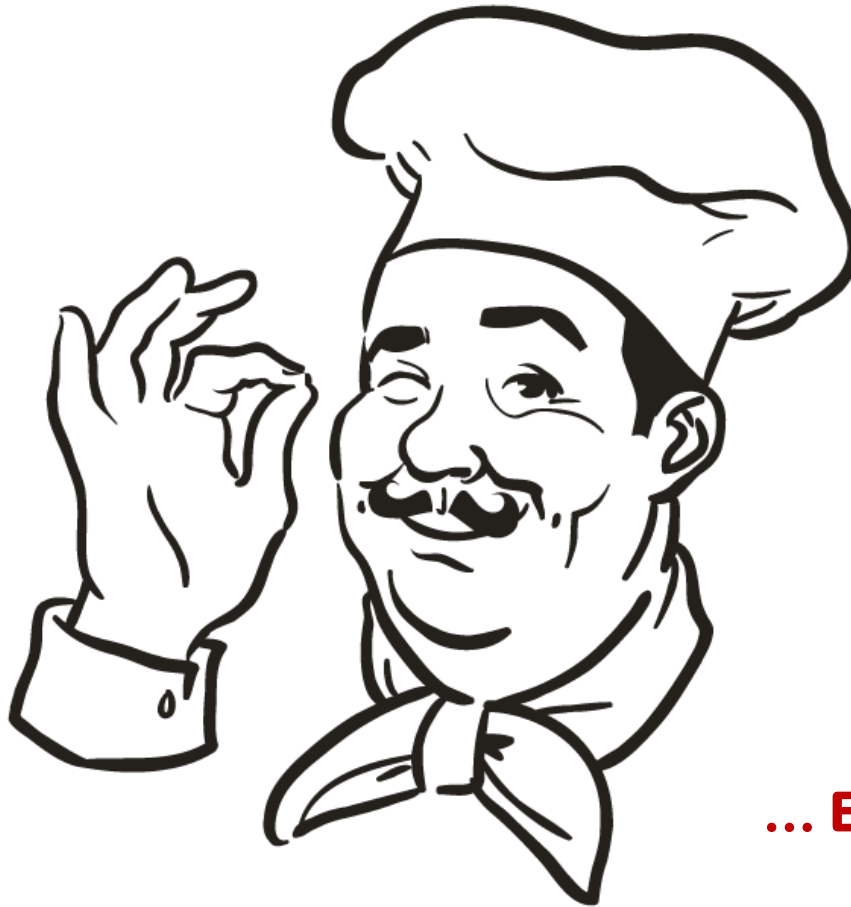




## La recette pour bien tester...



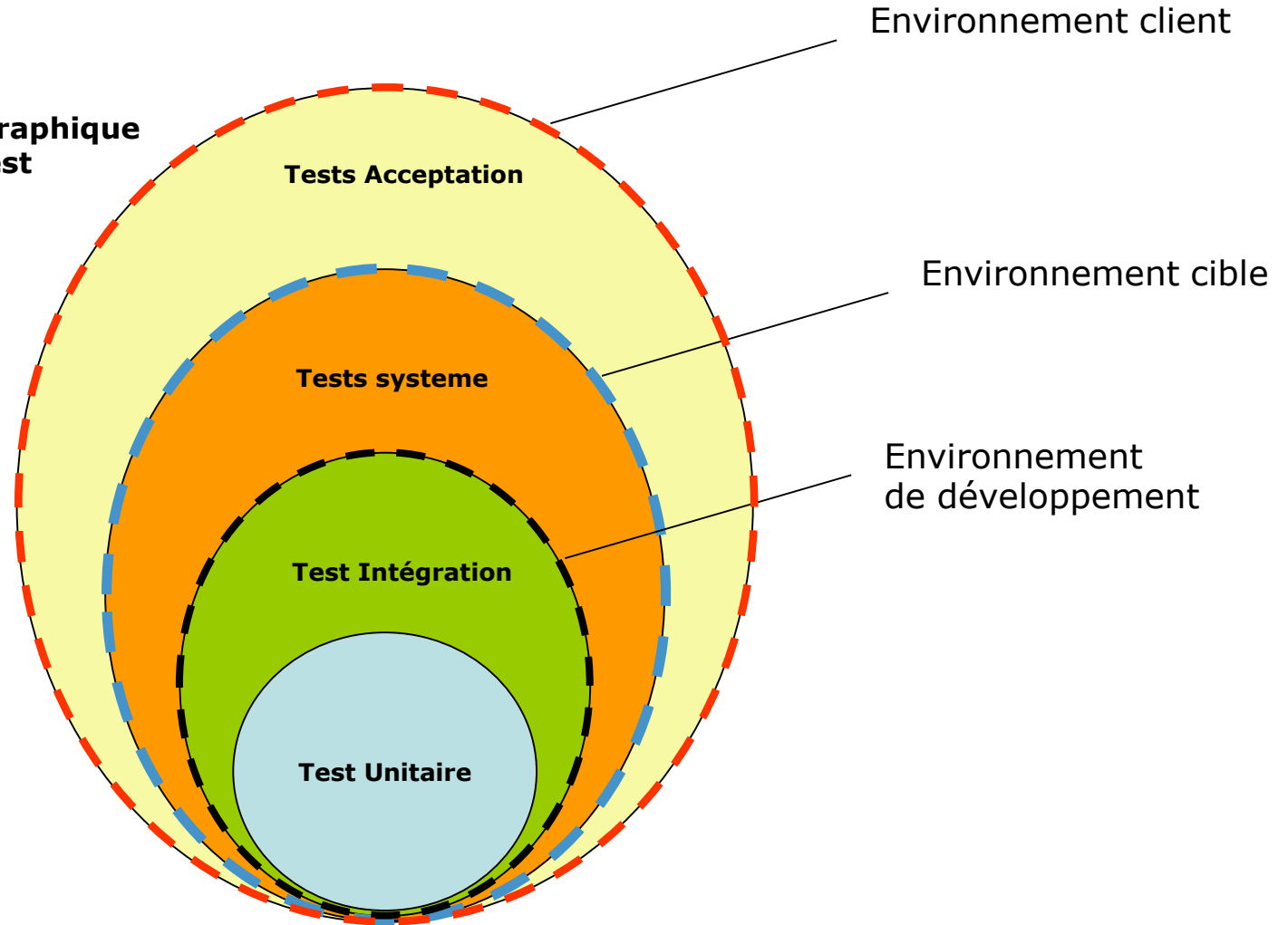
... En 7 étapes

# 1. Une stratégie de tests



## Les niveaux de test

### ■ Visualisation graphique des niveaux de test





## Les thèmes de test

On retrouve dans la norme ISO 9126 : 6 familles de qualité permettant une approche méthodique des niveaux de tests.

- **Capacité fonctionnelle**
- **Fiabilité**
- **Facilité d'utilisation**
- **Rendement**
- **Maintenabilité**
- **Portabilité**



☆☆					
F	R	P	U	S	M
D	C	B	A	A+	N*



## Les thèmes de test

### ■ La Capacité fonctionnelle

Vérifier l'existence et l'exécution correcte des fonctions en utilisation normale

#### **L'aptitude**

correspond à l'usage qu'on fait avec le produit  
(enchaînement, Déplacement dans l'écran entre champs, processus métier dans un contexte client, adéquation contextuelle)

#### **L'exactitude**

correspond au contenu  
(exemple règle de gestion, menu, ecran, Conformité de champs, conformité raccourcis clavier et accélérateurs)

#### **L'interopérabilité**

correspond aux liaisons avec les autres systèmes (interface, intégration du logiciel dans les autres applications - bon fonctionnement du logiciel et des autres logiciels..)

