

KIT 24

Plan de recette

et

Fiches de test

Intitulé du Projet



Préparé par: Auteur

Préparé le : jj/mm/aa

Version 2015

Description du document

Nom du projet : Nom du projet

Nom du chef de projet : Nom du chef de projet
projet :

N° de version :
n° xx

Préparé par : Nom du rédacteur

Date de version : Date de rédaction

Revu par : Nom du valideur

Date de revue : Date de validation

Liste de distribution

De	Date	Téléphone

A	Action*	Date	Téléphone

* Types d'action : Approbation, revue, Information, fichier, action nécessaire, organisation de réunion, autre (à préciser).

Historique des versions

N° de version	Date de version	Revue par	Description	Nom du fichier
V0.1				
V0.2				
V1.0				

Acceptation

Client	Date et Signature	Chef de Projet	Date et Signature

Sommaire

Table des matières

1. Introduction	4
2. La méthodologie.....	4
2.1 Contexte de la recette	4
2.2 L'organisation des tests.....	5
2.2.1 Mode d'indentification et d'organisation des fiches de tests.....	7
3. Les conditions d'acceptation	7
3.1.1 Gestion d'un défaut sur un test.....	7
3.1.2 Le classement des types de défauts	7
3.1.3 Le classement des types de défaut de test.....	8
3.1.4 Gestion d'une fiche	8
3.1.5 Gestion d'un ensemble de fiches.....	8
3.1.6 La fiche d'anomalie	9
3.1.7 La recette globale.....	10
4. La gestion des fiches de test	10
4.1 Organisation des fiches de tests.....	10
4.2 Organisation d'une fiche de tests	12
4.3 Contenu d'une fiche de test.....	15
4.4 Description détaillée de toutes les fiches.....	15
5. Annexes	16
5.1 Les nomenclatures utilisées par le plan de recette (un exemple):.....	16
5.1.1 Divers.....	16
5.1.2 Pour les matériels :	16
5.1.3 Pour les logiciels :	16
5.1.4 Fonction HW (matériel) :	16

1. Introduction

Un plan de test se négocie avec votre client, les règles contenues dans un plan de recette sont propres à chaque client et à vous-même. Malgré toutes les règles appliquées, il y a des us et coutumes standards à la profession informatique (dans les administrations françaises ces règles sont codifiées).

En général un plan de recette comprend les grands chapitres suivants dans l'ordre :

1. La méthodologie : le contexte et l'organisation de la phase recette
2. Les conditions d'acceptation
 - Le principe de classement des défauts et leur gestion
 - La méthode de gestion des fiches de tests
 - Le principe d'acceptation de la solution complète
3. La description détaillée d'une fiche de test
4. La description détaillée de toutes les fiches de test *(il y a facilement plus de 100 fiches pour la recette d'un projet simple !)*

