PROJET VISHNU

DOCUMENTS DE REFERENCE

POLITIQUE

DE

TESTS

Ref. document VISHNU_D1_1e-POL-TESTS
Objet du document Référentiel de la politique de tests du projet EDF-Vishnu

TABLE DES REVISIONS

Version	Date	Modifications	Auteur
v0.1	04/01/11	Création du document	ISNARD Benjamin
v0.2	11/01/11	Modifications suite réunion tel.	ISNARD Benjamin
v0.3	12/01/11	Correction typo et traduction de termes normalisés	PERON Christophe
v0.4	13/01/11	Ajout de la partie organisation et planning	ISNARD Benjamin
v1.0	14/01/11	Finalisation	ISNARD Benjamin
v1.1	14/01/11	Corrections	ISNARD Benjamin

VALIDATION DU DOCUMENT

Date	Personne
14/01/11	Eddy Caron

SysFera
13 av. Einstein
69100 Villeurbanne

Siège Social
13 cité Jean-Baptiste Clément
69700 Givors

Tel. + 33 (0)4 36 42 15 30
contact@sysfera.com
www.sysfera.com

REFERENCES

Ref. document	Description

SysFera
46 allée d'Italie
69364 Lyon Cedex 07

Tel. + 33 (0)4 72 72 88 50
Fax + 33 (0)4 72 72 80 80
contact@sysfera.com
www.sysfera.com

1 Présentation du document

1.1 Objectifs du document

Les objectifs de ce document sont les suivants:

• Identifier les tâches du projet liées aux tests et les composants qui doivent être testés.

• Recommander et décrire les stratégies de test à employer.

• Identifier les livrables pour les tests.

1.2 Structure du document

Ce document décrit le domaine d'application, la stratégie, les ressources et l'organisation des activités de vérification et de tests qui s'appliquent au logiciel VISHNU et aux livrables associés.

Le chapitre 2 décrit le contexte dans lequel les activités de vérifications et de tests ont lieu.

Le chapitre 3 décrit l'ensemble des techniques de test qui s'appliquent aux objets à tester et détaille pour chaque technique la liste des objets à tester, les critères de complétude et de succès.

Le chapitre 4 décrit l'organisation nécessaire pour réaliser les tests ainsi que les rôles et responsabilités des différents intervenants dans le processus de vérification et de tests.

Le chapitre 5 décrit le planning des tests.

1.3 Glossaire et acronymes

1.3.1 Glossaire

- **Plateforme**: terme utilisé pour désigner un ensemble de machines connectées entre elles et faisant partie d'un unique domaine Vishnu.
- **Domaine Vishnu**: ensemble des données partagées par un groupe d'utilisateurs et de machines utilisant le logiciel Vishnu et une base de données commune.
- **Transaction**: une transaction est définie comme un cas d'utilisation qu'un acteur du système (utilisateur normal ou administrateur) est présumé exécuter en utilisant une cible de test, comme par exemple soumettre ou afficher un job

1.3.2 Abréviations

- Id. Identifiant
- Réf. Référence
- Rel.# Numéro de release/version
- Resp. Responsable(s)
- TBC à compléter
- TBD à définir

1.3.3 Acronymes

- DSG Document de Spécifications Générales
- STB Document de Spécifications Techniques des Besoins
- DSD Document de Spécifications Détaillées
- SCL Standards de Codage du Logiciel
- UML Unified Modeling Language
- OS « Operating System » (système d'exploitation)
- SWIG Simplified Wrapper and Interface Generator (http://www.swig.org)
- EMF Eclipse Modeling Framework

2 Contexte des tests pour le projet VISHNU

2.1 Livrables liés aux tests

Pour chaque module livré au cours du projet (UMS, TMS, etc.) deux livrables seront fournis à des dates différentes. Le contenu de ces livrables sera le suivant:

Nom du livrable	Auteur	Contenu
Dossier de tests	Concepteur des tests	• Liste de toutes les transactions à tester • Exemples de données à tester • Détails de l'environnement de test • Méthodes et outils utilisés pour les tests
Rapport de tests	Equipe de test	• Résultats de toutes les transactions testées • Rapport d'anomalies

2.2 Exigences à tester

Le document de spécifications [DSG] liste les exigences et règles auxquelles le logiciel VISHNU doit répondre. Toutes les exigences indiquées dans ce document sont à considérer pour les tests.