#### **La conformité réglementaire**

la loi

#### **La sécurité**

Vérifier qu'une erreur intentionnelle ou accidentelle ne peut affecter les données ou permettre des actions non prévues ou non autorisées

- Protection des données (intégrité, intrusion...)
- Confidentialité (authentification avec login/mdp)
- Habilitation (profil, droit) Lancement du logiciel, fonctions permises et non permises, données visibles et non visibles, suppression....



## Les thèmes de test

### ■ La fiabilité

**La maturité**      stabilité d'utilisation et d'exécution

**La tolérance aux fautes**

correspond à la robustesse au niveau de l'usage  
(capacité à absorber des incohérences utilisateurs)  
tester la robustesse et le fonctionnement dégradé en cas  
d'arrêt transitoire Fonctions accessibles, informations à  
l'utilisateur, performance, solution de remplacement

**La capacité de récupération** (coté utilisateur)

se rendre compte de ne pas être dans un bon état et agir  
se remettre en état de fonctionner, ou alerter ou corriger  
pour continuer



## Les thèmes de test

### ■ La facilité d'usage

**L'exploitabilité** Toutes acceptation opérationnelle et dans le temps

- la disponibilité
- la simultanéité
- défaillance (CPU, utilisation mémoire)
- tolérance aux pannes – point de vue opérationnel (possibilité de récupération en cas d'arrêt transitoire ou d'arrêt prolongé..)

### **La facilité d'apprentissage**

Dynamique, exemple aide en ligne, système de message, manuel...  
Cohérence avec les autres produits utilisés (forme, fond)

**La capacité de compréhension** statique (vocabulaire)

*Vérifier la documentation c'est vérifier sa forme (Lisibilité, compréhensibilité, cohérence,etc.)  
et son fond (conformité par rapport au logiciel -écran, cinématique...)*



## Les thèmes de test

### ■ l'efficacité

On parle également de rendement ou de performance

**L'efficacité des ressources employées** disponibilité de la ressources, présence, synchronisation et temps de réaction..

### L'efficacité des temps de réalisations

#### la performance

on vérifie l'aptitude à répondre dans un temps donné, dans des conditions normales (pas de dégradation ni de charge)

#### La charge:

On vérifie l'aptitude du logiciel à fonctionner dans des conditions réelles d'utilisation par une analyse de la baisse de performance

#### Le stress

On cherche la limite acceptable voir le déni de service : le but étant ici de prévoir ce qui se passerait ou se passera si on atteint ce niveau (comportement) et quelle serait les indicateurs de suivi pour ne pas atteindre cette limite.







## Les thèmes de test

### ■ La maintenabilité

Ce thème (item) de test est surtout intéressant pour le fournisseur. N'oublions pas qu'un des objectifs de test est la réduction des coûts d'exploitation du logiciel. Son contenu doit donc être de qualité pour intervenir facilement

**La stabilité** (conforme aux normes d'architecture, programmation)

**la facilité de modification** (lié au code, lié à la documentation)

**La facilité d'analyse** (présence de documentation, d'étude technique, norme)

**La facilité à être testé** (lié à la standardisation et à la modularité (objet réutilisable))





## Les thèmes de test

### ■ La Portabilité

#### La facilité d'installation

Valider les procédures d'installation, de configuration et/paramétrage selon les documentations fournies, valider les daemon d'installation, les assistants, les prérequis techniques (configuration). Il est important de valider la première installation, mais également les réinstallation et montée en version/restauration de version

#### La facilité de migration

Changer de version du logiciel et migrer ces données, sa plate forme

#### L'adaptabilité

on est dans le même environnement, mais celui-ci évolue (exemple des patchs microsoft sur windows) ou utilise des système différents pour le même usage (navigateur= Internet explorer ou Firefox)

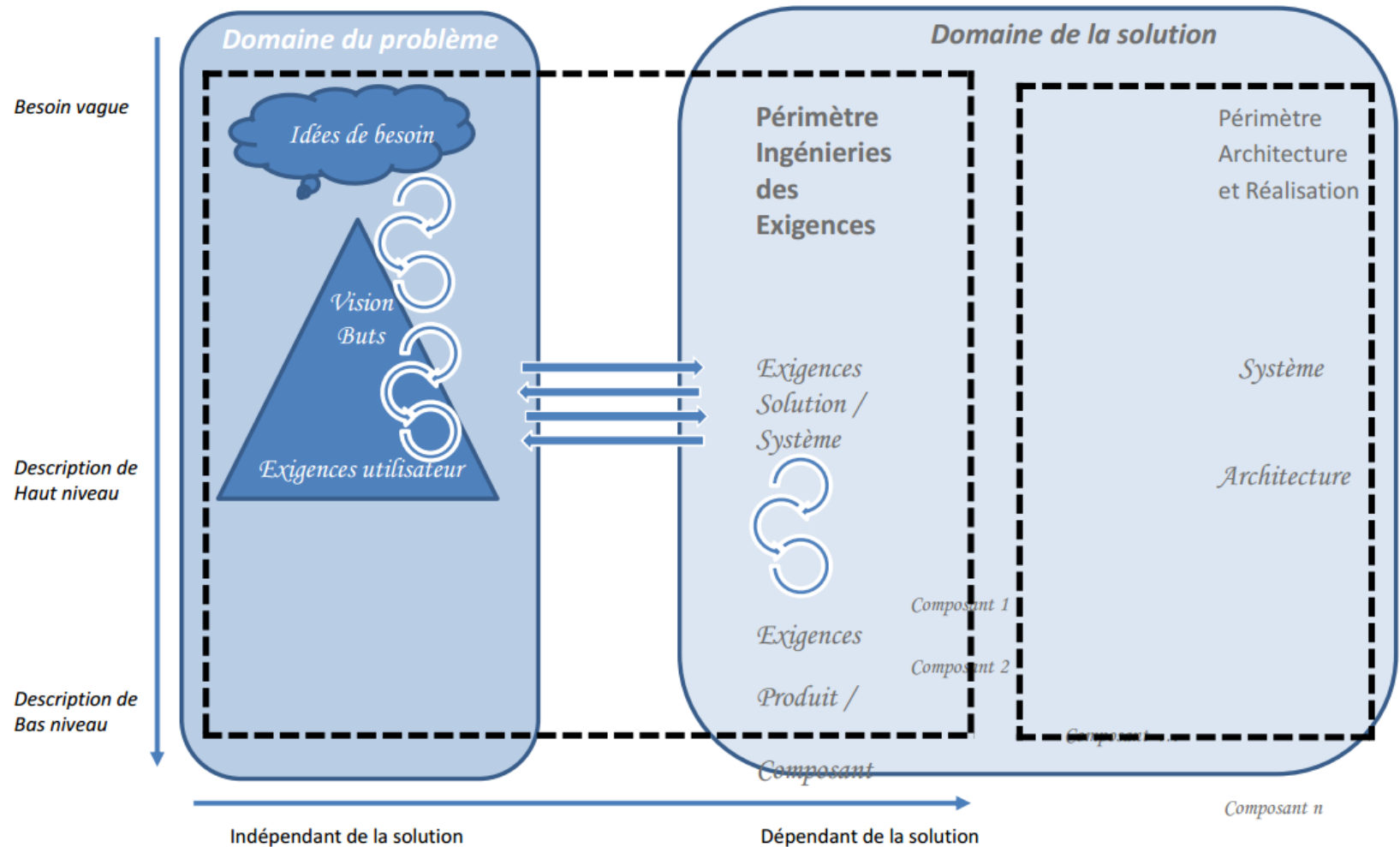
#### L'interchangeabilité :

le contexte change, est ce que l'application fonctionnera toujours (exemple: passer de windows à linux)



