

Méthodologie de production logicielle

Le test : enjeu de qualité, de sécurité, de satisfaction client et de pérennité de l'entreprise



Anthony LABARRE



Anthony LABARRE



21 ans d'expérience à MGDIS, dont 15 en tests et qualité logicielle

Responsable du pole Assurance Qualité Logicielle

Certifié ISTQB Niveau Fondation

Certifié ISTQB Agile Tester

Certifié ISTQB Niveau Gestionnaire Avancé

Certifié ITIL

Certifié REQB

Club MforTest




Membre MforTest

Intervenant UBS sur Qualité logicielle



 <https://fr.linkedin.com/in/anthony-labarre>

 labarre-a@mgdis.fr



Que va-t-on voir dans ce cours?

- > Le test : c'est quoi et surtout ca sert à quoi?
- > Les objectifs des tests
- > Quand tester?
- > Le processus de tests
- > Le plan de tests
- > Les niveaux de tests
- > Les thèmes de tests
- > Une usine de production logicielle
- > La gestion des incidents
- > Les métiers du test
- > Les outils de tests
- > Focus sur le tests de charge
- > Terminologie



Les Bugs, une histoire sans fin...

9/9

0800 Antares started
 1000 " stopped - antares ✓
 1300 (032) MP-MC 1.98367000 9.037 847 025 correct
 (033) PRO 2 2.13047645 4.615925059(-2)
 correct 2.13067645
 Relays 6-2 in 033 failed special speed test
 in relay 10,000 test.

1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.

1545 Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630 Antares started.
 1700 closed down.





Les Bugs, une histoire sans fin...

Clubic Pro > Réseaux sociaux et sites communautaires > Facebook

Facebook corrige un bug responsable de 50% des plantages de son appli iOS

Publié par **Alexandre Laurent**
le jeudi 14 août 2014

Facebook affirme avoir identifié et corrigé un bug responsable de plus de la moitié des plantages observés sur son application iOS. Il explique également pourquoi la résolution du problème a duré plusieurs mois.

Certains bugs ont la vie dure. Les ingénieurs de Facebook indiquent avoir résolu cette semaine un problème technique localisé depuis plusieurs mois au sein de l'application iOS dédiée au réseau social. Ce bug, limité aux terminaux Apple, était d'après eux à l'origine de plus de la moitié des plantages enregistrés par l'application. Soulagés sans doute par la fin de cette longue traque, deux des développeurs impliqués **livrent** un éclairage technique sur la source du problème, et expliquent pourquoi il a fallu aussi longtemps pour le résoudre.

Dans un premier temps, ils déclarent avoir constaté dernièrement que la première cause de crash observée au sein de l'application iOS se situait au niveau du **framework Core Data**, qui pour simplifier permet au développeur de

← → ↺ www.zdnet.fr/actualites/changement-d-heure-nouveau-bug-sur

t d'heure : nouveau

7

roisième année, le système d'exploitation . bug au moment du changement d'heure. nanifeste dans le calendrier avec un e actuelle.

ctobre 2013

ient d'heure ne réussit pas à iOS. C'est la) que l'OS Apple subit un bug au moment rer ou l'heure d'été.



munauté des experts sécurité | en savoir plus >

DEMANDEZ AUX EXPERTS

LE COIN DES

logies Conformité & Bonnes pratiques Carrière Cyber Po

Microsoft paie 100 000\$ à un chasseur de bugs

En bref

Jerome Saiz le 11 octobre 2013 - 10:54, dans la rubrique **Produits & Technologies**

Aucun commentaire, soyez le premier à participer !

exploit - microsoft

Microsoft vient de verser, pour la première fois, une récompense de cent mille dollars à un chercheur en sécurité informatique. Cette récompense s'inscrit dans le programme de « bounties » **annoncé avant l'été** par le géant.

3 Paris Ile-de-France
Toutes les régions

Dans l'actu : PSG | Paris le Club | Faits-d

santé

Un bug informatique à l'origine du décès d'une patiente à Versailles ?

Les logiciels médicaux utilisés par les hôpitaux sont-ils fiables ? Un rapport d'experts pointe des dysfonctionnements aux conséquences parfois très graves. Plusieurs cas suspects ont été signalés dont la mort d'une patiente il y a 2 ans, à Versailles.

Pôle Internet | Publié le 10/07/2013 | 18:49

Modifi



GARE MONTPARNASS

→ **Ce que nous voulons en tant que client**

Un produit qui répond parfaitement à vos besoins ?
Un planning de mise en production très serré ?
Un produit 0 défaut ?



Et pourtant ! Voici ce qu'un développeur produit:

6 à 10 bugs
pour 1000 lignes de code!





Pour limiter ces bugs, il existe une solution

TEST

Méthodes

Hommes

Outils



Un bug c'est quoi

Erreur (Error)

Action humaine qui a pour résultat l'introduction d'un défaut dans le logiciel



Défaut (fault – defect)

Ce qui est produit dans un logiciel et qui ne devrait pas s'y trouver



Défaillance ou Anomalie (Failure)

Comportement d'un logiciel observé, différent du comportement attendu ou spécifié



Un bug c'est quoi

■ Les catégories d'erreurs

- Spécifications incomplètes ou erronées
- Interprétation erronée du besoin du client
- Déviation intentionnelle des spécifications
- Non respect des standards de programmation
- Représentation erronée des données
- Module d'interface inconsistant
- Défaut en conception détaillée
- Tests défaillants ou incomplets
- Documentation incomplète ou imprécise
- Défaut de programmation
- IHM inconsistante ou ambiguë
- Autre défaut

IES
MCC
IDS
VPS
EDR
IMI
EDL
IET
IID
PLT
HCI
MIS

Principales causes de la majorité des défauts



Tester c'est quoi et surtout ca sert à quoi?



- Exécuter un logiciel afin de trouver des défaillances
- Exécuter un logiciel pour fournir un niveau de Qualité
- Exécuter un logiciel pour apporter une confiance
- Analyser un logiciel pour prévenir les défauts



Le test: Une définition Complexe

Le **test** est l'**exécution** ou l'**évaluation**
d'un **système** ou d'un **composant**,
par des moyens **automatiques** ou **manuels**
pour **vérifier** qu'il répond à des **spécifications**
ou identifier les différences entre les **résultats attendus** et les
résultats obtenus

IEEE (Standard Glossary of Software Engineering Terminology)

Dans le but de **satisfaire le client**
Dans un modèle **économique viable**



Importance des tests

■ Cas pratique pour comprendre l'importance des tests

Quel est le coût d'une anomalie (en j/h) ?

En phase
d'identification

2 heures

En phase
de conception

1 heure

En phase
de développement

3 heures

= 30 heures

En phase
de mise en oeuvre

5 heures

En phase
de maintenance

19 heures

Cette anomalie n'a un impact que sur l'économie de l'entreprise. On ne compte pas ici les coûts non chiffrables comme la non satisfaction du client!!!