2.3 Description d logiciel

Les éléments cités ci-dessous font partie du domaine des tests.

2.3.1 Programme et données

[UMS]	VISHNU User Management System client/server
[TMS]	VISHNU Task Management System client/server
[FMS]	VISHNU File Management System client/server
[IMS]	VISHNU Information Management System client/server
[CLI]	VISHNU Command Line User Interface
[PGAPI]	VISHNU C/C++/Python Programmable API

[WSAPI]	VISHNU Web Services API

2.3.2 Documentation

[SUM]	VISHNU User Manual (english)

2.3.3 Code source

[CSC]	C / C++ Source Code
[JSC]	Java Source Code

2.4 Elements logiciel exclus des tests

2.4.1 Logiciels commerciaux ou Open Source

Les éléments logiciels ci-dessous ne sont pas adressés spécifiquement par les techniques de test décrites dans ce document.

[CORBA]	Bus de communication CORBA (OmniORB)
[DIET]	Intergiciel DIET
[PostgreSQL]	Base de données PostgreSQL
[Oracle]	Base de données Oracle
[JBoss]	JBoss 6.0 Web Application Server
[SWIG]	C/C++ Wrapper
[EEMF]	Eclipse EMF Serialisation / Deserialisation
[BatchSched]	LoadLever / Torque Batch Scheduler

2.4.2 Documentation

[DSG]	Document de Spécifications Générales
[DSD]	Document de Spécifications Détaillées
[RCD]	Rapport de Conception Détaillée

2.4.3 Code source

[CGSC]	C / C++ Generated Source Code (Eclipse EMF et SWIG)
[JGSC]	Java Generated Source Code (JBoss et SWIG)
[PGSC]	Python Generated Source Code (SWIG)

3 Spécification des tests

3.1 Objectifs de tests

Le tableau ci-dessous résume les différentes catégories de tests applicables aux différents éléments à tester:

Objectifs (*)	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Programmes et données	**X**	**X**		**X**		**X**
Documentation utilisateur			**X**			
Documentation technique						
Code source		**X**			**X**	**X**

(*) Au sens des caractéristiques des produits logiciels définies dans la norme *ISO/IEC 9126-1:2001 (E) - Software Engineering – Product Quality – Part 1: Quality Model.*

3.2 Sélection des techniques de tests

La table ci-dessous recense les différents tests sélectionnés compte-tenu des spécifications du projet:

- **C**(omplet) : la technique sera appliquée à tous les composants logiciels,
- **P**(artiel) : la technique sera appliquée à certains composants logiciels,
- **NS** : Non Sélectionnée en raison de contraintes techniques,
- **NA** : Non Applicable.

Dans le cas où la technique est sélectionnée (C ou P), le nom de la section est suivi par le numéro de section décrivant la technique de test dans le chapitre suivant.

Techniques de tests	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Revue de spécification	**NS**					
Revue du plan de test	**NS**	**NS**				
Tests fonctionnels - §3.3.1	**C**					
Tests de classes d'équivalence - §3.3.2	**P**					
Tests de valeurs limites - §3.3.3		**P**				
Tests de stress - §3.3.4		**P**				
Inspection de l'interface graphique			**NA**			
Contrôle de la documentation utilisateur - §3.3.5			**P**			
Analyse des profils de performance				**NS**		
Contrôle des performances- §3.3.6				**C**		
Tests de charge - §3.3.7				**P**		
Vérification des règles de programmation - §3.3.8		**C**			**C**	**C**

Techniques de tests	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Mesure de la complexité du code - §3.3.9					C	
Revue manuelle de code		NS			NS	
Vérification de la complétude du livrable - §3.3.10					P	P
Tests d'installation - §3.3.11						F
Détection de duplication de code - §3.3.12					P	
Tests de compilation - §3.3.13						F