## 2. Un Référentiel D'exigences

## ■ Les exigences de test



## ■ Les exigences de test

### Définition d'une Exigence :

- **Condition** ou **aptitude** requise par **l'utilisateur** pour **résoudre un problème** ou atteindre un **objectif**
- **Condition** ou **aptitude** requise qui doit être remplie par un système ou un composant pour **satisfaire un contrat**, un **standard**, une spécification

### Types exigences :

- Fonctionnelle : **CE QUE** le système doit faire (fonction, comportement ...)
- Non fonctionnelle : **COMMENT** le système doit se comporter (attribut qualité)

### Niveaux exigences :

- Métier, Client, Utilisateur
- Système / Solution
- Produit / Composant  
(matériel, interface, logiciel ...)

### Attributs exigences :

- Priorité
- Criticité
- Traçabilité

### Qualité des Exigences :

- Valide, correcte
- Faisable, réalisable
- Utile, nécessaire
- Priorisée
- Nom ambiguë
- Vérifiable, testable
- Unique (tracé)
- Indépendante pour l'implémentation

### Qualité des spécifications d'Exigences :

- Complète
- Cohérente
- Modifiable
- Traçable
- Conforme au standard

**IEEE 830-1993** : Pratique recommandée par IEEE pour la préparation de spécifications d'exigences de logiciel

**IEEE 1233-1998** : Guide de l'IEEE pour la Spécification d'Exigences de Systèmes.

Exemple de référentiel d'exigences



Exemple d'exigence



# 3. Des cas de tests pertinents





## Fiche de test

Procédure: Test de la fonction ...					Testeur
Exigences : EXI-MG2008-..					Date
N° de fiche					
Ordre	Cas de test	Données Actions	Résultat attendu	OK/NOK	Anomalie

Elle permet de recenser:

l'action testée

le résultat attendu

le résultat obtenu

le résultat du test (ok ou ko)

le numéro de la fiche anomalie

Le testeur

Les prérequis...

Elle sera transmise au chargé de test/client pour qu'il déroule l'ensemble des cas de test énoncés.

Elle est importante pour tracer ce qui est fait

Elle est souvent complétée par un tableau d'exécution qui donne l'avancement des tests





## Construire les cas de test

### ■ Graphe à états finis

**Détermination des cas de test à partir d'une représentation des états et changements d'états d'un système**

=>A l'aide d'une matrice de transition

=>A l'aide d'un graphe

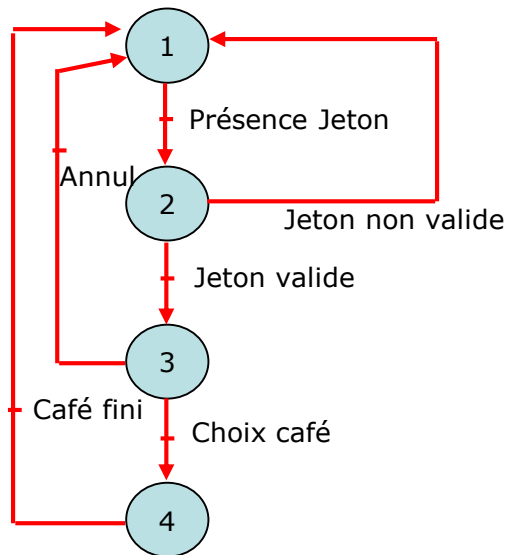
#### **Démarche:**

1. Identifier les différents états du système et les conditions des transitions entre états
2. Etablir la matrice de transitions, puis le graphe d'état
3. Identifier les circuits indépendants à couvrir par des cas de test de telle sorte à couvrir le graphe: **Nb de circuit = nombre d'arcs – nb états + 1**
4. Définir les cas de test (circuit par circuit)

## ■ Un exemple : Machine à café

**Etat de l'automate 4**

1. Accueil client
2. Contrôle Jeton
3. Sélectionneur de café
4. Fourniture du café

**Graphe****Nb arc: 6****Nb Etat: 4****Matrice de transition**

	1	2	3	4
1	X	Présence Jeton		
2	Jeton non valide	X	Jeton valide	
3	Annulation		X	Choix café
4	Café fini			X

**Nb de cas à couvrir:  $6-4+1=3$** **Circuit à couvrir**

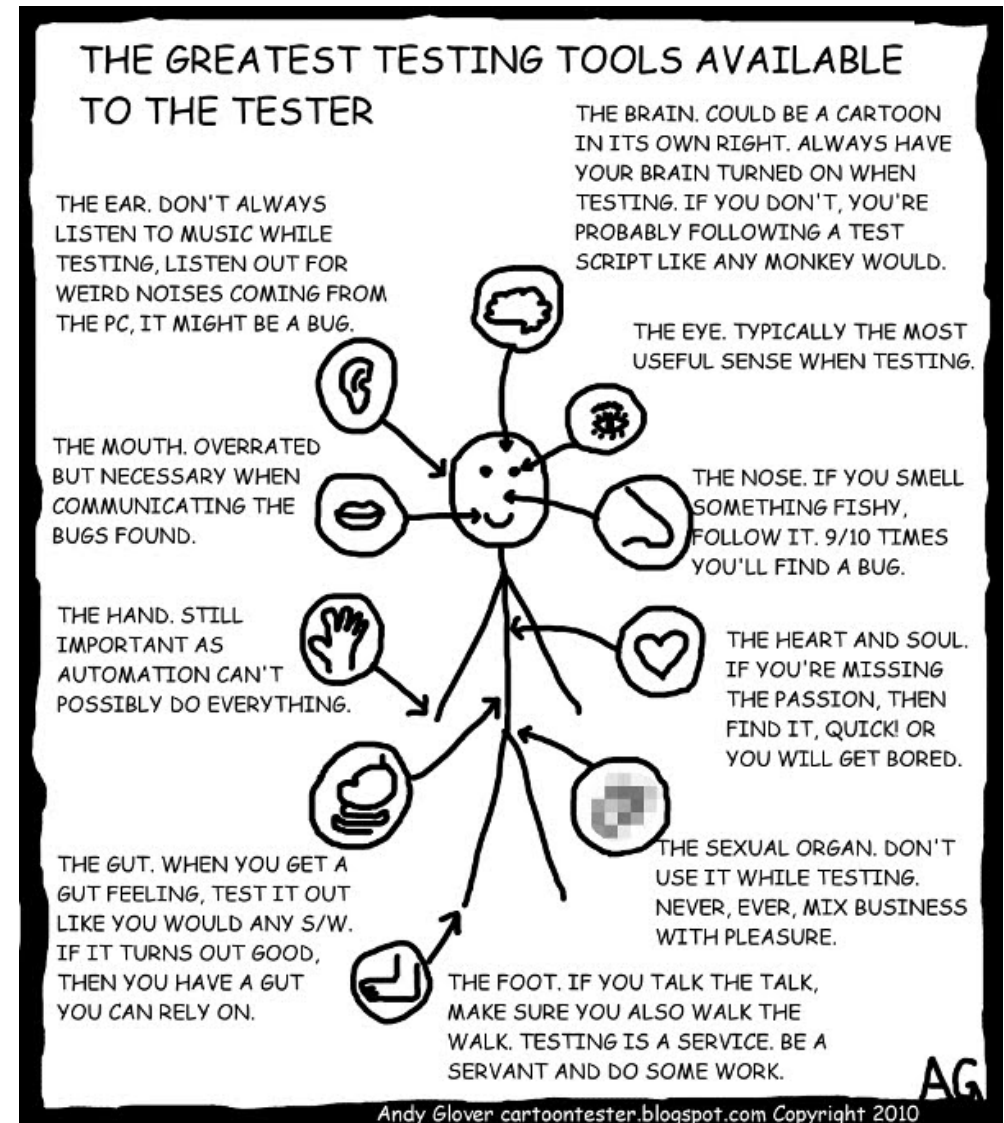
1. 1 2 3 4 1
2. 1 2 1
3. 1 2 3 1

# 4. De bons testeurs



Il y a le bon testeur....