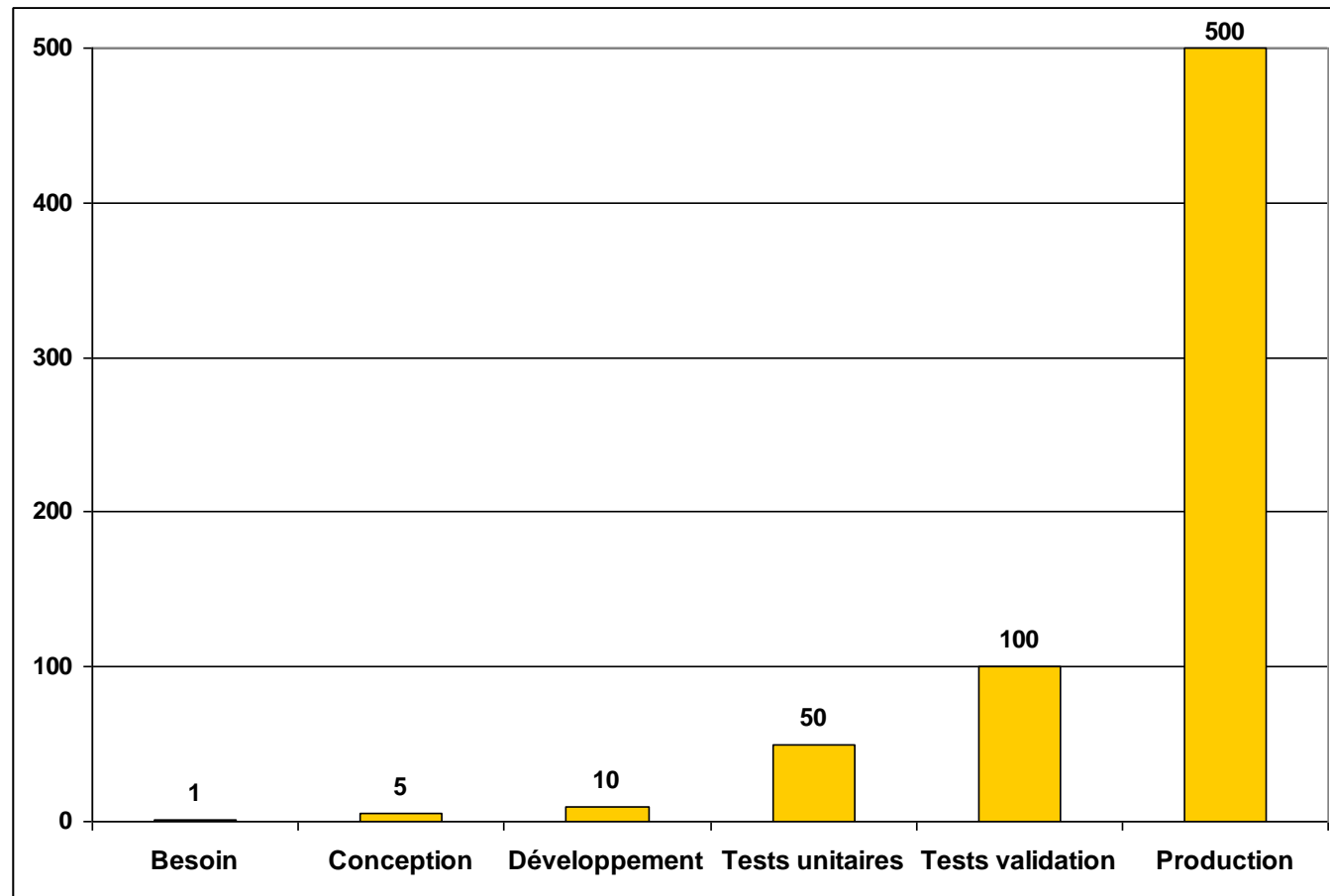
Des situations pires peuvent exister provoquant notamment des dégâts matériels et humains



Importance des tests

■ Coût relatif du défaut

Plus une anomalie est détectée tardivement, plus elle coûte cher





Importance des tests

Trouver le bon équilibre entre le cout du test /
Bénéfices de test et les risques



Enjeux:

Assurer la pérennité de l'entreprise

Garantir l'image de marque

➔ Objectifs des tests

■ Objectif n°1:

SATISFACTION DU CLIENT

Pour atteindre cet objectif il faut être capable de répondre à deux questions :

- QUI EST LE CLIENT ?
- QUAND EST-IL SATISFAIT ?

Qui est le client : tous les utilisateurs directs et indirects de la solution développée

Quand est-il satisfait : Quand il reçoit un produit conforme à ses besoins,
à la date prévue et avec le sentiment de payer le juste prix.

Il est donc très important d'envisager les tests dès la phase de conception

➔ Objectifs des tests

■ Objectif n°2:

LA QUALITÉ DU PRODUIT

- le **contenant** (le logiciel ou progiciel)
- le **contenu** (code source, technique de développement ...)

On parle bien de qualité et pas de sur-qualité

➔ Objectifs des tests

■ Objectif n°3:

MINIMISER LES COÛTS

Coûts de production (fabrication du logiciel)

Coûts d'exploitation (maintenance et support du logiciel)

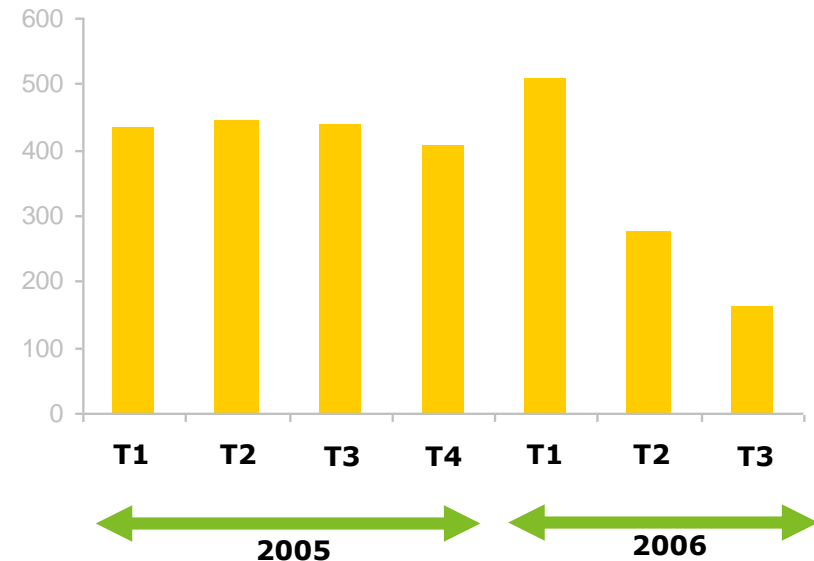
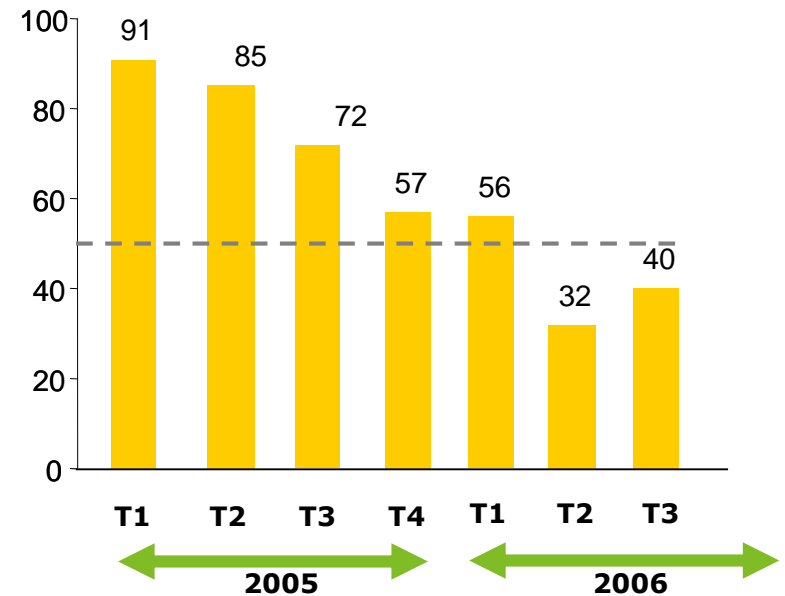
Exemple à MGDIS

% FAS détectées
par nos clients

Phase d'effort consentie par une systématisation
- des tests unitaire (ODT)
- des tests fonctionnels avant livraison

=> **baisse significative du nombre
d'interventions**

Intervention de
maintenance



➔ Objectifs des tests

■ Objectif n°4:

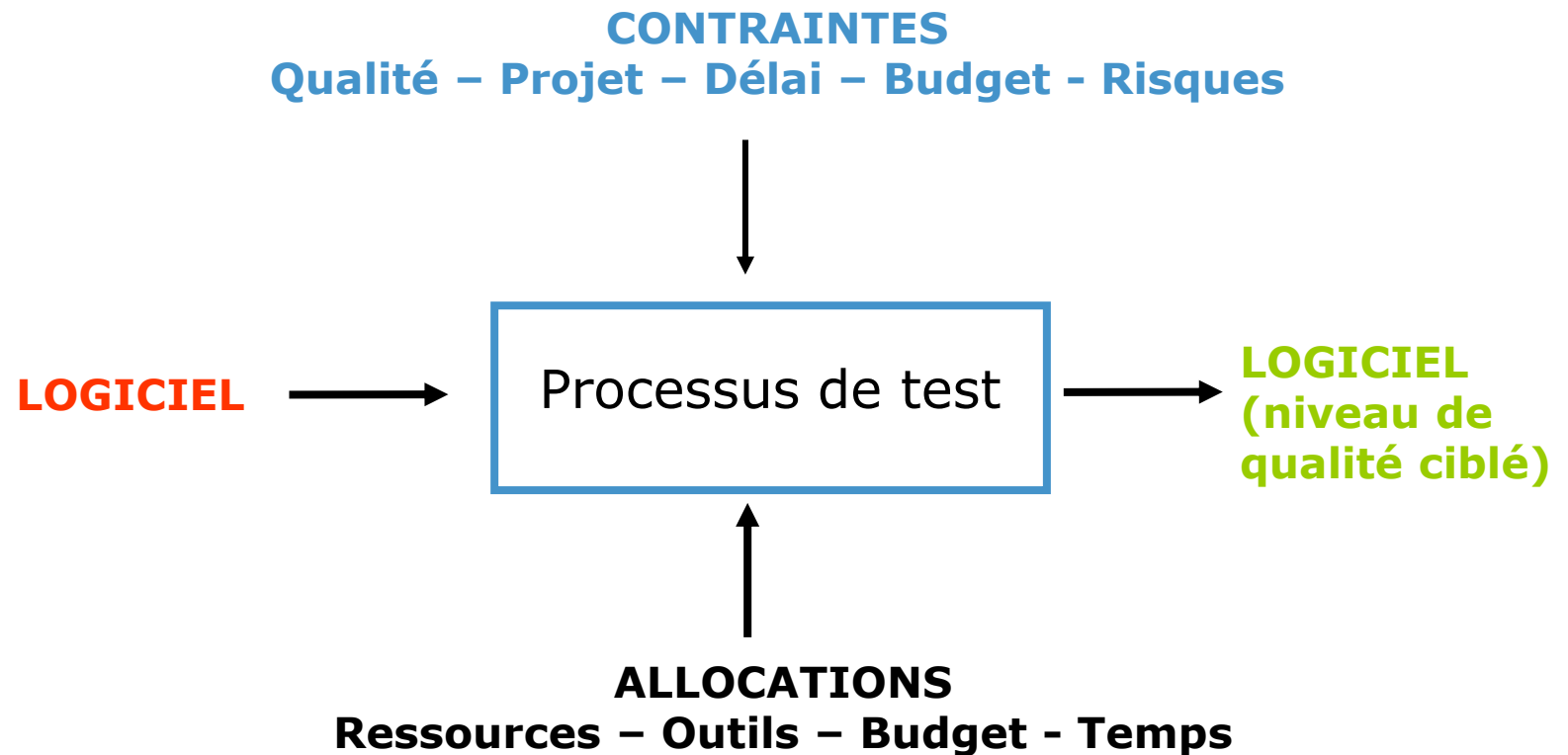
RESPECTER LA LOI

La loi du 19 mai 1998 stipule :

- Un produit est défectueux « lorsqu'il n'offre pas la sécurité à laquelle on peut légitimement s'attendre »,
- Le producteur est responsable du dommage causé aux personnes et aux biens par un défaut de son produit et ce pendant 10 ans à compter de la mise en circulation du produit.

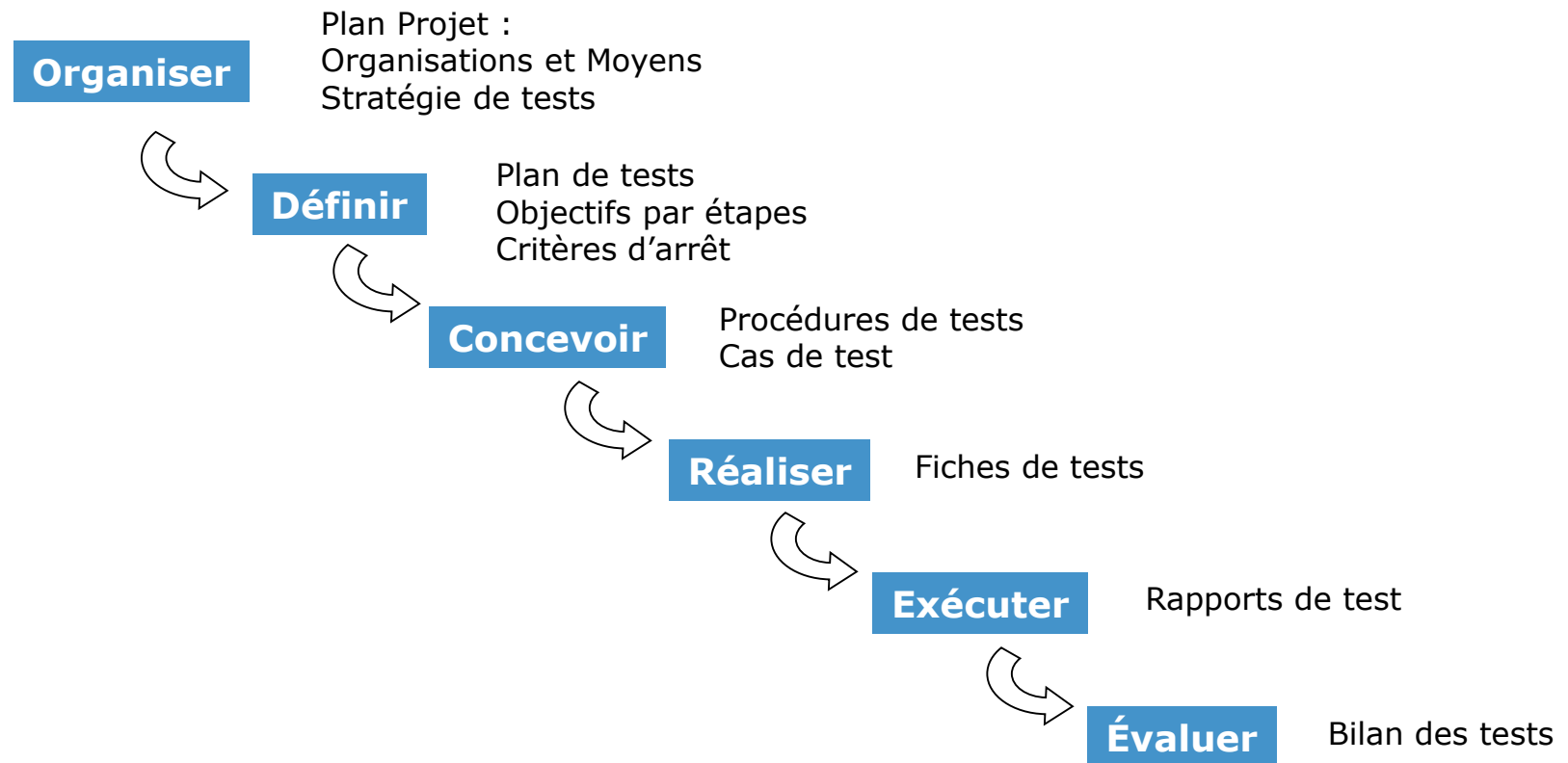


Le processus de tests





Les étapes du processus de tests





Quand tester ?

