
- Identification du panel d'utilisateur,
- Identification des tâches par utilisateur,
- Classification de la probabilité et du ratio d'utilisation,
- Détermination de la charge critique par scénario (accès concurrents)
- Détermination du tableau d'ordonnancement (charges, cycles et déclenchements).

#### 5.2.3.2 Simulation des utilisateurs

La simulation est réalisée soit par des automates de test (cas des gros projets) soit par des utilisateurs de l'application (petits projets).

### 5.3 Cas des projets en langage objets

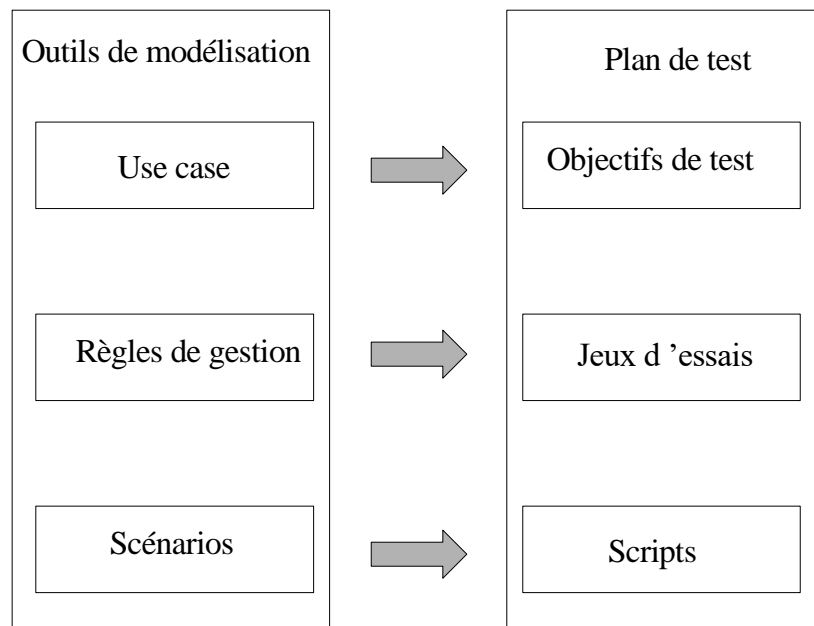
La validation des projets développés en langage objet diffère légèrement des projets en langage traditionnel. Ces tests doivent être abordés en fonctions des intégrations dans les applications. Chaque objet fait partie d'un environnement clairement identifié appelé service. La norme UML (en vigueur chez le prestataire) propose par défaut trois services principaux :

- Le service utilisateur (IHM),
- Le service métier (fonctionnel métier),
- Le service Base de données (accès base de données).

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	<b>Projet</b>  <b>Combat pour la Vie Stratégie de tests</b>	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

### 5.3.1 Stratégie issue de l'UML

Le schéma suivant s'applique aux développements basés sur l'UML :



Les use case sont définies dans la phase de spécification du système d'information. Ils constituent la meilleure façon de décrire les fonctions principales de l'application et leur interaction avec le monde extérieur. Ils décrivent les fonctionnalités de l'application qui auront été modélisées et réalisées. Lorsque le use case est trop vague pour être utilisé comme tel, on le découpe en objectifs secondaires rattachés à l'objectif principal. Ce découpage autorise la hiérarchisation des objectifs de test.

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 6 Garantir l'iso périmètre

Afin de garantir l'iso périmètre fonctionnel de la nouvelle application, il est important de maîtriser fonctionnellement l'ancienne application. Pour cela la méthodologie Fournisseur respecte plusieurs étapes incontournables :

- Application existante
  - Analyse de l'application existante,
  - Définition du dossier de validation de l'application existante (le dossier doit couvrir au minimum 80% de l'application et contiendra par test : données en entrées, description du traitement, résultats attendus)
  - Définition du jeu d'essai
  - Exécution du dossier de tests de validation sur l'application existante,
  - Enregistrement des observations et résultats constatés.
- Nouvelle application
  - Adaptation éventuellement du jeu d'essai définis précédemment pour la nouvelle application
  - Alimentation de la nouvelle application par le jeu d'essai,
  - Exécution du dossier de tests de validation définis pour l'ancienne application
  - Enregistrement des observations et résultats constatés
- Analyse
  - Analyse et comparaison des résultats observés sur les deux systèmes
  - Rédaction d'une fiche d'anomalie pour tous les écarts constatés et non justifiés.

Note No 1 : s'ils existent, les dossiers de tests rédigés pour la validation de l'ancienne application seront naturellement repris et adaptés pour la nouvelle application

Note No 2 : Le contrôle de la régression peut se faire soit manuellement soit au travers de l'utilisation d'un outil permettant l'automatisation des tests (dans ce cas les scénarii de test seront à écrire à partir de l'ancienne application pour pouvoir être rejoués sur la nouvelle application).

Ref : Test-CPLV-0C Emetteur : Romain Duret Thomas Martins Client : Sébastien Mavromatis Projet : Combat Pour La Vie	Projet  Combat pour la Vie Stratégie de tests	Date : 08/01/2019 Version : 0C Service : Ecole Etat : Préliminaire
---	---	---

## 7 Choix d'outil

En fonction du nombre de test à réaliser et du nombre de version successive de l'application, on peut décider la mise en place d'un outil de test automatique garantissant la non-régression de l'application. Ce choix est effectué au début du projet.

JUnit