- **RIGoureux**
- **CREATIF**
- **IMPARTIAL**
- **COMMUNiquANT**
- **ORGANISE**
- **EQUIPIER**





## Les métiers du tests

### **Testeur (Bac+3)**

Exécute les tests que des concepteurs ont écrits

### **Analyste de tests (Bac+4/5)**

Conçoit les tests plutôt fonctionnel, s'adapte aux organisations dans lesquelles il est envoyé en mission. Il dispose d'une expérience.

### **Analyste technique de tests (Bac+4/5)**

Définit et prend en charge les tests techniques (interopérabilité SI, performance, sécurité, l'automatisation des tests de non régression, l'analyse statique de code)

### **Gestionnaire d'environnements de tests**

Met en place l'outillage adéquat dans l'organisation.



## Les métiers du tests

### **Consultant tests et consultant senior**

Audite les processus de l'entreprise, propose des axes d'amélioration, prodigue des formations... en centre de test ou chez le client, en forfait ou en régie

### **Chef de projet de tests**

Organise les tests pour un ou plusieurs projets, manage opérationnellement les testeurs, analyste, concepteur.

Rédige la stratégie de tests, coordonne l'activité de son équipe, suit les indicateurs et informe l'entreprise de l'avancement et de la qualité des systèmes en test

### **Responsable Méthodes et Processus de tests**

Met en place la politique de tests de l'entreprise, dirige l'amélioration continue, veille au respect des procédures mises en place en conformité avec le système de management Qualité de l'entreprise.



## Les certifications



# 5. Un banc de test





## Le plan de test

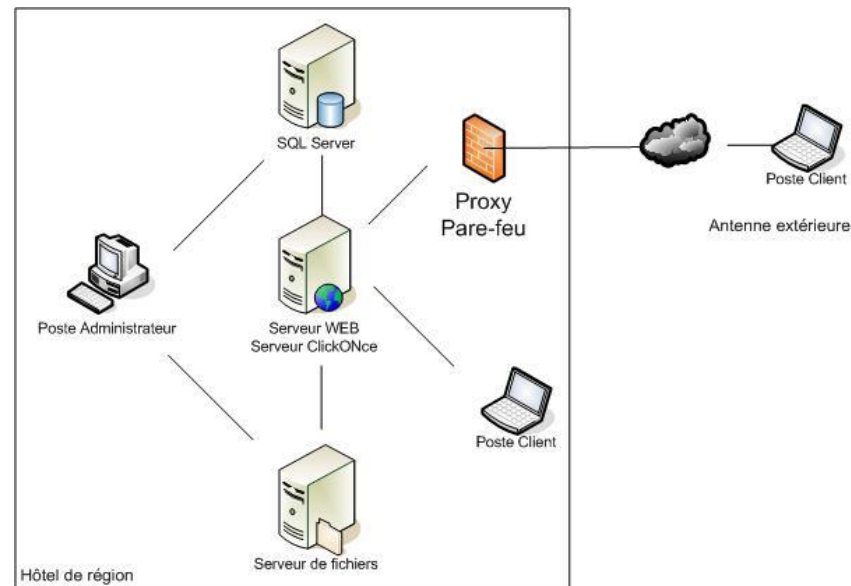
### ■ Environnement de test

**Définir le banc test** c'est :

Préciser sur quelle infrastructure l'application doit fonctionner.

C'est à préciser dès le début du projet afin d'éviter de mauvaises surprises!!! Et orienter les développements sur ce type de plate forme

Anticiper la mise en œuvre de la plate forme



# 6. De bons outils





## Outillage de tests

**Des Outils, mais pour faire quoi?**

Management  
des tests

Exécution des tests

Générateur  
d'environnement

Gestion des défauts



## Outillage de tests

### Comment choisit on un outil de tests ?

#### Facilité d'utilisation

(Enregistreur, Langage de développement, Mots clés)



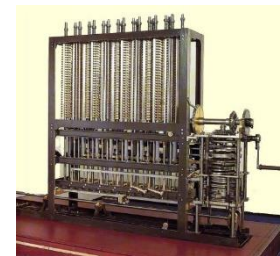
**Intégration** entre les outils  
(Suite logicielle, interopérabilité, ALM)



**Maturité** (obsolescence de certains outils, projet open source)



**Cout** (Rentabilité ne veut pas nécessairement dire Moins Cher, on cherche le ROI)



**Technologie**  
(techno utilisée par le produit et techno sur lequel il opère)



## Outillage de tests

### Management des tests

#### Objectifs

- Gérer les campagnes de tests
- Tracer les exigences et cas de tests
- Définir les pas de tests
- Synthétiser l'avancement des tests
- Référencer, centraliser
- Piloter les tests et les testeurs
- Rendre compte de la qualité logicielle

#### Quelques logiciels

- Testlink
- QaComplete
- HP Quality Center
- SquashTm
- Referty Test
- Xstudio
- Microsoft Test Manager...

Management  
des tests

Exécution  
des tests

Générateur  
d'environnement

Gestion des  
défauts

## Squashtm

Espace Exigences

Bibliothèque Recherche

+

📁

🔍

🔗

🗑️

★ CG38 Projet 32847- Mise en oeuvre C

✓ EXCG38-0010 - Mise à jour Pack et C

✓ EXCG38-0020 - REGLES 1 ET 2 : LIM

✓ EXCG38-0030 - REGLE 3 : L'OBJET D

★ ETUD01

★ ETUD02

★ Mobilité - Territoire

✓ 1. Sécurité

✓ 2. Sélection du territoire

✓ 3. Synthèse territoriale

✓ 4. Détail des aides

✓ 5. Recherche d'aides

✓ 6 - Détail par tiers

✓ 7 - Collaboration

✓ Installation/Désinstallation

★ Prognos - Non Régression

Exigence : 3. Synthèse territoriale

Créé le : 21/02/2013 15:38 (labarre-a)

Modifié le : 07/06/2013 12:38 (labarre-a)

Renommer

Supprimer

Créer une nouvelle version

Informations

Pièces jointes

Informations générales

No de version : 1 [\[Consulter l'historique des versions\]](#)

ID : 16

Description :

La synthèse territoriale permet d'afficher sur le territoire sélectionné :

- la liste des actions et le montant voté par actions

- le graphique d'évolution de l'action sélectionné ou du total sur les 6 dernières années

L'utilisateur peut accéder à la liste et au détail des aides.