2. La méthodologie

2.1 Contexte de la recette

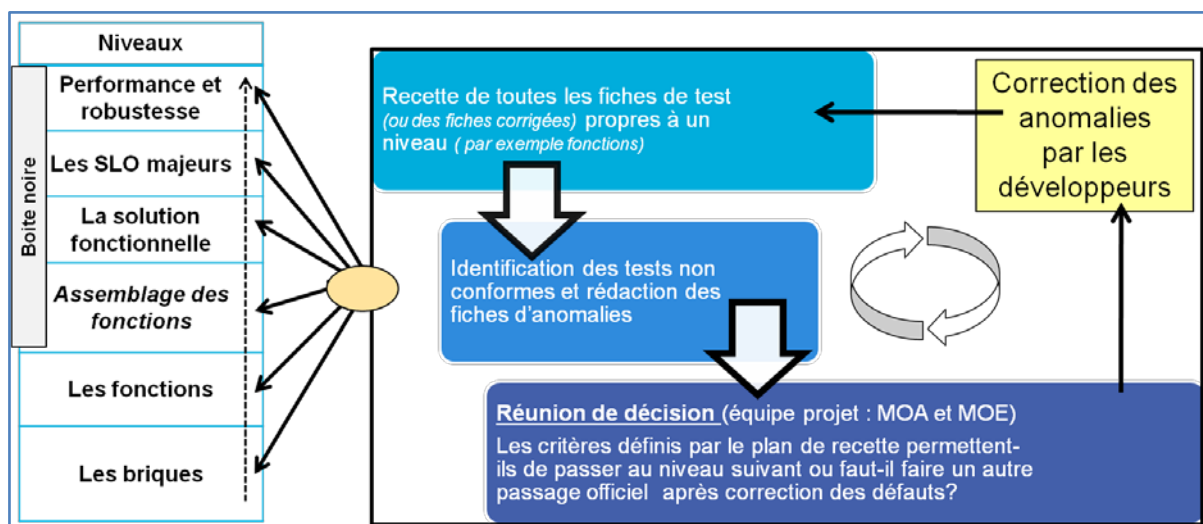
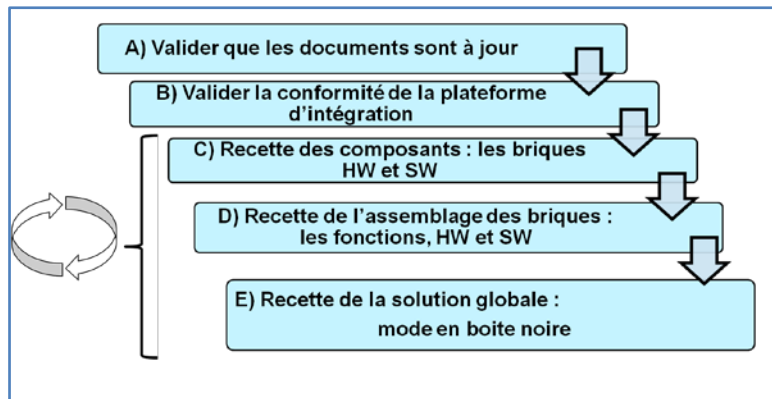
Le début de la phase de recette est déclenché lorsque le MOE considère que la plate-forme d'intégration est conforme aux spécifications et qu'elle permet de démontrer que les objectifs de la phase de développement sont atteints.

La recette est réalisée sur la plate-forme d'intégration conformément aux règles suivantes :

1. En premier, recette en interne par le MOE pour valider que ses équipes ont bien atteint les objectifs. C'est le chef de projet qui va jouer le rôle du client et dérouler les fiches de test contenues dans le plan de test. Le but est de présenter au MOA une solution aboutie. Il est très important qu'il ne découvre pas de problème critique lors de ces tests.
2. Ensuite une réunion de décision interne (*MOE face à sa direction*) décidera du passage en phase de recette officielle par le client.
3. Enfin, elle est réalisée par le client qui va dérouler les fiches de tests contenues dans le plan de test. La recette globale est déclarée conforme quand toutes les fiches sont déclarées acceptées ou comportant des défauts acceptables pour le passage en phase de déploiement

Nota : dans le cas où le client fournit un composant majeur (exemple : réseau...), le MOE va faire une recette partielle et officielle sur ce composant lors de sa livraison.

A chaque cycle complet d'une recette, les mêmes vérifications et tests sont faites, étape par étape. On trouve dans l'ordre :



C'est au client d'apprécier s'il veut voir toutes les fiches ou s'il ne démarre qu'au niveau de l'application en boîte noire. Souvent le client passe au hasard certaines fiches de test sur des briques ou fonctions critiques pour le projet : le but est de valider que vous avez bien fait votre travail de spécialiste !

