



La Démarche Qualité pour la Qualification des Logiciels 2017

La recette fonctionnelle des Systèmes d'Information

1-) Qu'est-ce que la recette fonctionnelle des SI ?

- ❖ **Moyen de vérification de la conformité d'un SI livré par rapport aux spécifications d'une commande**
- ❖ **Une des phases de développement des projets informatiques**

2-) Pourquoi réaliser une recette fonctionnelle?

- ❖ **Pour attester qu'un contrat entre deux parties a bien été respecté**
- ❖ **Pour assurer la démarche qualité quant à la qualification de logiciels**

3-) Comment réaliser une recette fonctionnelle?

- ❖ **Réalisation de tests logiciels**

La recette fonctionnelle des Systèmes d'Information

Le Test

- ❖ Qu'est-ce qu'un test ?
 - Un test appartient à l'activité de validation et de vérification du logiciel qui consiste à déterminer si ce dernier a été développé correctement en fonction des exigences
 - C'est le vecteur principal de l'amélioration de la qualité du logiciel

- ❖ En quoi consiste-t-il ?
 - Le test consiste à exécuter un programme dans le but de faire apparaître les failles (ou anomalies) enfouies dans celui-ci
 - C'est un processus manuel ou automatique

La recette fonctionnelle des SI

SOMMAIRE

- ❑ Les Grands Concepts
- ❑ La Démarche
- ❑ L' Organisation de la Qualification
- ❑ Le Plan de Test

La recette fonctionnelle des SI

Les Grands Concepts

Les tests fonctionnels manipulent les concepts suivants

- Les exigences
 - ✓ **Ce sont des données** que doit respecter l'AAT*
- Les cas de test
 - ✓ **Les Tests peuvent être plus ou moins détaillés**
 - ✓ **Le Test est la version opérationnelle d'un cas de test.**
 - ✓ **Les Tests doivent couvrir les exigences présentes dans les spécifications**
- Les scénarios
 - ✓ **Ils regroupent des** cas de test permettant de tester une fonction ou un pan de l'AAT
 - ✓ **Les scénarios se rapportent en général au processus métier**
- Les sessions
 - ✓ **Elles consistent à exécuter** des scénarios de test par des testeurs donnés
- Les Anomalies
 - **Défaut constaté et associé à un cas de test**

La recette fonctionnelle des SI

La Démarche

- 1. Les Spécifications**
- 2. Les exigences**
- 3. Les cas de test**
- 4. Les scénarios**
- 5. Les sessions**
- 6. Les Anomalies**

1) Les spécifications

Le **Cahier des Charges du SI** va servir de support à la recette.

A partir du cahier des charges :

1. Il faut relever toutes les règles (métiers, gestion...) importantes qui sont contenues dans les spécifications du CdC
2. Ces règles sont également appelées **les exigences**
3. Les tests de logiciels auront pour but de démontrer que les produits logiciels issus de la phase du cycle de développement sont conformes à ces spécifications (incluant les exigences)

2) Les exigences

Une exigence correspond à un besoin devant être couvert par l'application à réaliser

Caractéristique d'une exigence

- **Non ambiguë**
 - ✓ Aucune interprétation **ne doit être possible**
 - ✓ **Utilisation de mots précis et non polysémiques**
- **Délimitée**
 - ✓ L'exigence peut être délimitée par des chiffres
 - ❖ Le code produit est à renseigner si l'indicateur « produit local » est à 0 ou 2
 - ✓ L'exigence peut être délimitée par des données présentes dans les objectifs
- **Testable**
 - ✓ L'exigence doit pouvoir être testée via les cas de test
 - ✓ L'exigence peut présenter différents aspects qui doivent également être testables
- **Unique**
 - ✓ Elle ne définit pas pour autant un objectif particulier
 - ✓ Plusieurs exigences peuvent recouvrir un seul objectif

3) Les Cas de Test

Rédaction des Cas de Test

L'objectif de cette phase est de rédiger les cas de test pouvant être organisés en scénarios.