Lorsque l'utilisateur arrive sur la synthèse territoriale, il visualise le total des aides versées sur le territoire.

L'utilisateur doit pouvoir effectuer une capture d'écran

L'utilisateur doit pouvoir envoyer un mail depuis le détail des aides.

Référence : (Cliquez pour éditer...)

Criticité : 1-Majeure

Catégorie : Métier

Statut : 1-En cours de rédaction

Cas de test vérifiant cette exigence

Associer des cas de test

Supprimer les associations

#	Projet	Référence	Cas de test	Type	
1	Mobilité - Territoire		<a href="#">Affichage des actions d'une commune avant 0 actions</a>	manuelle	[-]
2	Mobilité - Territoire		<a href="#">Affichage des actions d'une commune sélectionnée</a>	manuelle	[-]
3	Mobilité - Territoire		<a href="#">Affichage des actions du canton sélectionné</a>	manuelle	[-]
4	Mobilité - Territoire		<a href="#">Affichage des informations d'une action</a>	manuelle	[-]
5	Mobilité - Territoire		<a href="#">Affichage des informations d'une action à 0</a>	manuelle	[-]
6	Mobilité - Territoire		<a href="#">Affichage du graphique avec 1 année à 0</a>	manuelle	[-]
7	Mobilité - Territoire		<a href="#">Passage au détail des aides</a>	manuelle	[-]
8	Mobilité - Territoire		<a href="#">Retour à la sélection du territoire</a>	manuelle	[-]

Afficher 50 éléments :

Historique des modifications

30

UBS – Septembre 2017



## Outillage de tests

# Générateur d'environnement

## Objectifs

Fournir un contexte d'exécution proche de la cible attendu

- au niveau du banc de test
- au niveau de la donnée utilisée

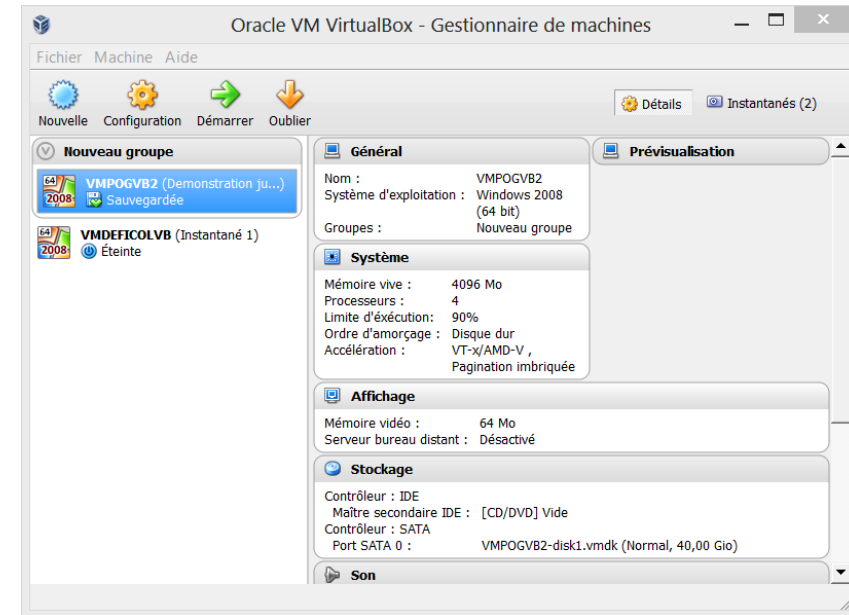
Récupérer les métriques système

## Exemples de logiciels

Oracle VM VirtualBox  
Virtualize  
VMWare  
Virtual System center  
Virtual PC

Credit Card Numbers Generator  
GenerateData.com

Shoonra  
Saucelabs





## Outillage de tests

Management  
des tests

Exécution  
des tests

Générateur  
d'environnement

Gestion des  
défauts

## Gestion des défauts

### Objectifs

Référencer les anomalies rencontrées  
Suivre l'avancement de correction des anomalies  
Etudier les types de défauts

### Exemples de logiciels

Jira  
Firebug  
Mantis  
Bugzilla  
Itracker

The screenshot shows the Jira Issue Navigator interface. At the top, there's a search bar with a query: `project = JIRA AND issuetype in (Bug, Improvement, "New Feature") AND fixVersion = "4.2" AND status in (Resolved, Closed) ORDER BY votes DESC, issuetype DESC, priority DESC`. Below the search bar, it says "Displaying issues 1 to 50 of 123 matching issues." The table below lists various issues with columns for Key, Summary, Assignee, Reporter, P, Status, Resolution, Created, Updated, Due, Votes, and Backlog Order.

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due	Votes	Backlog Order
	JIRA-868	Resolve & Time spent	Unassigned	Primož Piršan		Resolved	Fixed	01/Oct/02	13/Aug/10		281	
	JIRA-14076	DoubleConverter / NumberCFTType is not as 118N as it could / should be - decimal separator does not respect user's locale	Pawel Newiadomski	Brad Baker [Atlassian]		Resolved	Fixed	02/Dec/07	17/Sep/10		12	
	JIRA-21439	Support Transparent Image for Project Avatar	Chris Mountford [Atlassian]	Jack Low [Atlassian]		Resolved	Fixed	27/May/10	11/Aug/10		11	Disabled
	JIRA-21189	View Issue screen custom tabs show fields from all custom tabs, on page load and when Activity tabs are clicked	Unassigned	Rahmani Guler [Atlassian]		Resolved	Fixed	29/Apr/10	28/Jul/10		11	Disabled
	JIRA-20351	AXIS Internal SOAP type representation is unstable.	Brenden Bain [Atlassian]	Brad Baker [Atlassian]		Resolved	Fixed	04/Feb/10	27/Sep/10		8	Disabled
	JIRA-20956	Issue View shows fields of second tab when first opened	Unassigned	Dieter Paul		Resolved	Fixed	09/Apr/10	22/Jul/10		7	Disabled
	JIRA-19557	Thumbnail of certain image attachments fail and cause ERROR in log	Chris Mountford [Atlassian]	Michael Tokar [Atlassian]		Resolved	Fixed	27/Oct/09	11/Oct/10		7	Disabled
	JIRA-15862	Thumbnail in JPEG breaks transparency used in PNG/GIF	Chris Mountford [Atlassian]	Peter de Zwart [Atlassian]		Resolved	Fixed	27/Oct/08	11/Aug/10		7	Disabled
	JIRA-20995	Privilege escalation vulnerability when administrator access is compromised	Unassigned	Edwin Wiong [Atlassian]		Resolved	Fixed	13/Apr/10	20/Apr/10		6	Disabled
	JIRA-20562	JQL breaks issue security levels based on custom fields	James Winters [Atlassian]	Adam Herbert		Resolved	Fixed	28/Feb/10	17/Sep/10		6	Disabled
	JIRA-17759	CAPTCHA image broken when running in OpenJDK	Unassigned	Jeff Turner [Atlassian]		Resolved	Fixed	23/Jun/09	21/Sep/10		6	Disabled
	JIRA-21605	New UI makes issue key very hard to select for copy and paste	Unassigned	Matt Ryall [Atlassian]		Resolved	Fixed	21/Jun/10	23/Jul/10		5	Disabled
	JIRA-21166	Can't select issue summary in issue view screen	Unassigned	John Sloat		Resolved	Fixed	28/Apr/10	02/Aug/10		4	Disabled





## Quand on teste, on trouve des anomalies...

### ■ Remonter les anomalies

A qui remonter les anomalies

- aux chefs de projets
- aux développeurs...