Expression du besoin

Analyse

Développement

Tests et intégration

Livraison

Formation

Validation

Réception / maintenance

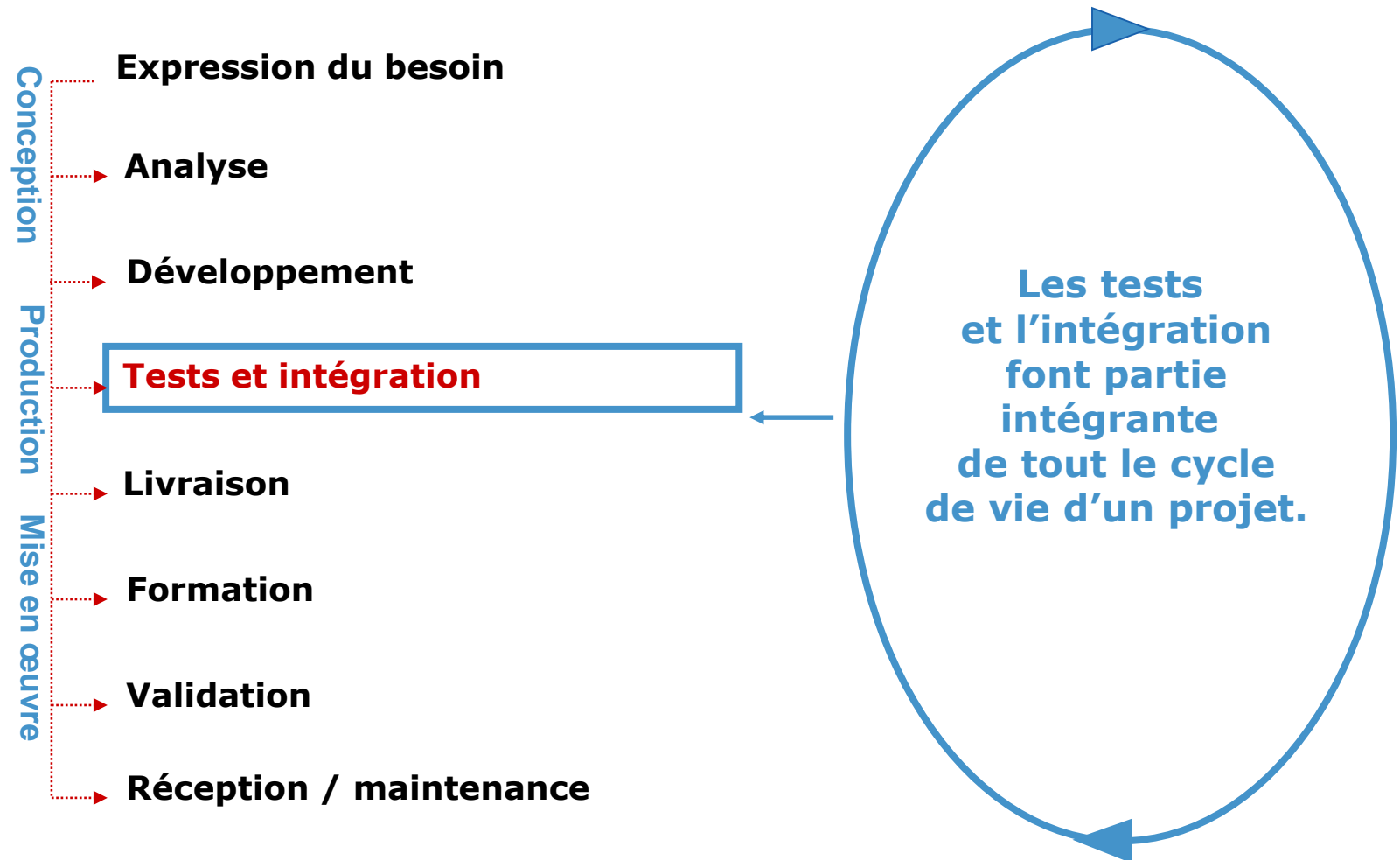


N'envisager les tests
et l'intégration qu'à
ce niveau du projet est
le plus sûr moyen de
le planter !





Quand tester ?

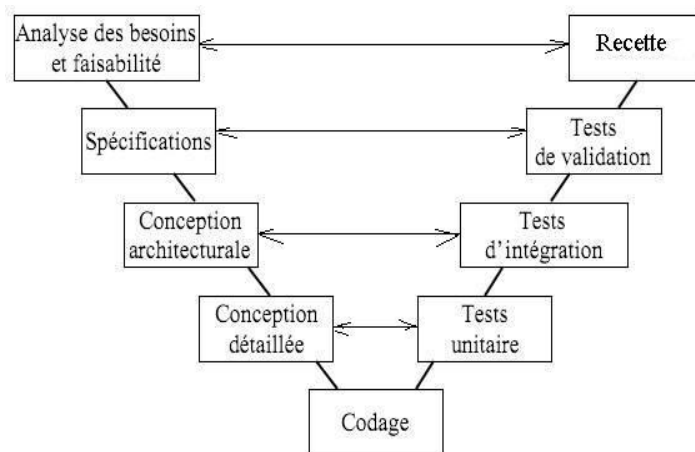




Les tests dans la gestion de projet

■ Les méthodes prédictives

Cycle de développement en VV

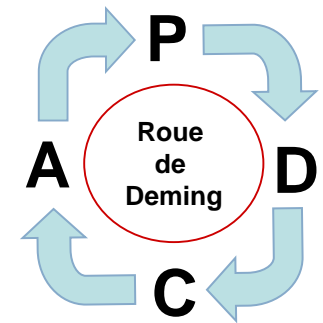


Avantage: Tests préparés en amont
Validation de chaque étape

Inconvénients: Effet tunnel

■ Les méthodes itératives/Agiles

Amélioration continue
par itération



Plan : Préparer, Planifier (ce que l'on va faire)
Do : Dérouler, faire, mettre en oeuvre
Check : Contrôler, vérifier
Act : Acter, standardiser

Test incrémental

Beaucoup de non régression à tester



Les tests dans la gestion de projet

Exigence = expression du besoin



Une idée



Un client



Anomalie



SPADE / SPADE-395
Record & Replay de la création d'un graphique

Modifier Commentaire Attribuer Suite Rédaction Gérer Exporter

Informations

Type: Exigence
Priorité: Important
Affecte la/les version(s): Aucune
Composants: Aucune
Étiquettes: LTS_2017_Santé Prioritaire_EHPAD
Lien d'épopée: Outillage de la modélisation
Sprint: LTS 2017
Intervention: 229 052
Origine: PO MGDIS
Diffusion: NON

Etat: VALIDATION (Afficher le flux de travaux)
Résolution: Fini
Version(s) corrigée(s): 2.14

Personnes

Responsable: Amiot David
Me l'affecter
Rapporteur: Bolchot Gildas
Votes: Voter pour cette demande
Gérer les observateurs: Commencer l'observation de cette demande

SPADE / SPADE-493
TEST - SPADE-395 - Record & Replay de la création d'un graphique

Modifier Commentaire Refusée Retour Rédaction Gérer Cloner Plus d'actions Exécuter

Informations

Type: Test
Priorité: Important
Affecte la/les version(s): Aucune
Composants: Aucune
Étiquettes: LTS_2017_Santé
Environnement: http://DEVSPADE1/TEST_MODELISATION_SANTE_CLK/TEST_MODELISATION_SANTE.application Login: CONCEPT et m...
Sprint: LTS 2017
Intervention: 229 052
Niveau d'importance: Important
Fréquence d'usage: 4. Mensuelle
Origine: Chargé de tests MGDIS
Diffusion: NON