Suivant les besoins et possibilités, ces Cas de Test pourront être plus ou moins détaillés

Exemple:

1. **Nom du cas de test**
2. **Description globale** du test et de la vérification
3. **Description des étapes** sans détail des noms de champs
4. **Description détaillées** des actions avec nom des champs
5. **Description détaillées** des actions avec **nom des champs et valeurs des données**

3) Les Cas de test

Profondeur de rédaction des tests

Objectif

- Optimiser le temps de rédaction des cas de tests
- Définir les profils de testeur pouvant exécuter un cas de test

Niveau de rédaction des cas de tests

- Niveau 1 « **Description** »
 - ✓ Titre et description du cas de test, pas de détail.
 - ✓ Testeur : Connaissance détaillée de cette partie de l'application
- Niveau 2 « **Basis Steps** »
 - ✓ Description des étapes pour exécuter le cas de test, sans détail ni nom de champs.
 - ✓ Testeur : Connaissance générale de l'application

3) Les Cas de Test

➤ **Niveau 3 « Detailed Steps if Time »**

- ✓ Niveau de rédaction 4-Detailed Steps, mais si le projet n'a pas le temps (à justifier), décrire en niveau 2-Basic Steps

➤ **Niveau 4 « Detailed Steps »**

- ✓ Description détaillée des étapes et enchaînements, sans données de test figées
- ✓ Testeur : Lambda, avec fourniture des jeux de données avant exécution

➤ **Niveau 5 « Complete Step »**

- ✓ Description détaillée des étapes et enchaînements avec données de test figées
- ✓ Testeur : Lambda

3) Les Cas de Test

Dans l'étape de rédaction, les Cas de Test doivent contenir les informations suivantes

➤ **Titre**

- ❖ Le titre d'un CdT devrait contenir un verbe d'action (créer, modifier, enregistrer, supprimer...)

➤ **Étapes identifiées**

- ❖ Un cas de test contient une ou plusieurs étapes
- ❖ Chaque étape est identifiée de façon unique

➤ **Actions unitaires**

- ❖ A chaque étape correspond une action utilisateur unitaire
- ❖ L'action est explicite
- ❖ L'action n'est pas interprétable

➤ **Résultats attendus**

- ❖ A chaque Action, un résultat attendu est obligatoire
- ❖ Le résultat est non ambigu et systématique

3) Les Cas de Test

Etapes de la rédaction

Définition

- ❖ Une étape est la brique de base du cas de test.
- ❖ Une étape correspond à une action unitaire de l'utilisateur.
- ❖ Une étape possède un identifiant (Step Name)
 - ✓ préférer un ordre séquentiel
- ❖ Une étape possède une description (Description)
 - ✓ ce que l'utilisateur doit accomplir
- ❖ Une étape possède un résultat attendu (expected result)
 - ✓ ce que l'action effectuée provoque comme résultat
 - ✓ s'il y a anomalie ou non

Stratégie

- ❖ Au maximum **15 étapes** (hors exception) par cas de test
- ❖ Une étape ne comprend qu'une action utilisateur
- ❖ Une étape ne comprend qu'un résultat attendu
- ❖ Il n'y a jamais de conditions (saut, alternative, choix...) dans une étape

3) Les Cas de Test

Actions unitaires

Définition

- ❖ L'action unitaire est la description détaillée d'une action à effectuer sur l'application testée.
- ❖ L'action unitaire est non ambiguë (saisir la donnée...)
- ❖ L'action unitaire est intelligible (si la donnée est saisie, vérifier le choix d'une valeur)

Stratégie

- ❖ Compréhension de l'action
 - ✓ Varie en fonction du profil technique ou fonctionnel du testeur.
 - ✓ Par défaut considérer que le niveau est faible
- ❖ Utilisation de paramètres
 - ✓ Donner des noms explicites aux paramètres. La personne chargée de valoriser les paramètres ne connaît probablement pas le contexte

3) Les Cas de Test

Résultats attendus

Définition

- ❖ Le résultat attendu est la répercussion, dans l'application testée, de l'action décrite.
- ❖ Le résultat attendu n'est pas ambigu (**anomalie, pas d'anomalie**)
- ❖ Le résultat attendu est vérifiable (**Valeur H - Engagements hors bilan présente...**)
- ❖ Le résultat attendu est unique et invariable (**Affichage de PB-Public...**)

Stratégie

- ❖ Le dernier résultat d'un CDT devrait être le révélateur de la vérification la plus importante
 - ✓ Permet d'affirmer que ce test valide bien son objectif
- ❖ Certains résultats peuvent être difficiles à décrire
 - ✓ Quand l'action est de remplir un champ : le champ est rempli
 - ✓ Quand l'action déclenche un processus : décrire le résultat du processus et le temps de latence
- ❖ Un résultat difficile à décrire indique souvent une description erronée

3) Les Cas de Test

Le cycle de vie des CdT

Les Cas de Test doivent être créés, mais ils doivent également être maintenus et détruits (il faudrait indiquer la raison).

L'investissement de rédaction des Cas de Test nécessite de prévoir leur réutilisation dans le test.