C'est **informer** les bonnes **personnes** d'un dysfonctionnement sur le logiciel pour lequel une **intervention de correction** est nécessaire

Il faut:

S'assurer de la qualité de l'enregistrement

**Son unicité** -> afin d'éviter les doublons et leurs gestions

**Granularité** -> scission de l'anomalie en plusieurs corrections possibles

**Formalisation** à la manière d'un test -> assure une parfaite compréhension

Quand remonter une anomalie:

Plus l'anomalie est découverte **tôt** dans le développement, plus elle sera retourner rapidement au développeur.

Appliquer une **gravité** (BLOQUANT, MAJEUR, MINEUR)



## Quand on teste, on trouve des anomalies...


### ■ Exemple de fiche d'anomalie

Fiche N° 1	Date de mise en évidence: <b>10/11/2008</b>	Testeur: <b>Mr xxx</b>
Nature de l'anomalie: <b>Documentation</b>		Objet concerné: <b>Manuel utilisateur</b>
Description de l'anomalie: <b>Message d'erreur 4 non-conforme à celui du produit</b>		
Reproductibilité: <b>Oui</b>		
Action demandée: <b>Correction du manuel</b>		Réponse obtenue: <b>Correction effectuée le 12/11/2008 par modification de la copie d'écran du message d'erreur</b>
Résolue : Oui <input checked="" type="checkbox"/> Non <input type="checkbox"/>		Visa du testeur: <b>OK</b>



## ■ Exemple de fiche d'anomalie dans Mantis

recette.commeunservice.com/mantis/view.php?id=57

 **MGDIS**®

Connecté en tant que : *labarre-a* (LABARRE Anthony - administrateur) 2013-11-01 09:23 GMT Projet: CRAU - Projet 25531 - PROGOS Aides Web

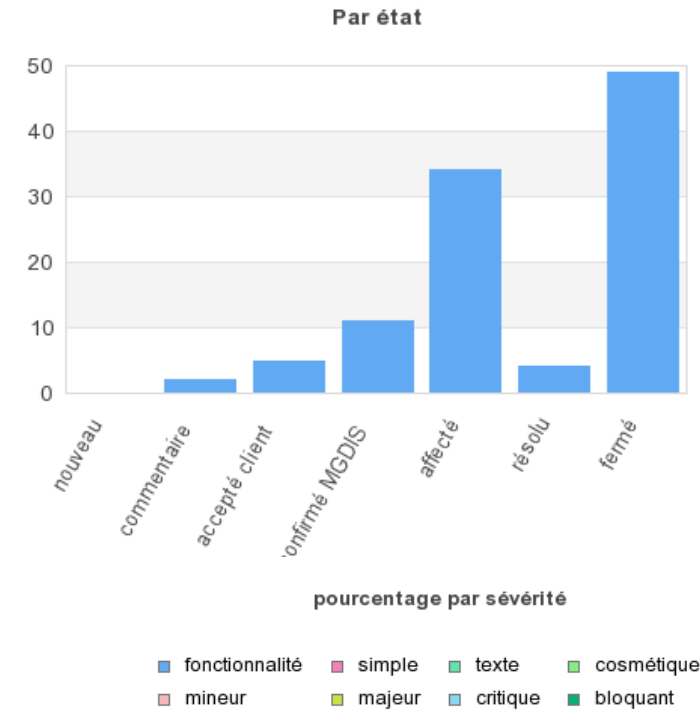
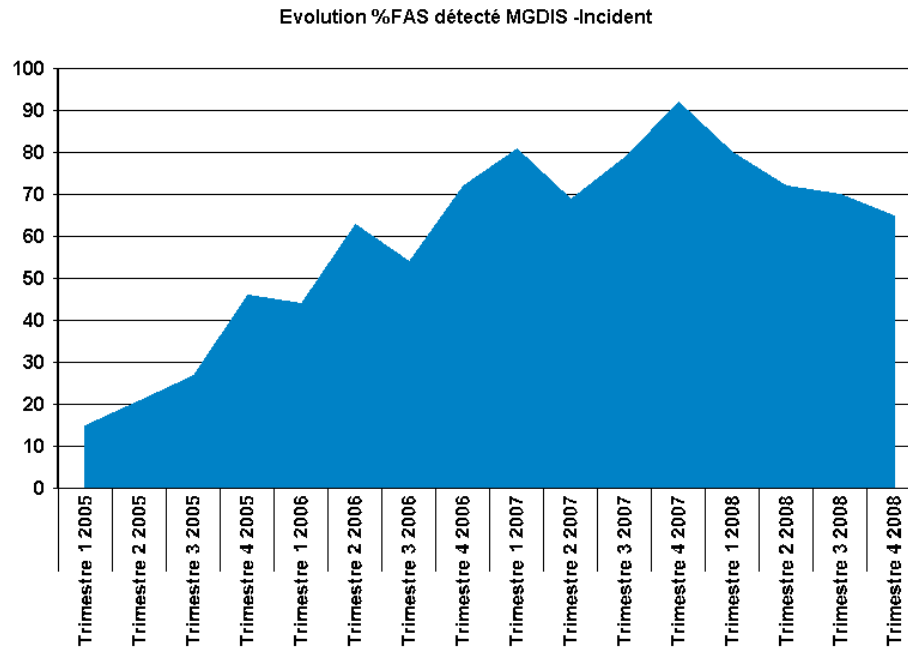
[Accueil](#) | [Mon affichage](#) | [Afficher les bogues](#) | [Rapporter un bogue](#) | [Historique des changements](#) | [Calendrier](#) | [Synthèse](#) | [Administration](#) | [Mon compte](#) | [Fermer la session](#)

Visité récemment: 0000098, 0000097

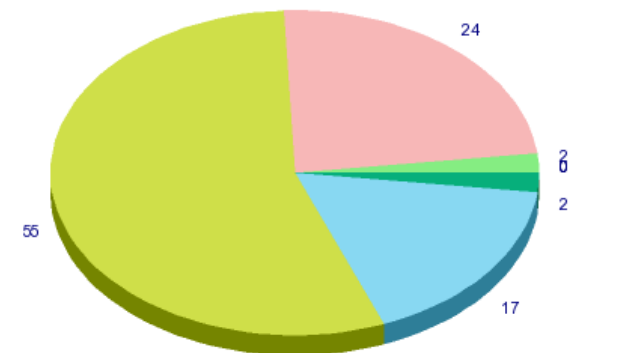
Visualiser les détails du résultat [ [Aller aux notes](#) ] [ [Envoyer un rappel](#) ] [ << ] [ >> ] [ [Historique du bogue](#) ] [ [Imprimer](#) ]