3.3 Liste des tests

3.3.1 Tests fonctionnels

Stratégie pour les tests fonctionnels	
Identifiant de tests :	FT
Objectifs:	Vérifier que la fonctionnalité correspond à sa spécification
Environnement de test:	2 environnements de test: 1/ Environnement 1: Client Linux Plateforme VISHNU cible comprenant: - au moins 1 serveur de calcul de chaque type décrit dans [STB] - 1 serveur de base de données PostgreSQL 2/ Environnement 2: Client Linux Plateforme VISHNU cible comprenant: - au moins 1 serveur de calcul de chaque type décrit dans [STB] - 1 serveur de base de données Oracle
Unité de test:	Une transaction = un cas d'utilisation
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI], [PGAPI]
Données utilisées:	
Technique:	Exécuter le scénario de base du cas d'utilisation et chaque chemin alternatif du cas en utilisant des données valides tout en vérifiant les résultats attendus.
Critères de complétude:	•100% des cas d'utilisations sur les différents environnements de tests
Critères de succès:	•Tous les tests prévus ont été exécutés avec succès. •Toutes les anomalies identifiées ont été enregistrées.
Considérations particulières:	

3.3.2 Tests de classes d'équivalence

Stratégie pour les Tests de Classes d'Equivalence	
Identifiant de tests :	TCE
Objectifs:	Vérifier que les fonctionnalités qui s'appliquent à des catégories d'objets distinctes fonctionnent pour ces différentes catégories
Environnement de test:	Environnement 1 (cf §3.3.1)
Unité de test:	Une transaction = un cas d'utilisation
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI]
Données utilisées:	
Technique:	Exécuter le scénario de base du cas d'utilisation en fournissant un jeu de données pour chaque classe d'objets qui aura été établis selon les types de données considérées.
Critères de complétude:	100% des cas d'utilisations s'appliquant à plusieurs classes d'objets
Critères de succès:	Tous les tests prévus ont été exécutés avec succès.
Considérations particulières:	

3.3.3 Tests de valeurs limites

Stratégie pour les Tests de Valeurs Limites	
Identifiant de tests	TVL
Objectifs:	Vérifier que le logiciel répond de manière prévisible lorsque les paramètres fournis atteignent les limites spécifiées.
Environnement de test:	Environnement 1 (cf §3.3.1)
Unité de test:	Une transaction = cas d'utilisation contenant une valeur limite
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI]
Données utilisées:	
Technique:	Pour chaque paramètre du cas d'utilisation ayant une valeur limite N, vérifier que: •à N-1: le système fonctionne comme spécifié •à N: le système fonctionne comme spécifié •à N+1: le système renvoie un message d'erreur approprié à l'utilisateur
Critères de complétude:	Tous les cas d'utilisation contenant une valeur limite
Critères de succès:	Tous les tests passés avec succès
Considérations particulières:	

3.3.4 Tests de stress

Stratégie pour les tests de stress	
Identifiant de tests	STR
Objectifs:	Vérifier que les processus de récupération, manuel ou automatique, maintiennent ou restaurent correctement les services de VISHNU dans un état connu désiré. Les types de conditions suivantes doivent être prises en compte : •Mise hors tension du poste client. •Mise hors tension d'une machine serveur •Mise hors tension d'une machine agent vishnu •Interruption de communication du réseau
Environnement de test:	Environnement 1 (cf §3.3.1) Environnement dédié. L'action sur des ressources systèmes et réseau peut être requise.
Unité de test:	Une transaction = un ou plusieurs cas d'utilisation d'un module donné
Éléments à tester:	[UMS], [TMS], [FMS], [IMS]
Données utilisées:	
Technique:	Les tests créés pour les tests fonctionnels seront utilisés pour créer une série de transactions. Lorsque le moment désiré pour tester est atteint, les opérations suivantes doivent être exécutées ou simulées : •Mise hors tension du poste client •Mise hors tension d'une machine serveur ou initiation d'une procédure d'arrêt de la machine. •Mise hors tension d'une machine agent vishnu ou initiation d'une procédure d'arrêt de la machine. •Interruption des serveurs réseau ou simulation de pertes de communication avec le réseau en débranchant physiquement ou en mettant hors tension les serveurs réseau ou les routeurs.
Critères de couverture:	100% de messages d'erreurs activés au moins une fois par un test
Critères de succès:	Dans tous les cas précédents, la base de données et le système doivent après les procédures de récupération retrouver un état connu et souhaité.
Considérations particulières:	

SysFera
46 allée d'Italie
69364 Lyon Cedex 07

Tel. + 33 (0)4 72 72 88 50
Fax + 33 (0)4 72 72 80 80
contact@sysfera.com
www.sysfera.com

3.3.5 Contrôle de la documentation utilisateur

Stratégie pour le Contrôle de la Documentation Utilisateur	
Identifiant de tests	CDU
Objectifs:	Vérifier que la documentation utilisateur permet à l'utilisateur de comprendre et d'utiliser les fonctions spécifiées
Environnement de test:	Client Linux
Unité de test:	Une transaction = une page de manuel utilisateur
Éléments à tester:	[SUM] Vishnu User Manual
Données utilisées:	
Technique:	Vérifier que la page de manuel est cohérente avec la spécification du logiciel
Critères de complétude:	100% des pages de manuel utilisateur
Critères de succès:	Toutes les pages de manuel passent la vérification avec succès.
Considérations particulières:	Non applicable dans le cas où la documentation est générée automatiquement à partir des spécifications.