2.2 L'organisation des tests

Un cycle de recette est très formel, il est planifié d'avance par le comité de pilotage du projet en fonction du niveau de défaut. On décide au début du projet du nombre maximum de cycles par niveau, par exemple 3. Ces étapes apparaissent dans le plan projet (au niveau MS Project), elles sont couteuses et figurent au budget du projet car :

- Vous devez avoir présent sur site tous les spécialistes du projet (architecte, base de données, réseau...) et les responsables des divers équipes (développeur logiciel, charte graphique, exploitation, support)
- Le client se fait assister pas des spécialistes de même profil que vos équipes et il fait venir :
 - Les représentants de son exploitation et support
 - Les représentants de ses utilisateurs pour dérouler les tests fonctionnels : parfois un par pays concerné (l'aspect langue) ; si votre projet est sur trois continents, cette phase peut prendre beaucoup d'ampleur
- Il faut mobiliser les managers du client et votre entreprise (comité de pilotage projet) pour avoir la capacité de gérer rapidement un conflit, car très souvent le planning du projet est en retard
- Vos sous-traitants sont présents et ceux de votre client (par exemple celui qui fournit son réseau)

Le client valide toujours au démarrage d'un cycle :

- les documents
- la conformité de la plate-forme

Nota: ils doivent être à la dernière version connue au comité de pilotage.

Le plan de recette, avec les fiches de tests, fait l'objet d'une première validation formelle par le client lors de la validation de la spécification fonctionnelle et avant la phase de développement (mécanisme que vous retrouverez dans le projet Yaka).

Toute évolution doit faire l'objet de demande du MOE ou du MOA en comité de pilotage et doit faire l'objet d'une validation formelle : signature des deux parties.

Le client ne peut ajouter d'action de test non spécifiée sur la fiche de test.

La progression des tests suit le principe d'assemblage de la plate-forme d'intégration, en partant du niveau des briques de base jusqu'aux niveaux de la solution en mode boîte noire.

La boîte noire application est testée suivant quatre niveaux :

1. Validation de l'assemblage complet de la solution avec les bons composants : basée sur la nomenclature
2. Vérification fonctionnelle : l'application fournit des fonctions conformes aux spécifications
3. Validation des SLO : simulation de la panne et vérification que le temps de reconstruction d'un service est conforme (4h, une ½ journée...)

4. Vérification de la performance et de la solidité de l'ensemble : tests de charge et puis tests de robustesse lors d'une mauvaise utilisation de l'application

Niveaux	HW	SW
Performance et solidité	Solution en mode boîte noire	
Les SLOs		
La solution fonctionnelle		
Assemblage des fonctions		
Les fonctions	HW	SW
Les briques	HW	SW

2.2.1 Mode d'indentification et d'organisation des fiches de tests

Ce point est propre à chaque projet. En se référant à l'exemple suivi dans ce document on pourrait avoir :

3. Les conditions d'acceptation

3.1.1 Gestion d'un défaut sur un test

Si un test est KO, le défaut est classé en fonction de sa gravité et une fiche d'anomalie est rédigée par le responsable du test. Cette fiche est envoyée au développeur responsable de l'élément en cause, ce dernier va proposer une correction dans un délai bref : soit défini à l'avance soit négocié (*ce délai est de toute manière en jours*). Ensuite la correction est présentée pour une nouvelle recette. Suivant l'importance du défaut, le déroulement de la recette peut être arrêté.

3.1.2 Le classement des types de défauts

Lorsqu'un test n'est pas conforme, le (*ou les*) défaut est classé soit comme :

- **Majeur** : lorsqu'un défaut est bloquant, car il ne permet pas de poursuivre la recette
- **Mineur** : le défaut doit être corrigé mais ne bloque pas la progression du processus de recette

Il faut définir avec son client les conditions d'acceptation ou de refus. Par exemple il sera décidé que pour une fiche de test il y a :

Une acceptation :

- si pas de défaut

- si défaut mineur et un plan de correction simple
-

Un refus :

- si un défaut majeur
- si le plan de correction d'une erreur mineure est non satisfaisant ou complexe
- si c'est la troisième fois que la fiche présente encore un défaut mineur
- ...