- ✓ Exemple : Politique systématique de test de non-Régression permettant de vérifier s'il n'y a pas eu de dégradation des fonctions par rapport à la version précédente

4) Les Scénarios

Les scénarios de test:

- ❖ Ils regroupent des cas de test
- ❖ Ils peuvent représenter une séquence fonctionnelle réelle
 - ✓ Le processus métier de la saisie d'un budget
- ❖ Ils peuvent représenter un contrôle homogène de test
 - ✓ Vérification des contrôles de rejets d'une saisie de budget invalide

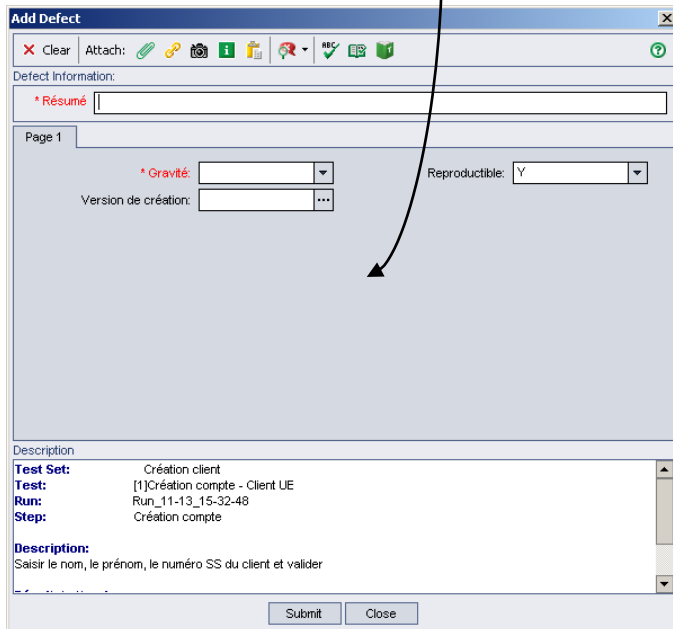
Les scénarios sont créés dans Test Director ou Quality Center (QC):

- ❖ Etablissement de la relation Scénario / Cas de Test
- ❖ Rédaction en Anglais

5) Les Sessions

Le scénario décrit :
les actions à faire (Description)
et les résultats attendus (Expected Result).

En cas de divergence... création d'un défaut



Add Defect

Clear Attach: [Icons]

Defect Information:

* Résumé: [Text Box]

Page 1

* Gravité: [Dropdown] Reproductible: Y [Dropdown]

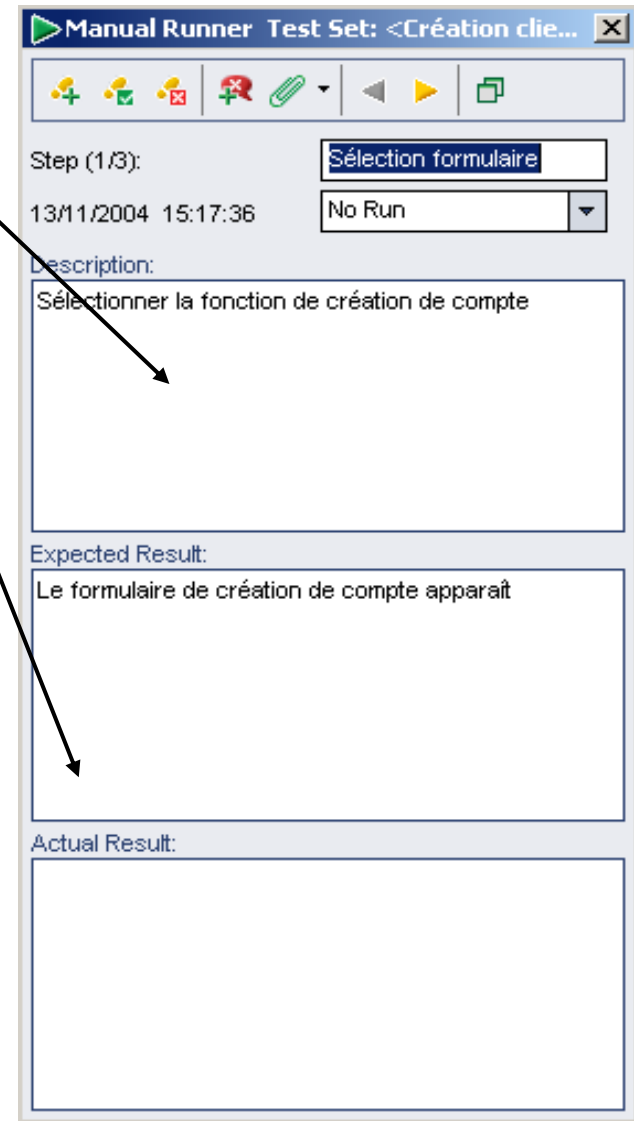
Version de création: [Text Box]