Etat: VALIDATION (Afficher le flux de travaux)
Résolution: Fini
Version(s) corrigée(s): 2.14

Description

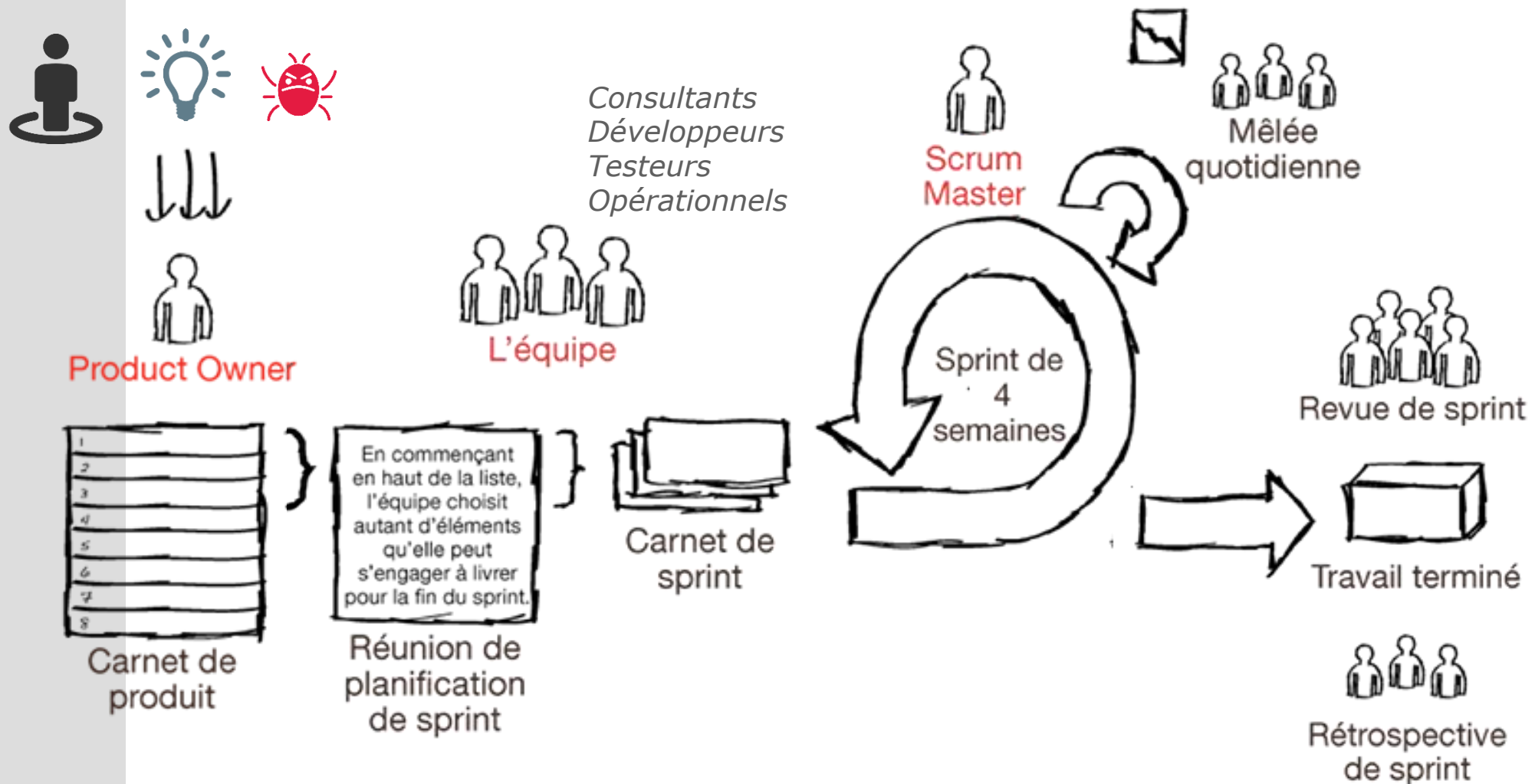
Domaine "Administration", "Paramétrage modélisateur", "Gestion des parcours", Recherche et sélectionner "Stratégie financière"
Cliquer sur "graphique", Créer un graphique

Détails du test

	Étape de test	Données de test	Résultats attendus
1	Lancer le record		Enregistrement actif
2	Créer un graphique	Libellé: test Mettre une variable	Création ok
3	Arrêter le record		un .zip est créé sous " (CheminDuPoste) \\SOF\\record\\record\\output"

Cas de test = Validation de réalisation

➔ Les tests dans la gestion de projet





Les tests dans la gestion de projet

■ Focus méthode Scrum

Une description de la fonctionnalité

Je veux que l'appli me propose les noms correspondants à ma saisie, au fur et à mesure que je tape

Une priorité métier

Must / Should / Could / Wish

1, 2, 3, 4, 5, 6 ...

Une complexité technique

Trivial / Facile / Normal / Difficile / Galère !

1, 2, 3, 5, 8, 13, 20, 40, 100



Les cas de gestion

Chaque lettre saisie doit être convertie en majuscule à la volée

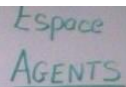
La recherche commence si on a au moins 2 caractères

Les tests fonctionnels

Je tape le libellé en entier, une seule réponse apparaît

Je saisis partiellement le mot, une liste de choix apparaît

Je complète partiellement ma saisie, la liste de choix diminue et ne contient que ma demande...





La recette pour bien tester...