Identifiant	Projet	Catégorie	Afficher l'état	Date de soumission	Dernière mise à jour
0000057	CRAU - Projet 25531 - PROGOS Aides Web	[Tous les projets] Anomalie	public	2013-10-08 13:18	2013-10-08 13:32
<b>Rapporteur</b>	jamette-h				
<b>Assigné à</b>					
<b>Priorité</b>	normale	<b>Impact</b>	majeur	<b>Reproductibilité</b>	Aléatoire
<b>État</b>	accepté client	<b>Résolution</b>	ouvert		
<b>Résumé</b>	0000057: Perte visuelle des actions courriers dans le suivi de dossier (pb pack)				
<b>Description</b>	Perte visuelle des actions courriers dans le suivi de dossier.  Cas 1 : à la 1ère connexion après remonter l'environnement, après création dossier subvention, génération d'un courrier puis suppression coche => perte visuelle de toutes les actions courriers (si on recharge le dossier ou que l'on clique sur icône "masquer éditions", elles réapparaissent => pas systématique cas 2 : suite perte de connexion, à la reconnexion, les icônes courriers n'apparaissent pas => pas systématique				
<b>Étapes pour reproduire</b>	logué : ANDREA en réinitialisant le mot de passe 123456 - création dossier subvention - génération d'un courrier puis suppression coche => perte visuelle de toutes les actions courriers (si on recharge le dossier ou que l'on clique sur icône "masquer éditions", elles réapparaissent				
<b>Balises</b>	Aucune balise n'est associée.				
<b>Associer balises</b>	(Séparer par ',') <input type="text"/> <input type="button" value="Balises existantes"/> <input type="button" value="Associer"/>				
<b>CS installé</b>	aucun				
<b>Domaine</b>	Subventions				
<b>N° Intervention/FAS MGDIS</b>	FAS 2013-1704				
<b>Pack installé</b>	8.20.11				
<b>Résolution sur composant spécifique</b>					
<b>Résolution sur pack</b>					
<b>Pièces jointes</b>					

## ■ Exemple de synthèse dans Mantis



Pilotage de nos défauts  
Visualisation de l'état d'avancement des projets





## Outillage de tests

Management  
des tests

Exécution  
des tests

Générateur  
d'environnement

Gestion des  
défauts

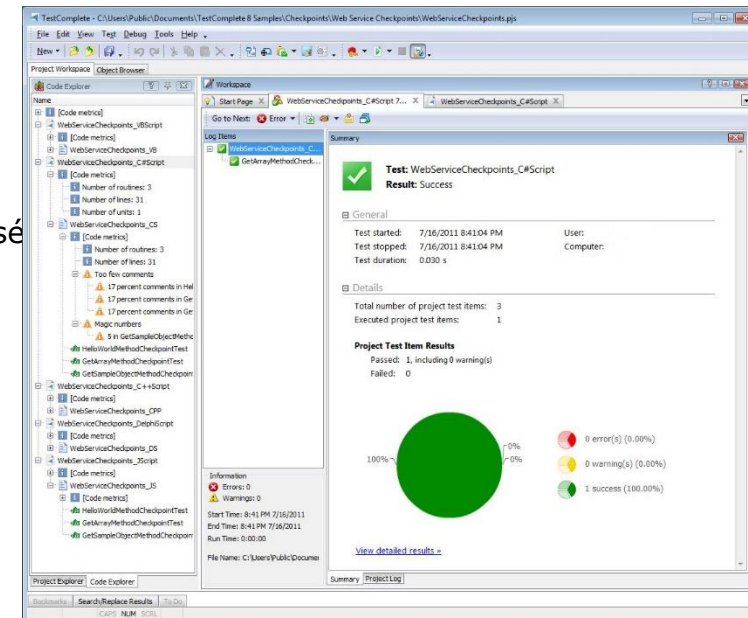
## Exécution des tests

### Objectifs

Jouer et rejouer le scénarii de test de manière automatisée  
Gagner en temps d'exécution de la batterie de test  
Rationaliser et systématiser les tests  
Mesurer la qualité logicielle

### Exemples de logiciels

Testcomplete  
HPQuality Center  
Selenium  
SoapUi  
Katalon  
Jmeter  
Neoload...



## Automatisation des tests

### ■ Pourquoi automatiser

#### Problèmes

Les tests manuels sont appropriés dans certains cas mais restent toujours:

**Longs,**

**Fastidieux**

En **inadéquation avec la brièveté** des cycles de développement actuels

#### Conséquences

Ces inconvénients empêchent de réaliser des tests minutieux  
et laissent passer des bugs, pouvant parfois s'avérer critiques à l'utilisation

#### En outre

Lorsque des applications doivent fonctionner sur **plusieurs plates-formes**, la charge des tests manuels croît proportionnellement en multipliant les risques d'erreurs humaines

## Automatisation des tests

### ■ Avantages

#### ■ Réemploi

- > Réduire les délais de livraison en réemployant des tests déjà existants
- > Déceler et corriger un plus grand nombre d'erreurs plus tôt dans le cycle de développement
- > Découpler les cas de test sans effort (En faisant varier les paramètres d'entrée)

#### ■ Prévisibilité et cohérence

- > Les tests de non-régression permettent de rapidement vérifier que les fonctionnalités de la version précédente sont toujours opérationnelles
- > Et de fournir un retour immédiat aux équipes de développement

#### ■ Productivité

- > Lancer des tests sans surveillance (24/24h 7/7j) et valider simultanément le bon fonctionnement d'une application sur plusieurs plates-formes, navigateurs et environnements.
- > Ces gains de productivité présentent le double avantage de
  - raccourcir les cycles de test
  - De multiplier les opportunités d'amélioration de la qualité logicielle

## Automatisation des tests

### ■ Les erreurs à ne pas commettre

Cependant cette démarche n'est pas évidente à mettre en place, près de 80% des entreprises échouent dans leur démarche d'automatisation pour diverses raisons:

- Coût de développement élevé.
- Maintenance des scripts d'automatisation
- Choix des tests à automatiser.

### **Important**

Il est indispensable de bien cerner les limites des outils d'automatisation et d'identifier les vecteurs qu'il est judicieux et rentable d'automatiser



*L'application de jeu et rejeu miracle n'existe pas.  
il ne faut surtout pas chercher à refaire une application, il faudra elle aussi dans ce cas la tester!!!*