Description

Test Set: Création client
Test: [1]Création compte - Client UE
Run: Run_11-13_15-32-48
Step: Création compte

Description:
Saisir le nom, le prénom, le numéro SS du client et valider

Submit Close



Manual Runner Test Set: <Création clie... X

[Icons]

Step (1/3): Sélection formulaire

13/11/2004 15:17:36 No Run [Dropdown]

Description:

Sélectionner la fonction de création de compte

Expected Result:

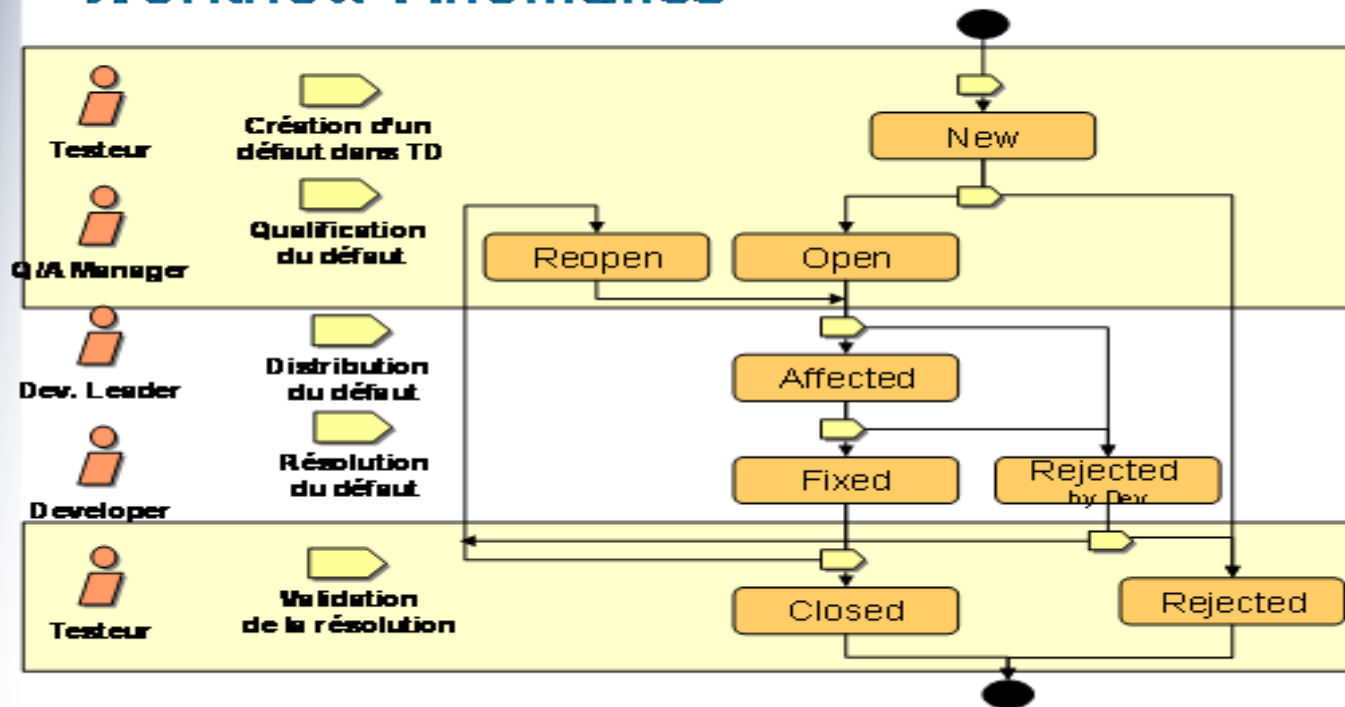
Le formulaire de création de compte apparaît

Actual Result:

6) Les Anomalies

Suivi des anomalies corrigées ou ouvertes durant la session

Workflow Anomalies



La recette fonctionnelle des SI

L' Organisation de la Qualification

Construire le plan de test (critères projet)

Découpage fonctionnel : Matrice de dépendance

- Découpage de l'application en fonctions et sous fonctions
- Création de la matrice de dépendance fonctionnelle
 - Définir pour chaque domaine, les niveaux de dépendances des fonctions (3 niveaux)

Définition des Exigences priorisées

- Identification et Association des exigences par rapport aux fonctions
- Définition de la priorité des exigences
 - Matrice de dépendance + critères fonctionnels