3.3.6 Contrôle des performances

Stratégie pour le Contrôle Des Performances	
Identifiant de tests	CDP
Objectifs:	Vérifier que les exigences de performance quantifiées sont effectivement satisfaites dans l'environnement et selon les contraintes spécifiées
Environnement de test:	Les tests seront réalisés sur une plateforme dédiée afin de permettre un contrôle complet et des mesures exactes. Le réseau d'interconnexion des machines sera également dédié. La plateforme inclura au moins 5 machines serveur. Les serveurs de calcul seront d'un type donné conforme à [STB]. Le serveur de base de données sera d'un type donné conforme à [STB] et le volume de la base de données sera conforme à [STB].
Unité de test:	Une transaction = une exigence de performance quantifiée
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI]
Données utilisées:	
Technique:	Executer un cas d'utilisation représentatif dans l'environnement de test, mesurer le temps de réponse obtenu ou l'état du système (consommation de ressources) et le comparer à l'exigence de

	performance
Critères de complétude:	100% des exigences de performance quantifiées définies dans [STB]
Critères de succès:	• Si le critère est un délai maximum, les valeurs obtenues ne sont jamais supérieures de plus de 10% à ce délai et sont en moyenne inférieures. • Si le critère est une consommation de ressource maximum, les valeurs obtenues ne sont jamais supérieures de plus de 10% à la valeur maximum et sont en moyenne inférieures.
Considérations particulières:	

3.3.7 Tests de charge

Stratégie pour les tests de charge	
Identifiant de tests:	LOAD
Objectifs:	Vérifier les temps de performance pour des transactions données selon des conditions de charge de travail particulières.
Environnement de test:	Les tests seront réalisés sur une plateforme dédiée afin de permettre un contrôle complet et des mesures exactes. Le réseau d'interconnexion des machines sera également dédié. La plateforme inclura au moins 5 machines serveur. Les serveurs de calcul seront d'un type donné conforme à [STB]. Le serveur de base de données sera d'un type donné conforme à [STB].
Unité de test:	Une transaction = un cas d'utilisation
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI]
Données utilisées:	Seul le critère de volume des données sera pertinent dans le cadre de ces tests. Le contenu sera donc générique.
Technique:	Utiliser les tests développés pour les tests fonctionnels, en modifiant les fichiers de données pour accroître le nombre de transactions ou le volume des données.
Critères de complétude:	Pour chacune des exigences de capacité spécifiées (par ex. le nombre de requêtes simultanées), les requêtes les plus consommatrices de ressources doivent être testées.
Critères de succès:	Réalisation des tests sans défaillance et dans un délai acceptable
Considérations particulières:	Les bases de données utilisées pour les tests de performance doivent avoir une taille courante.

3.3.8 Conformité aux règles de programmation

Stratégie pour la Conformité aux Règles de Programmation	
Identifiant de tests:	CRP
Objectifs:	Vérifier que le code source respecte les règles spécifiées dans le Standard de Codage Logiciel applicable [SCL]
Environnement de test:	N/A
Unité de test:	1 transaction = 1 module de code source
Éléments à tester:	[CSC] C/C++ Source code [JSC] Java Source code
Données utilisées:	-
Technique:	Analyse statique de code outillé (SQuORE et Logiscope)
Critères de complétude:	100% des fichiers de code source
Critères de succès:	Le code respecte toutes les règles de codage de catégorie «REQUIRED » spécifiées pour le projet [SCL] sauf dérogation clairement documentée dans le code.
Considérations particulières:	