Nota: Cette partie fait l'objet de votre TP

3.1.3 Le classement des types de défaut de test

Les erreurs sont classées par grands types de défauts :

- Liés à la charte graphique
- Aux fonctionnalités de l'application
- Au lien vers l'extérieur : autres application
- A la performance
- A la robustesse
- A la sécurité

3.1.4 Gestion d'une fiche

L'acceptation d'une fiche (exemple) :

- si pas de défaut
- si défaut mineur et un plan de correction

Le refus d'une fiche (exemple) :

- si défaut majeur
- si plan de correction d'une erreur mineure non satisfaisant
- si troisième fois que la fiche est présentée et a toujours un défaut mineur

3.1.5 Gestion d'un ensemble de fiches

Un test peut nécessiter plusieurs fiches de tests : *pour une brique, une fonction, ou une fonctionnalité de la boîte noire.*

Il faut aussi définir des règles pour un ensemble de fiches:

A) L'acceptation sera prononcée si par exemple il y a (exemple) :

- aucune fiche en défaut majeur
- moins de 10% des fiches sont en défaut mineur

- moins de 30% des fiches sont en défaut mineur, mais le comité de pilotage du projet a donné son accord sur les corrections (*non risque pour le déroulé de la recette*)

B) La non acceptation sera prononcée si par exemple il y a (exemple) :

- une fiche est en défaut majeur
- si plus de 40% des fiches sont en défaut mineur....

Remarque: l'équipe projet décide qu'une alerte sera déclenchée par le comité de pilotage du projet si trois corrections successives d'un défaut majeur ne règlent pas le problème : on est toujours en défaut majeur.

En mode alerte : un plan d'action est alors présenté au niveau des directions (MOA et MOE).

3.1.6 La fiche d'anomalie

En cas de défaut mineur ou majeur, une fiche d'anomalie est émise :

- Dans une première étape c'est le testeur (MOE) qui fait une description plus détaillée du défaut constaté
- Ensuite le développeur (MOE) décrit, après analyse, la cause du défaut et la correction qu'il propose et le délai qu'il propose
- Enfin les deux parties signent la fiche pour accord avant le début de la correction

Après la correction, la fiche est représentée de nouveau au client lors d'un nouveau cycle de recette

Exemple de fiche d'anomalie (voir annexe pour détail) :

Fiche d'anomalie Projet T&F				Niveau	Mineur : <input type="checkbox"/> Majeur : <input type="checkbox"/>
Date du défaut	29/02/12	Acteur du test	Dupont - MOE- direction technique	Référence fiche de test	HW-SRV-web-04.05
Titre du test : Test de charge sur serveur WEB					
Description du défaut par le testeur (MOA)					
Impact fonctionnel					
Explication du niveau (pourquoi majeur / mineur)					
Nom et signature du MOA					
Solution proposée par le développeur (MOE)					
Identification de la cause du défaut					
Correction proposée					
Délai proposée					
Nom et signature du MOA et MOE		Acceptation MOA			
		Acceptation MOE			

3.1.7 La recette globale

Elle est prononcée suite à une réunion spécifique MOA / MOE où l'on vérifie que :

- Aucune fiche de test en défaut majeur n'est encore 'active'
- Les fiches avec un défaut mineur sont acceptables et ne mettent pas en risque les phases suivantes : *le déploiement et l'exploitation*
- Une fois ces points validés, le client analyse le déroulement de l'ensemble du plan de recette pour vérifier que :
- le nombre global de refus majeurs et mineurs n'est pas trop important : *cela voudrait dire que la solution présentée est encore fragile*
- Trop de défauts majeurs portent sur un composant sensible (*plusieurs fiches de tests sont liées à ce composant*)
- les fiches corrigées trois fois ne portent pas sur un composant sensible
- etc.

Cette analyse peut conduire le client à refuser de signer la phase recette. On est clairement en situation de crise : décision à prendre au niveau des directions !