Définition de la liste des Cas de test

- Pour chaque exigence identifier les critères de variance
 - Ajouter les test nécessaires
- Associer les cas de test aux domaines et fonctions (priorisées)
- Définition de la criticité des cas de test
 - Critères liés à la complexité technique ou technologique
- Définition de la profondeur de rédaction
- Identification des données manipulées

Définition des scénarios de test

- **Agréger les cas de test en scénarios en respectant les points suivants:**
 - ♣ Identification des principaux types de données manipulées
 - ♣ Logique de disponibilité des données
 - ♣ Logique d'enchaînement métier
 - ♣ Limiter de préférence à une fonction ou un domaine

Contrôle par les Scénarios de test « Métiers »

- Identification des scénarios métiers supportés ou traversés par l'application
- Vérification de la prise en compte de ces scénarios « Métier » par les scénarios de test
 - ♣ Complément sur les exigences et les cas de test
- Projection des processus sur la matrice de dépendance

Définition des priorités d'exécution sur les scénarios

- Dépendances entre les différents scénarios de test

Quelles sont les **Actions** à mener ?

1. Description des environnements de qualification (les plates-formes à utiliser pour la qualification de l'AAT)
2. Commander le matériel de qualification
3. Réévaluation précise des charges de qualification
4. Détermination des outils de gestion de configuration
5. Evaluation des besoins en formation
6. Préparation des scénarios de tests qui correspondent aux exigences ne pouvant pas être vérifiées sous la responsabilité du réalisateur (les flux sensibles, les aspects sécurité,....)

Qui sont les Acteurs ?

- En premier lieu, on a le **Chef de Projet de Qualification**.

Avec:

- ✓ Le Chef de projet
 - ✓ La maîtrise d'ouvrage
 - ✓ Les experts Qualification
- La Responsabilité revient au **Chef de Projet**.

❑ Les jeux d'essais et de données

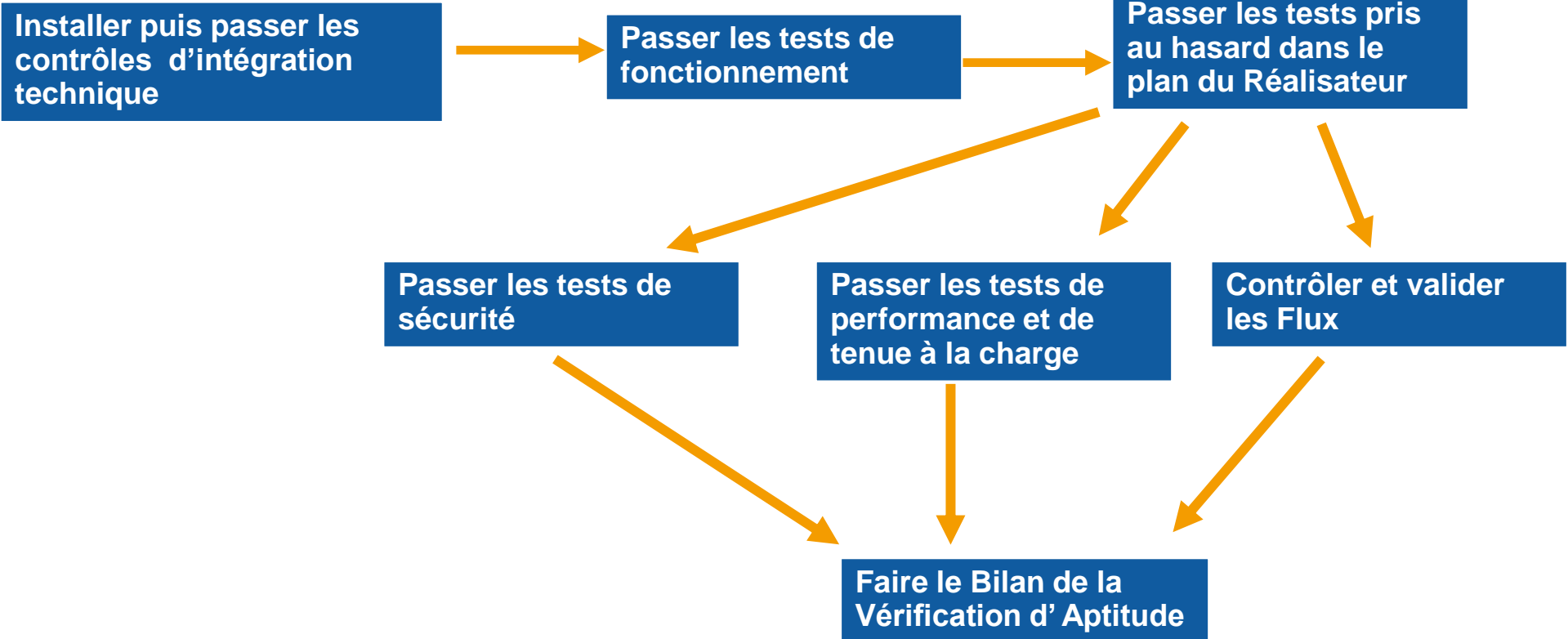
❖ Les jeux d'essais et de données dépendent de:

- ✓ La nature de l'objet à tester
- ✓ L'existence de spécifications liées à cet objet
- ✓ L'environnement de test

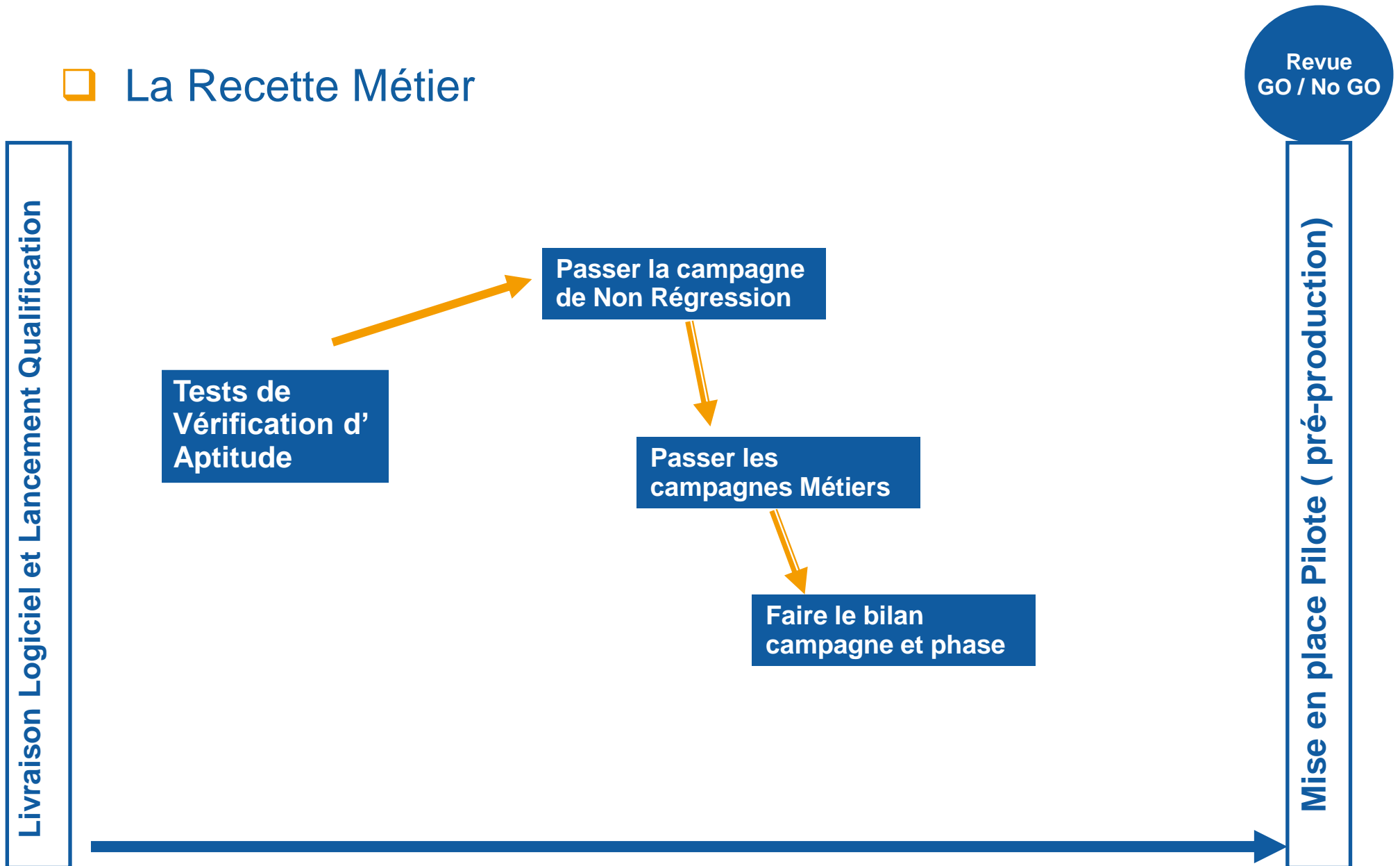
❖ Les jeux d'essais et de données peuvent être issus:

- ✓ Des données de production
- ✓ Des données importées avec des valeurs réalistes
- ✓ Des données saisies par le logiciel en tests

□ La Vérification d' aptitude



□ La Recette Métier



□ La Recette Provisoire (PV)

Revue
GO / No GO

Livraison Logiciel et Lancement Qualification

Tests de
Vérification d'
Aptitude

Tests Métier

Tests de pré-
production /
exploitabilité

PV de Recette
Provisoire

Mise en place Pilote (pré-production)



La recette fonctionnelle des SI

Le Plan de Test

Le plan de test

Le Plan de tests décrit :

- ❖ **Les objectifs de test :**
- ❖ **L'environnement de test**
 - technique
 - Organisationnel
 - Planning
- ❖ **Les critères de priorité et de rédaction des tests**
- ❖ **Le cycle de vie des défauts**
- ❖ **La liste des cas de test**
- ❖ **La liste des scénarios avec leur cas de test associés**

Le plan de test

□ Les objectifs de test :

- **Il est nécessaire de se fixer des objectifs qui permettent de réaliser les tests avec un maximum d'efficacité en garantissant la couverture des objectifs définis**
- **Les objectifs de tests de chaque phase de tests seront identifiés et définis en considérant :**
 - ♣ Les exigences (fonctionnelles, de performance, de qualité, de sûreté, légales, réglementaires...)
 - ♣ Les contraintes (d'implémentation, de coût, de temps...)

Le plan de test

□ *L'environnement de test :*

➤ **Il est un des éléments clés de la réussite des tests**

➤ **Il comprend :**

❖ **L'environnement de l'AAT***

❖ **Les données supportant les tests**

❖ **L'outillage de tests**

✓ Outils de pilotage des tests (ex: Test Director)

✓ Outil de gestion des éléments de test : Cas de test..

✓ Outil de gestion des anomalies

***AAT=Application à tester**

Le plan de test

❑ Ses contraintes:

- **L'environnement de test ne doit pas être perturbé par d'autres équipes.**

Ex: Tests unitaires des développeurs

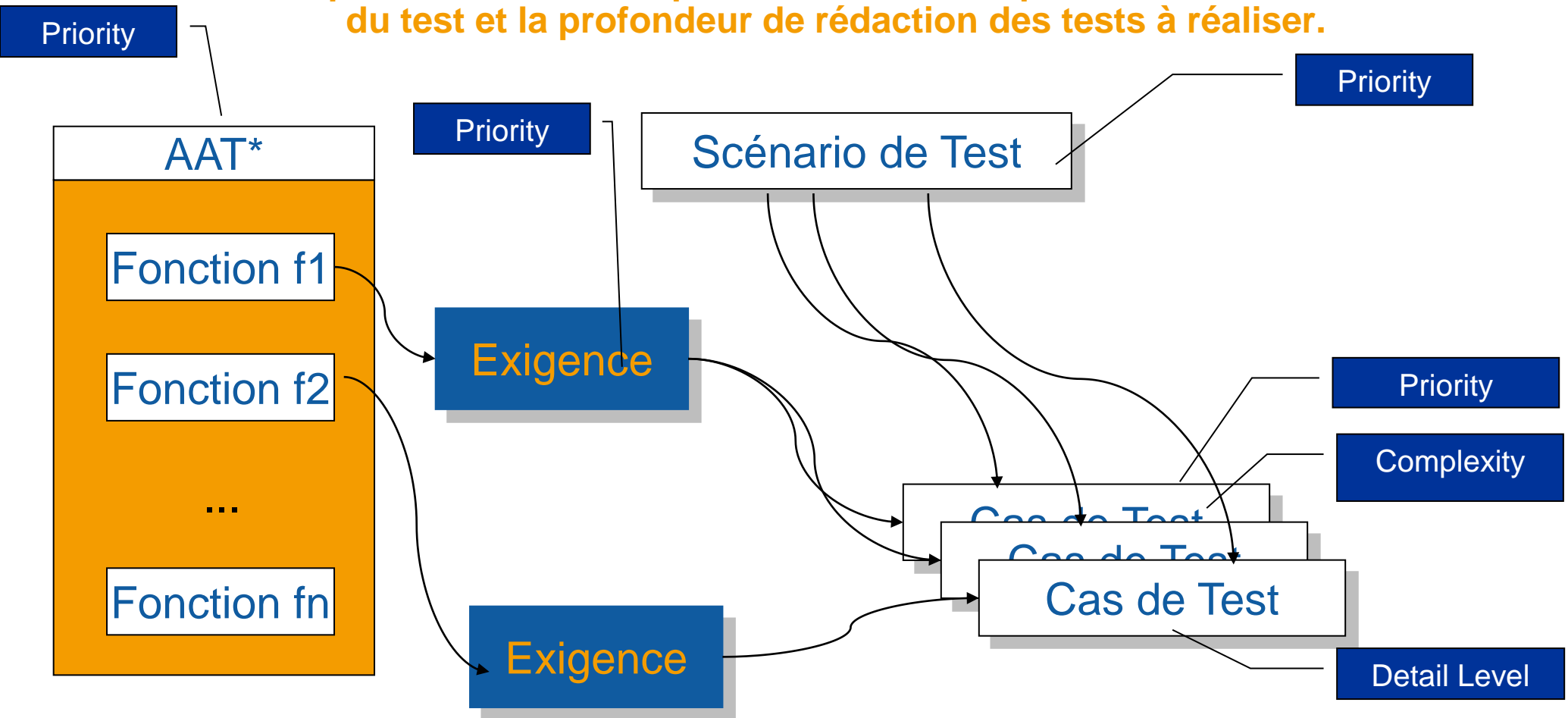
- **Les données de tests doivent idéalement être réinitialisées avant l'exécution des scénarios de tests.**