3.3.9 Mesure de la complexité du code

Stratégie pour la Mesure de la Complexité du Code	
Identifiant de tests:	MCC
Objectifs:	Vérifier que le code source ne contient pas de composants (classes; fonctions) dont la complexité intrinsèque pourrait altérer le niveau de maintenabilité.
Environnement de test:	NA
Unité de test:	1 transaction = 1 module de code source
Éléments à tester:	[CSC] C/C++ Source code [JSC] Java Source code
Données utilisées:	-
Technique:	Analyse statique de code outillé (SQuORE et Logiscope)
Critères de complétude:	100% des fichiers de code source
Critères de succès:	Pas plus de 1% de composants classés « A risque »
Considérations particulières:	

3.3.10 Vérification de la complétude du livrable

Stratégie pour la Vérification de la Complétude du Livrable	
Identifiant de tests:	VCL
Objectifs:	Vérifier que le livrable contient tous les éléments prévus.
Environnement de test:	NA
Unité de test:	1 livrable
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI], [PGAPI], [WSAPI]
Données utilisées:	-
Technique:	Récupérer le livrable depuis le dépôt de livraison puis suivre les instructions d'installation et de démarrage.
Critères de complétude:	100% des éléments du livrable
Critères de succès:	Installation et démarrage réalisés sans nécessité d'obtenir des informations / fichiers additionnels.
Considérations particulières:	

3.3.11 Tests d'installation

Stratégie pour les tests d'installation	
Identifiant de tests:	INS
Objectifs:	Vérifier que le logiciel peut être installé conformément à la documentation d'installation
Environnement de test:	Environnements cible d'installation du produit comprenant: • serveurs de calcul (Minimum 2 serveurs) • serveurs de stockage (Minimum 2 serveurs) • serveurs dédiés VISHNU • serveur de base de données • serveur web • client VISHNU poste de travail
Unité de test:	Une transaction = 1 environnement d'installation
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI], [PGAPI], [WSAPI]
Données utilisées:	N/A
Technique:	Installer le produit selon la procédure décrite dans le manuel d'installation Vérifier le bon fonctionnement du produit dans l'environnement d'installation
Critères de complétude :	Installation sur 100% des environnements cible définis dans [STB]

Critères de succès:	100% des installations réussies
Considérations particulières:	

3.3.12 Détection de duplication de code

Stratégie pour la Détection de Duplication de Code	
Identifiant de tests:	DDC
Objectifs:	Vérifier que le code source ne contient pas de duplication de code
Environnement de test:	-
Unité de test:	1 transaction = 1 module de code source
Éléments à tester:	[CSC] C/C++ Source code [JSC] Java Source code
Données utilisées:	-
Technique:	Analyse statique de code outillé (SQuORE et Logiscope)
Critères de complétude:	100% des fichiers de code source
Critères de succès:	Moins de 10% de code dupliqué.
Considérations particulières:	

3.3.13 Tests de compilation

Stratégie pour les tests de compilation	
Identifiant de tests:	COMP
Objectifs:	Valider le fait que les différents modules peuvent être compilés sur les plateformes spécifiées dans les besoins techniques.
Environnement de test:	Environnement logiciel spécifié dans les besoins techniques: • Plateforme Linux i386 • Compilateur C/C++: GCC • Sun Java SDK 1.6
Unité de test:	1 transaction = Un module logiciel
Éléments à tester:	[UMS], [TMS], [FMS], [IMS], [CLI], [PGAPI], [WSAPI]
Données utilisées:	N/A
Technique:	Lancer la compilation du module sur les différentes plateformes applicables
Critère de complétude :	Compilation de tous les exécutables

Critère de succès:	Aucune erreur de compilation
Considérations particulières:	

SysFera
46 allée d'Italie
69364 Lyon Cedex 07

Tel. + 33 (0)4 72 72 88 50
Fax + 33 (0)4 72 72 80 80
contact@sysfera.com
www.sysfera.com

4 Organisation des tests

4.1 Objectifs

Les objectifs de l'organisation des tests seront les suivants:

- Associer aux différentes techniques de tests les ressources adéquates
- S'assurer que les processus de tests vérifient les standards de bonnes pratiques définis dans le cadre du projet
- Identifier les personnes et organisations impliquées dans le processus de test

4.2 Rôles et responsabilités

Le tableau ci-dessous identifie les fonctions et ressources nécessaires pour l'exécution de la politique de test:

Fonctions	Ressources	Organisation
PM – Chef de projet	Benjamin Isnard	**SysFera**
TM – Manager des tests	<TBC>	**SysFera**
QA – Assurance Qualité	<TBC>	**SysFera**
TD – Conception des tests:	Equipe de développement	**SysFera**
TX - Execution des test: [FT], [TCE], [TVL], [STR], [CDU], [CDP], [LOAD], [VCL], [INS], [COMP]: [CRP]; [MCC]; [DDC]:	 Eddy Caron Daouda Traoré Eugene Pamba Kevin Coulomb Christophe Peron	 **SysFera** **SysFera** **SysFera** **SysFera** **SQuORING Technologies**
AE – Experts des domaines de l'application	Eddy Caron Daouda Traoré Eugene Pamba Kevin Coulomb	**SysFera** **SysFera** **SysFera** **SysFera**
LE – Expert language de programmation	<TBC>	**SysFera**
TE – Support de l'environnement de test	<TBC>	**SysFera**
UR – Représentant des utilisateurs	Boris Daix Samuel Kortas	**EDF** **EDF**
IA – Agent de vérification et validation indépendant	Christophe Peron	**SQuORING Technologies**

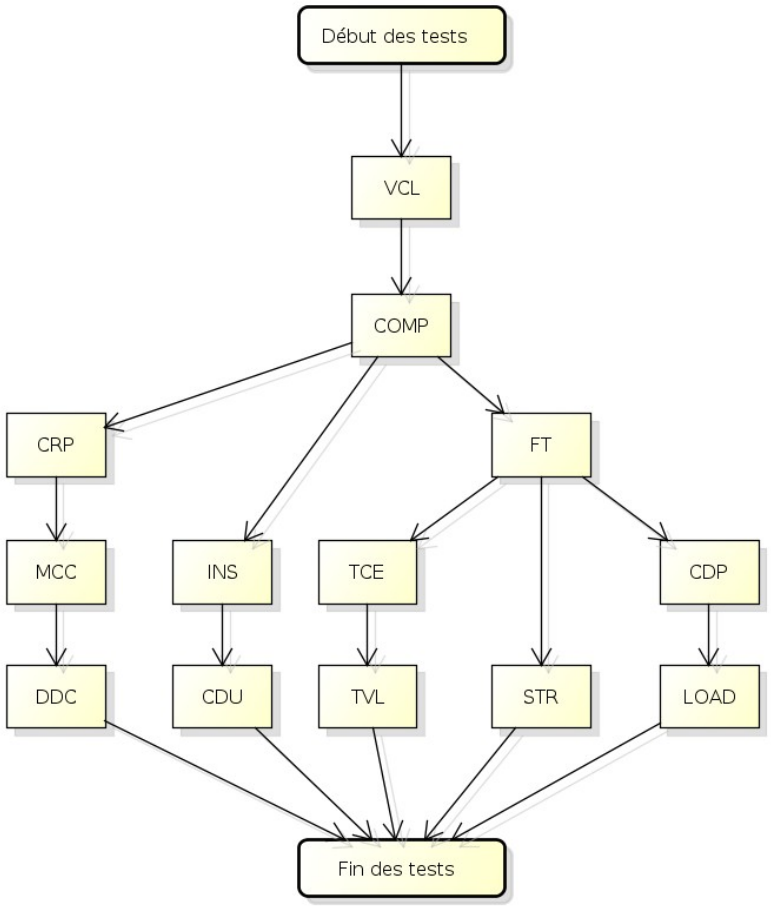
5 Planning des tests

5.1 Contraintes identifiées

Aucune

5.2 Phases de tests

Pour chaque release d'un module logiciel le planning des tests est le suivant:



[VCL]	Vérification de la Complétude du Livrable
[COMP]	Compilation du module
[INS]	Tests de la procédure d'Installation et configuration du module VISHNU (et ses dépendances)
[CDU]	Contrôle de la Documentation Utilisateur
[FT]	Tests Fonctionnels

[TCE]	Tests de Classes d'Equivalence
[TVL]	Tests de Valeurs Limites
[STR]	Tests de Stress
[CDP]	Contrôle De Performances
[LOAD]	Tests de charge
[CRP]	Conformité aux Règles de Programmation
[MCC]	Mesure de la Complexité du Code
[DDC]	Détection de Duplication de Code

SysFera
46 allée d'Italie
69364 Lyon Cedex 07

Tel. + 33 (0)4 72 72 88 50
Fax + 33 (0)4 72 72 80 80
contact@sysfera.com
www.sysfera.com