4. La gestion des fiches de test

4.1 Organisation des fiches de tests

Les fiches de tests sont classées par niveau et par domaine de ce niveau, par exemple en fonction de l'organisation décrite par le tableau suivant :

Niveaux	HW	SW	Solution en Boîte noire	Exemple	Qui
Performance et solidité	Simulation de charge, robustesse à l'usage			Charge maxi, mauvaise utilisation,	
Les SLA majeurs	Démonstration de la reconstruction d'un service KO: en 4h, 1jour ...			Rupture :réseau, NAS, serveur local, SGBD, lien application externe...	
La solution fonctionnelle	Les grandes fonctions de l'application + la sécurité			Login, bon de commande, tableau de bord, Hakage ...	
Assemblage des fonctions : boîte noire	Validation de l'assemblage des différentes briques par fonction: matériel , logiciel ...			Bloc hardware, structure de bases de données (centrale et locales)	Client
Les fonctions	Ferme serveurs web, accès internet (firewall)	Lien annuaire et login		Load balancing, composants d'un SLA majeurs	
Les briques	Serveur web ou central, NAS, réseau, porte de travail...	Login, PT, SGBD oracle central ou locale		Conformité à la nomenclature, au fonction logiciel élémentaire (niveau du code)	Interne

La gestion de fiches de tests est faite par un ensemble de tableaux qui permettent de gérer toutes les fiches et de suivre leurs évolutions .

Il y a un tableau de suivi des fiches par niveau (suivant le schéma précédent) :

- 1) Niveau brique matériel
- 2) Niveau brique logiciel
- 3) Niveau fonction matériel
- 4) Niveau fonction logiciel
- 5) Niveau validation assemblage de la boîte noire
- 6) Niveau fonctionnel boîte noire
- 7) Niveau SLA majeurs boîte noire
- 8) Niveau performance et robustesse boîte noire

Les tableaux suivants sont des exemples pour illustrer la démarche:

	Date	xx-yy-zzzz	Numéro du cycle de recette		2	Sur 4
	N° ordre	Libellé du test	Référence du test	Code du test	Résultat du test <i>(OK ou KO)</i>	Numéro fiche d'anomalie
Niveau brique matériel						
Serveur web						
	BH01	Serveur web central	Conformité nomenclature	HW-Srv-Webc-01.01	OK	
	BH02		Démarrage et écran accueil administrateur	HW-Srv-Webc-01.02	OK	
	BH03		Test de charge	HW-Srv-Webc-01.03	KO	AN-HW- 021
NAS						
	BH04	NAS central	Conformité nomenclature	HW-Nas-cent—01.01	OK	
	BH05		Lecture écriture de suite distant	HW-Nas-cent-01.02	OK	
Niveau brique logiciel						
Lien annuaire						
	FS02	Connecteur	Via login validation lien avec Exchange	SW-Cd-Ann-01.01	OK	
	FS02		Message erreur sur	SW-Con-Ann-01.02	OK	

		annuaire	coupure réseau			
	FS02		Accès simultanés 100 login	SW-Con-Ann-01.03	KO	AN-SW- 031
Niveau boîte noire						
Performance et robustesse						
	BN01	Login	100 Login concurrents	Rob-Log-01.01	OK	
	BN02		Login incorrect (champ mal rempli)	Rob-Log-01.02	KO	AN-BN- 018
	BN03		Rupture réseau vers annuaire	Rob -Log-01.03	OK	

En fonction de l'organisation des tests il faut classer les fiches de tests de manière rigoureuse car il y a rapidement plusieurs centaines de fiches à gérer même pour un projet simple.

Rappel d'une organisation possible présentée dans le support de formation

4.2 Organisation d'une fiche de tests

Les fiches de tests sont classées par niveau et par domaine de ce niveau, par exemple pour savoir rapidement et sans erreur à quoi correspond une fiche de tests, il faut impérativement mentionner :

- 1. le niveau auquel est liée la fiche** (brique, fonction ou boîte noire). Pour les niveaux brique et fonction, indiquer la fiche concerne
 - le HW : composant matériel
 - le SW : composant logiciel
- 2. La version de l'élément**(ou de l'ensemble d'éléments) concerné :
 - Pour brique et fonction il faut donner la version de la nomenclature si le test concerne un composant HW ou SW , puis sa référence dans la nomenclature. Par exemple, pour un des serveurs WEB on aura :
 - Nomenclature V02.3, référence : HW-SRV-web-04.05
 - Pour la boîte noire il faut donner la version de la solution installée pour le test sur la plate-forme d'intégration. En fait on a deux versions à fournir :