Afin d'avoir les mêmes données d'origine lors de la comparaison d'un résultat

Le plan de test

□ *Les critères de priorités et de rédaction des tests :*

La phase de définition permet de définir la priorité des éléments du test et la profondeur de rédaction des tests à réaliser.



Le plan de test

Critères des priorités

Objectif des priorités

- ❖ Permet de travailler en time boxing
- ❖ Gestion des risques : Aborder l'essentiel et le plus risqué en premier
- ❖ Optimiser les périodes et ressources de test

Définition des priorités

- ❖ Définir les exigences les plus importantes à contrôler lors de la validation (vision fonctionnelle)
- ❖ Définir les cas de tests les plus importants à contrôler lors de la validation (vision technique)
- ❖ Définir l'importance des scénarios de test ainsi que l'ordre éventuel d'exécution

Le plan de test

Critères de sélection:

- ❖ **L'affectation d'un niveau de profondeur de rédaction dépend des critères suivants :**
 - ✓ **Profil des testeurs disponibles**
 - ✓ **Besoin dans le temps du CdT**
 - ✓ **Cas de test nécessaire en non régression**
 - ✓ **Niveau de la priorité maximum des exigences associées**
 - ✓ **Jeux de données prévisibles**
 - ✓ **Stabilité graphique et fonctionnelle de l'application à tester**





Le plan de test

□ Le cycle de vie des défauts :

❖ **Objectif**











- ✓ L'objectif de la validation logicielle est de vérifier l'adéquation du logiciel aux besoins des utilisateurs (quels qu'ils soient).
- ✓ Durant les phases liées aux tests, des défauts peuvent être détectés.

Acteurs de la gestion des anomalies

Acteur	Groupe	Responsabilité versus anomalies
 Q/A Manager	MOA	Caractérisation du défaut
 Testeur	MOA	Identification des défauts Contrôle des corrections
 Dev. Leader	MOE	Analyse technique des défauts et organisation du travail de correction
 Developer	MOE	Correction des défauts

Le plan de test

Attributs anomalies

Action	Acteur	Informations à renseigner
 Création d'un défaut dans TD	 Testeur	champs TD classiques
 Qualification du défaut	 Q/A Manager	Severity Resolution Priority Default source (MOE, fonctelle, spec, Cdt, env, data...)
 Distribution du défaut	 Dev. Leader	Developer name Estimated Resolution Date
 Résolution du défaut	 Developer	
 Validation de la résolution	 Testeur	

Le plan de test

❑ La liste des cas de test :

Dans l'étape de définition, les Cas de Test doivent contenir les informations suivantes :

➤ Description du cas de test

- ♣ Description générale sans détail
- ♣ Données manipulées dans le cas de test (champs description)
- ♣ Les exigences couvertes par le cas de test
- ♣ Pré-requis pour exécution du CdT

➤ Priorité du cas de test

➤ La criticité du cas de test

➤ Le niveau de détail à utiliser lors de la rédaction du CdT

Le plan de test

☐ La liste des scénarios :

- **La liste des scénarios comprend tous les scénarios rédigés et classés selon une logique d'exécution**
- **Le scénario résulte de plusieurs cas de test**
- **Le scénario décrit les actions à faire et les résultats attendus**