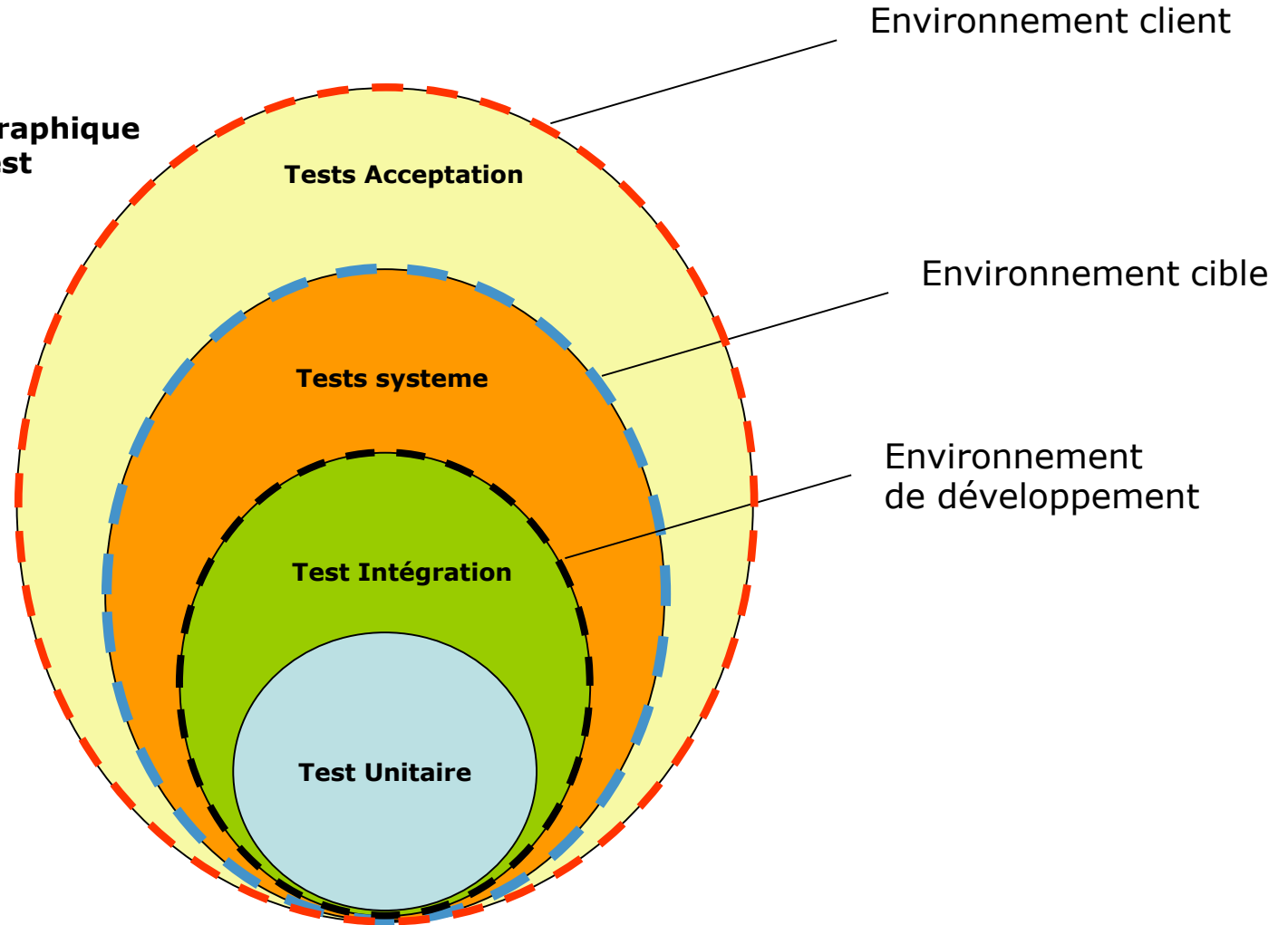
... En 7 étapes

1. Une stratégie de tests



Les niveaux de test

■ Visualisation graphique des niveaux de test

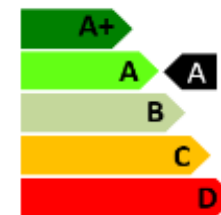




Les thèmes de test

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)



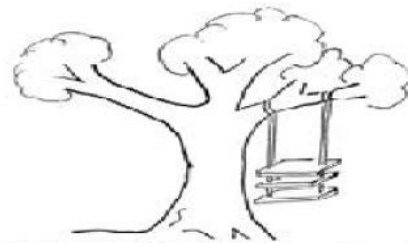
2. Un Référentiel d'exigences



Exigences

■ Les exigences de test

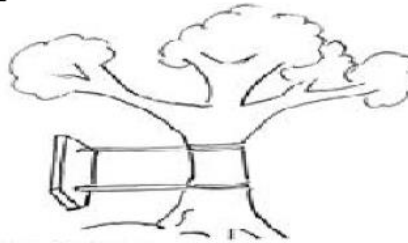
Définir le besoin et comprendre ce qu'attend le client: c'est la base du test.



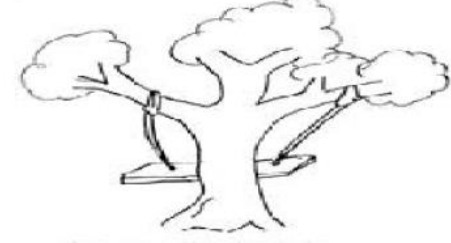
ce que demande l'utilisateur



ce qui est écrit dans le cahier des charges



ce que l'analyste a compris



ce que le programmeur a réalisé

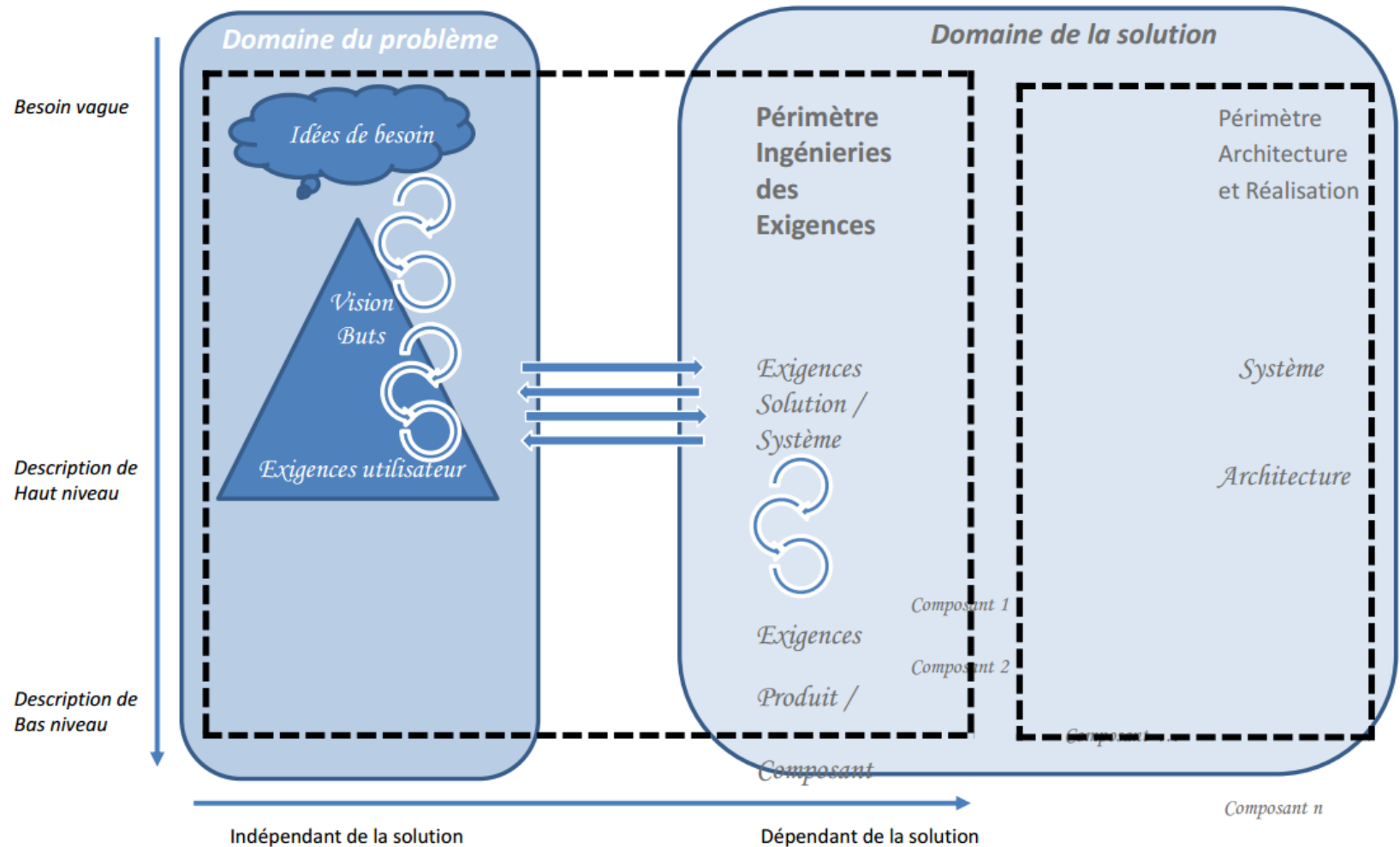


après la mise au point



Ce qu'il fallait ...

■ 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.

■ 2 moyens assez simple pour décrire une fonctionnalité

Basé sur Scrum

En tant que Je souhaite..... Afin de

En tant qu'utilisateur de l'application, je souhaite pouvoir me connecter avec mon compte facebook afin de pouvoir m'authentifier et accéder à l'application en étant authentifié

Basé sur BDD (Behavior driven test) – gherkin

Given..... When... Then (and).

Je suis sur l'écran de connexion, lorsque je clic sur l'icône Facebook alors je dois pouvoir m'authentifier avec mon compte facebook et me connecter à l'application et être reconnu en tant qu'utilisateur.