➤La version du moteur en place : Moteur - V05.02.3

➤La version du logiciel applicatif installé : Soft – V04.5.8

3. Dans notre exemple on va avoir une nomenclature des fiches de tests qui pourrait se présenter ainsi :

a) Les fiches de niveau brique (exemples) :

Fiches test pour les briques HW			
N°	brique	Titre test	Code du test
BH01	Serveur web central	Conformité nomenclature	HW-Srv-Webc-01.01
BH02		Démarrage et écran accueil administrateur	HW-Srv-Webc-01.02
BH03		Test de charge	HW-Srv-Webc-01.03
BH04	NAS central	Conformité nomenclature	HW-Nas-cent—01.01
BH05		Lecture écriture de suite distant	HW-Nas-cent-01.02
BH06	NAS pays	Conformité nomenclature	HW-Nas-cont-01.01
BH07		Lecture écriture de suite distant	HW-Nas-cont-01.02
BH08	Routeur central	Conformité nomenclature	HW-Rout-cent-01.01
BH09		Accès console exploitation	HW-Nas-cent-01.02
BH01	Poste travail camion	Conformité nomenclature	HW-PT-truck-01.01
BH10		Envoi message du camion	HW-Nas-truck -1.02
BH11		Prise de main à distance	HW-Nas-truck -1.03
BH12	etc		

Fiches test pour les briques SW			
N°	brique	Titre test	Code du test
BS01	Code login	Lien annuaire	SW-Cd-Log-01.01
BS02		Ecran d'accueil client interne	SW-Cd-Log-01.02
BS03		Test d'accès simultanés	SW-Cd-Log-01.03
BS04		Si OK : affichage page accueil	SW-Cd-Log-01.04
BS05	Code page accueil	Conformité charte graphique	SW-Cd-Acc-01.01
BS06		Champ fonction désirée	SW-Cd-Acc-01.02
BS07		Utilisation erronée des champs	SW-Cd-Acc-01.03
BS08	Connecteur annuaire	Via login validation lien avec Exchange	SW-Con-Ann-01.01
BS09		Message erreur sur coupure réseau	SW-Con-Ann-01.02
BS01		Accès simultanés 100 login	SW-Con-Ann-01.03
BS10	Code bon commande	Conformité charte graphique	SW-Cd-Com-01.01
BS11		Champs conforme et retour commande	SW-Cd-Com-01.02
BS12		Utilisation erronée des champs	SW-Cd-Com-01.03
BS13	etc		

b) Les fiches de niveau fonction (exemples) :

Fiches test pour les fonctions SW			
N°	fonctions	Titre test	Code du test
FS01	Code login	Lien annuaire	SW-Cd-Log-01.01
FS02		Ecran d'accueil client interne	SW-Cd-Log-01.02
FS03		Test d'accès simultanés	SW-Cd-Log-01.03
FS04		Si OK : affichage page accueil	SW-Cd-Log-01.04
FS05	Code page accueil	Conformité charte graphique	SW-Cd-Acc-01.01
FS06		Champ fonction désirée	SW-Cd-Acc-01.0
FS07		Utilisation erronée des champs	SW-Cd-Acc-01.03
FS08	Connecteur annuaire	Via login validation lien avec Exchange	SW-Cd-Ann-01.01
FS09		Message erreur sur coupure réseau	SW-Con-Ann-01.02
FS10		Accès simultanés 100 login	SW-Con-Ann-01.03
FS12	Code bon commande	Conformité charte graphique	SW-Cd-Com-01.01
FS13		Champs conforme et retour commande	SW-Cd-Com-01.02
FS14		Utilisation erronée des champs	SW-Cd-Com-01.03
FS15	etc		

c) Les fiches de niveau boîte noire / robustesse (exemples) :