## Automatisation des tests

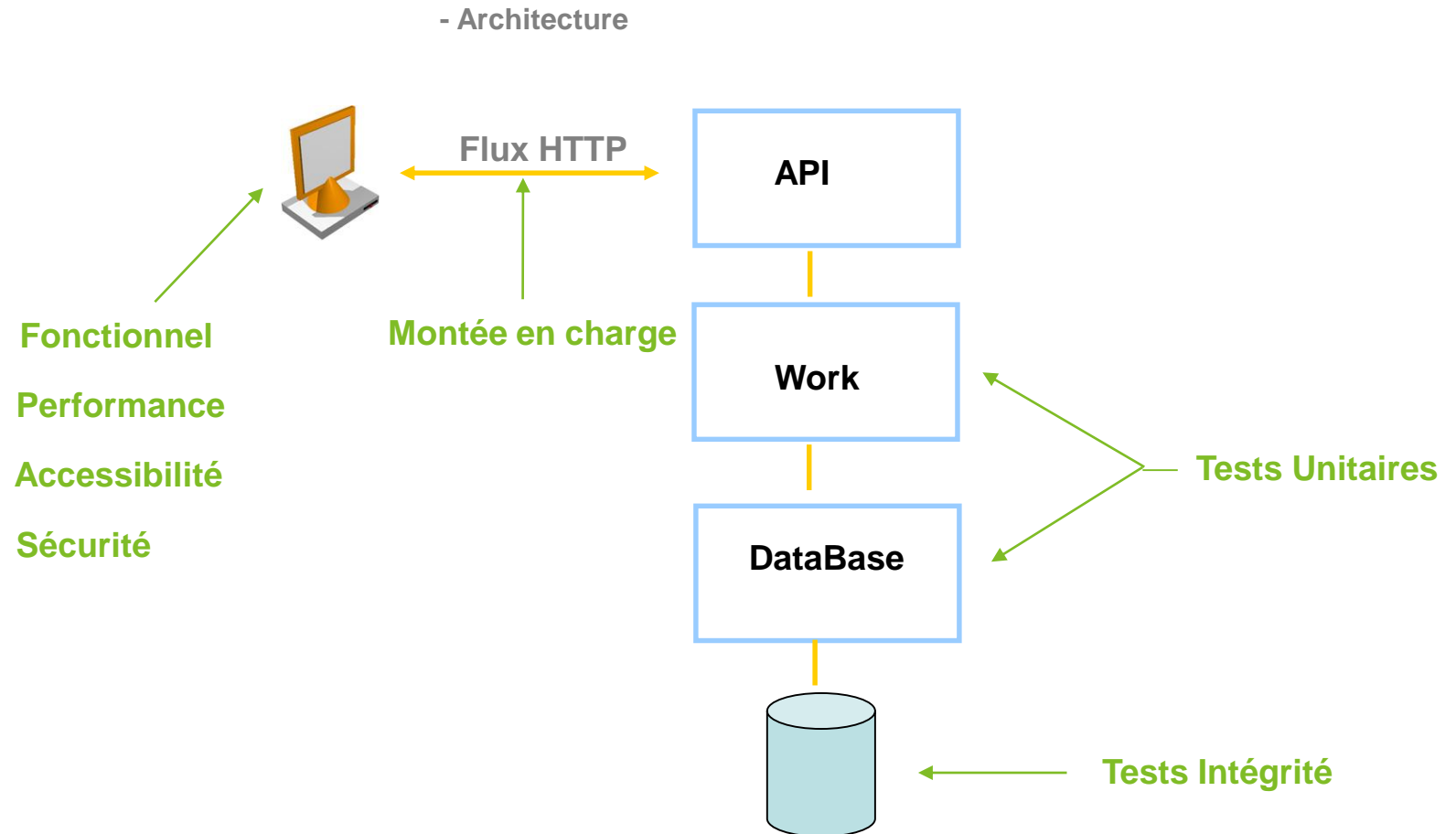
### ■ Les bonnes pratiques

Quelque soit l'approche utilisée pour automatiser les tests, certaines pratiques sont conseillées.

- Rédiger des **plans de test** avant d'automatiser.
- **Analyser** quels sont les tests à automatiser.
- **Planifier** les campagnes.
- **Centraliser** et réutiliser les tests le plus possible.
- **Ne pas enchaîner les tests**, si un échoue les autres échoueront certainement.
- Remonter et analyser les bugs trouvés avec un bugtracker.

## Focus sur l'automatisation les tests

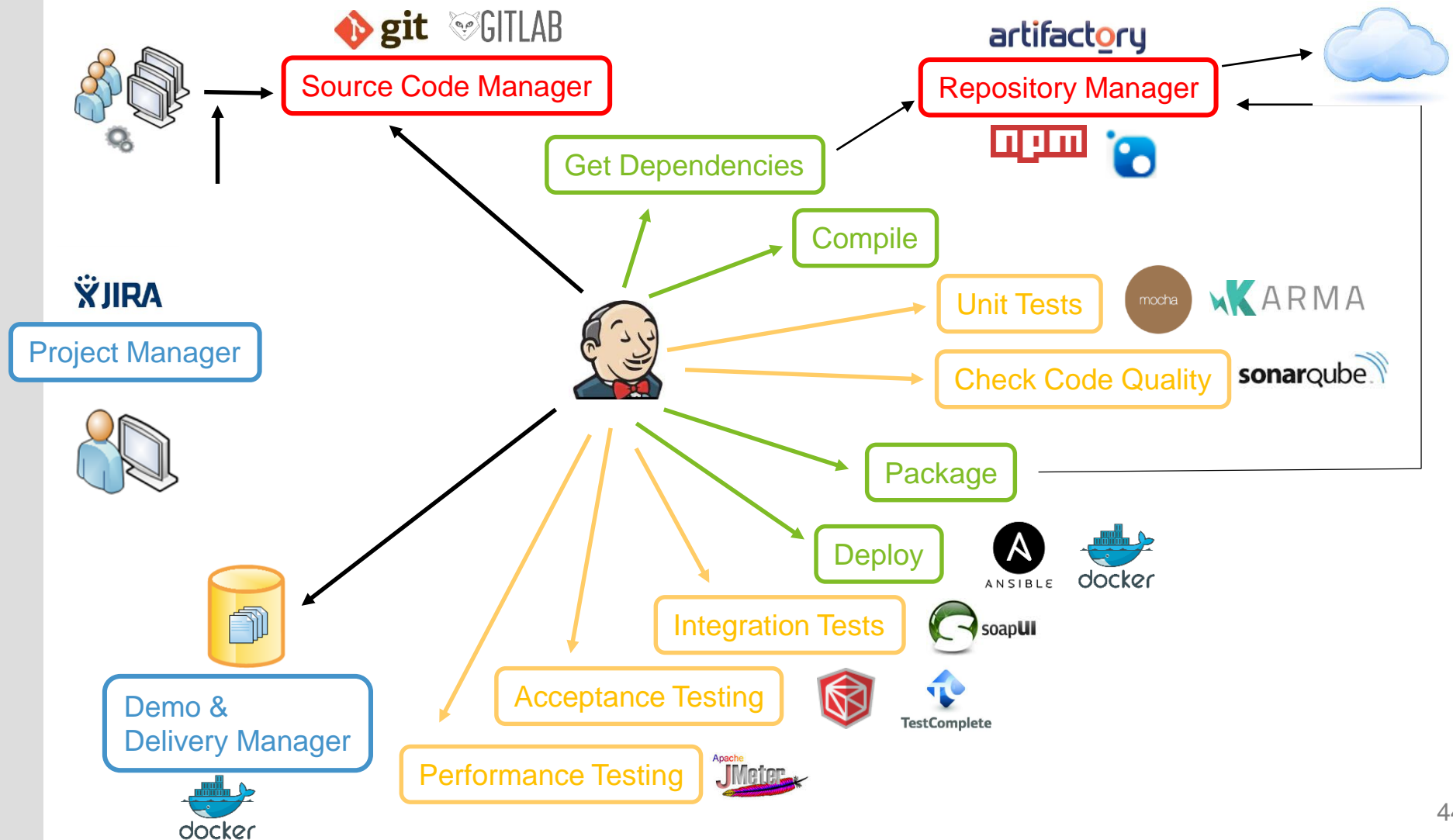
### ■ A chaque tiers son test



# 7. De bons outils



## ➔ Pour construire, il faut une usine de Production logicielle



# Usine de tests



Exécution de scénario de tests utilisateur

Gestion des tests de montée de version



Tests SOAPUI des Apis exposées par les services du pilotage des aides avec une approche basée sur les business-case



Sécurité: Détection des principales vulnérabilités Owasp



Risk Level	Number of Alerts
High	0
Medium	0
Low	3
Informational	0

	deploiement version précédente	soapui version précédente	deploiement version n	soapui version n	testcomplete	DEPOT_ET_INSTRUCTION	GESTION_DE_COMPTES	ESPACE_AGENT_GLOBAL	RATTACHEMENT	soapui	ZAP	jmeter
PORTAIL_DES_AIDES_prio_Inte4466	19min 52s	53s	10min 8s	1min 44s	11s	2h 10min	9min 24s	29min 40s	27min 9s	17min 14s	5min 37s	3min 57s



Entièrement automatisé sur Jenkins

## → En production, on continue de tester