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

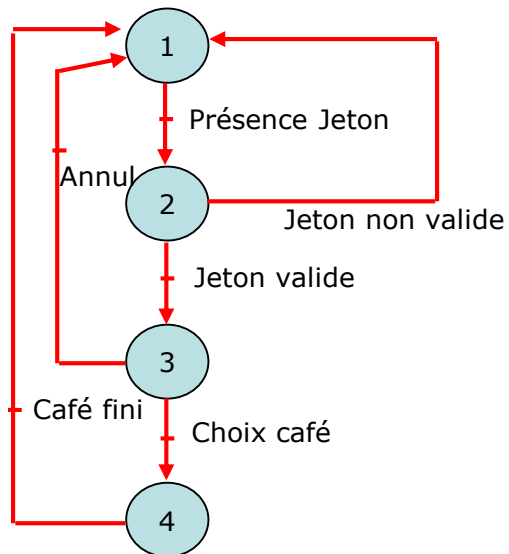
Démarche:

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

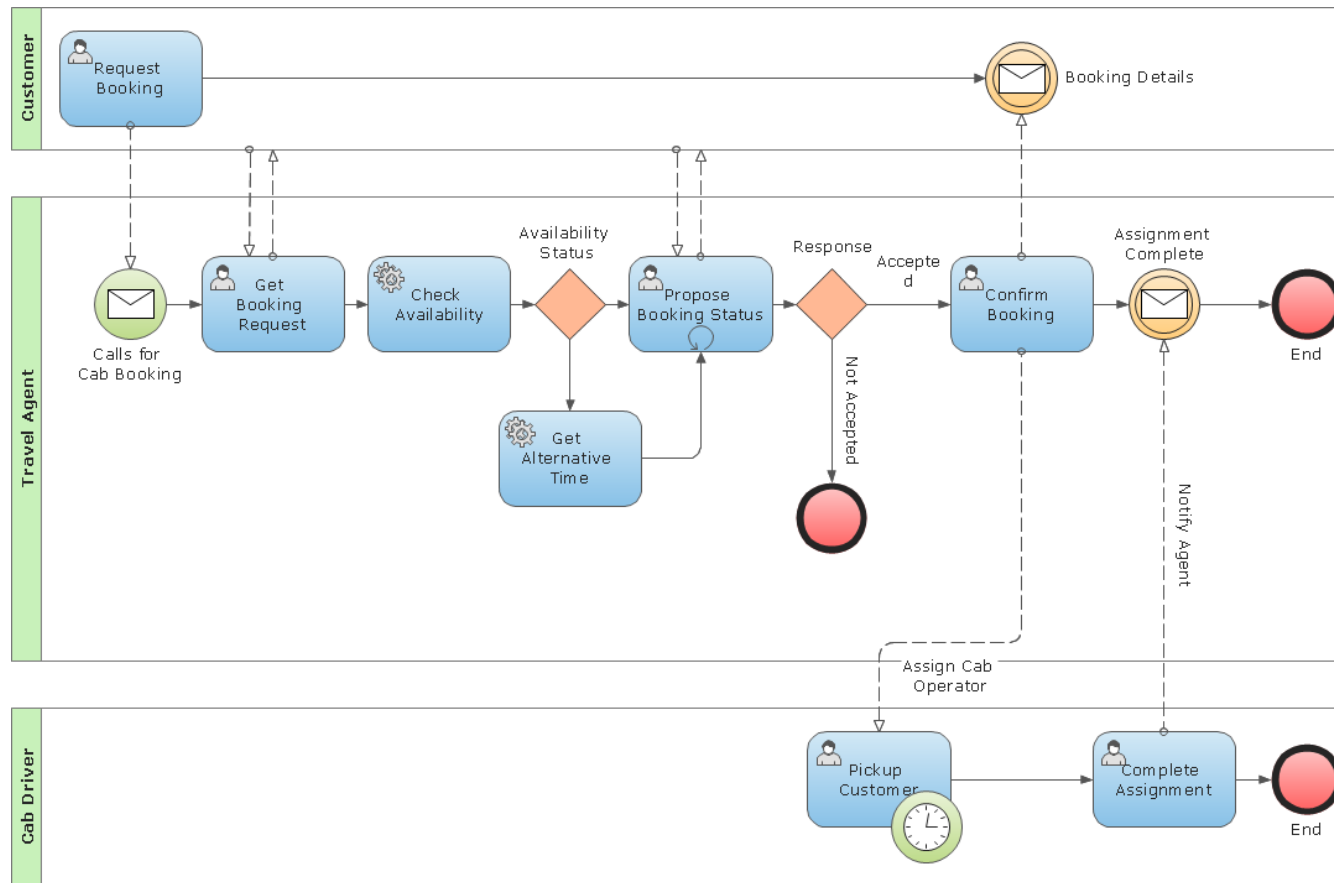
Graphe**Nb arc: 6****Nb Etat: 4****Matrice de transition**

	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

■ BPMN



■ Atelier Mockup

Technique permettant de maquetter les interfaces utilisateurs.

Le client peut tout de suite visualiser un rendu sans même faire une ligne de développement

Cela permet d'avoir un retour rapide sur la compréhension (spécification par le visuel)

Cela permet de vérifier l'adéquation de l'interface avec les cas d'usage



Test

Slider control

Email Address

Email Address

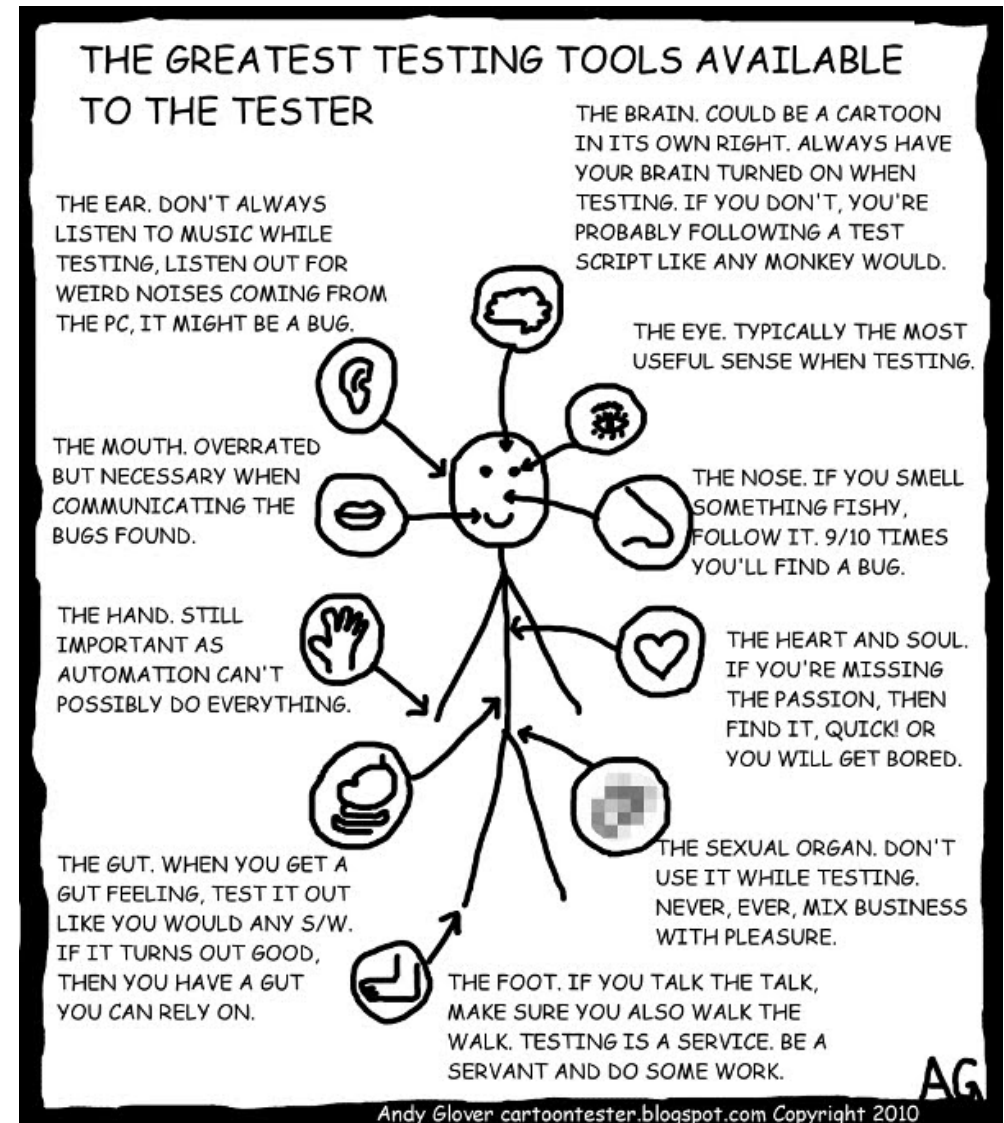
4/22/2012 [Calendar icon]