Fiches test pour boîte noire- robustesse			
N°	robustesse	Titre test	Code du test
BN01	Login	100 Login concurrents	Rob-Log-01.01
BN02		Login incorrect (<i>champ mal rempli</i>)	Rob-Log-01.02
BN03		Rupture réseau vers annuaire	Rob -Log-01.03
BN05	Page accueil	1000 accès concurrents	Rob -Acc-01.01
BN06		Champs mal remplis	Rob -Acc-01.02
BN07		D'un site distant, reprise après rupture réseau (le serveur central reprend la main)	Rob -Acc-01.03
Bn08	Ferme serveurs WEB	Charge nominale : 3000 utilisateurs	Rob -FWeb-01.01
BN09		Charge nominale + 20% : 3600 utilisateurs	Rob -FWeb-01.02
Bn10		Défaut d'un serveur sur les 6 de la ferme	Rob -FWeb-01.03
BN11	etc		

4.3 Contenu d'une fiche de test

Toute fiche de test comprend deux zones :

1. **Une zone d'identification** qui indique par exemple :
 - Le nom du projet
 - Qui fait le test : nom, fonction et origine (MOE ou MOA)
 - Identification : la date, la version de la fiche : une fiche peut être modifiée dans le temps
 - L'utilisation de la fiche : une fiche peut être 'jouée' plusieurs fois suite à un défaut
 - La référence de la fiche : pour identifier quel niveau est concerné (voir paragraphe précédent) : nomenclature, moteur version logiciel
 - De connaître la nature du test : son titre
 -

2. **Une zone de description** du test qui indique par exemple :
 - Le nom du test (son titre)
 - Le contexte du test : les outils utilisés, et les étapes préalables nécessaires au test
 - Les références de documents projet concernés (spécifications ...)
 - La description du test précise et DETAILLÉE : il ne peut y avoir d'équivoque. Le testeur doit pouvoir faire sans assistance et ne doit pas pouvoir faire autre chose.
 - ...

4.4 Description détaillée de toutes les fiches

Ce chapitre comprend en fait toutes les fiches de tests: ce sera le résultat de la rédaction de toutes les fiches de tests (votre travail).

5. Annexes

5.1 Les nomenclatures utilisées par le plan de recette (un exemple):

Une nomenclature commune pour les éléments testés est nécessaire pour avoir des fiches cohérentes entre elles. En effet les tests à réaliser et leur contenu détaillé sont définis par les équipes techniques (matériel et logiciel) en relation avec le chef de projet.

Voici quelques exemples de ce qui peut être défini et que l'on va retrouver dans la définition des références par test :

5.1.1 Divers

➤ Web	Web
➤ Central	C
➤ Pays	Cont
➤ Agence	Ag

5.1.2 Pour les matériels :

➤ Serveur :	SRV
➤ Routeur	RouT
➤ Poste de travail	PT
➤ NAS central	NasC
➤ NAS région	NasR
➤ Switch	Swt
➤ Poste de travail camion	PTC

5.1.3 Pour les logiciels :

OS Windows 7	Wind7
Logiciel balancing serveur :	Balsrv
Code page d'accueil	CftAc
Code page login	Cftlog
Code envoi impression	Cftimp
Lien annuaire Exchange	Lech
Ecriture SGBD Oracle	EcrOr

5.1.4 Fonction HW (matériel) :

Des serveurs de type WEB

Un NAS raid5 dédié

Un réseau haut débit spécifique aux serveurs web pour gérer les dialogues de répartition de charge

Tec...

SW (logiciel) par exemple :

- Le logiciel de load-balancing
- L'OS Windows7
- Une base de données spécifique propre au frontal
- Etc ...

La nomenclature des fonctions retenue est (exemples) :

Login :	Log	nature : web, central, distant ou stockage
Frontal WEB	Fweb	nature : N/A*
Poste de travail	Poste T	nature : camion ou classique

(*) : non applicable