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

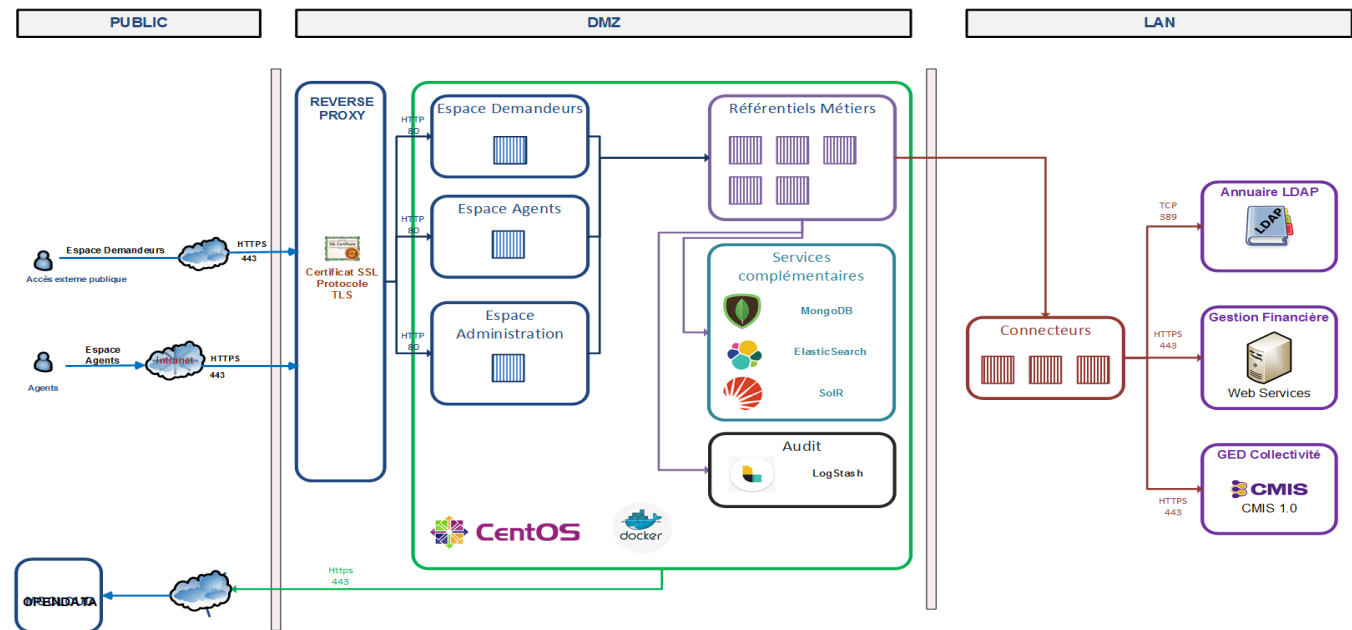
■ Environnement de test

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)



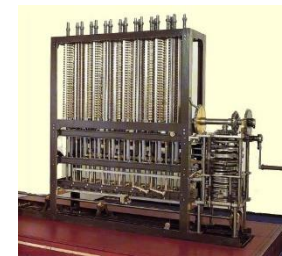
Intégration entre les outils
(Suite logicielle, interopérabilité, ALM)



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

Exécution
des tests

Générateur
d'environnement

Gestion des
défauts

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...

squash



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

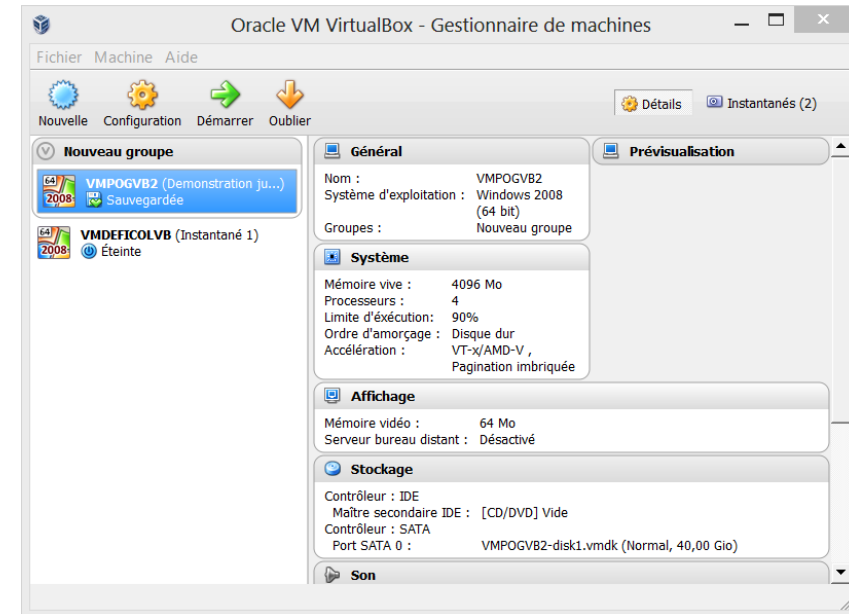
Récupérer les métriques système

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. The query bar contains the following 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`. The table below displays the first 10 issues matching the query.

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due	Votes	Backlog Order
	JRA-868	Resolve & Time spent	Unassigned	Primož Piršan		Resolved	Fixed	01/Oct/02	13/Aug/10		281	
	JRA-14076	DoubleConverter / NumberCFTYPE 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	
	JRA-21439	Support Transparent Image for Project Avatar	Chris Mountford [Atlassian]	Jack Low [Atlassian]		Resolved	Fixed	27/May/10	11/Aug/10		11	Disabled
	JRA-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
	JRA-20351	AXIS Internal SOAP type representation is unstable.	Brenden Bain [Atlassian]	Brad Baker [Atlassian]		Resolved	Fixed	04/Feb/10	27/Sep/10		8	Disabled
	JRA-20956	Issue View shows fields of second tab when first opened	Unassigned	Dieter Paul		Resolved	Fixed	09/Apr/10	22/Jul/10		7	Disabled
	JRA-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
	JRA-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
	JRA-20995	Privilege escalation vulnerability when administrator access is compromised	Unassigned	Edwin Wong [Atlassian]		Resolved	Fixed	13/Apr/10	20/Apr/10		6	Disabled
	JRA-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
	JRA-17759	CAPTCHA image broken when running in OpenJDK	Unassigned	Jeff Turner [Atlassian]		Resolved	Fixed	23/Jun/09	21/Sep/10		6	Disabled
	JRA-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
	JRA-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: 10/11/2008	Testeur: Mr xxx
Nature de l'anomalie: Documentation		Objet concerné: Manuel utilisateur
Description de l'anomalie: Message d'erreur 4 non-conforme à celui du produit		
Reproductibilité: Oui		
Action demandée: Correction du manuel		Réponse obtenue: Correction effectuée le 12/11/2008 par modification de la copie d'écran du message d'erreur
Résolue : Oui <input checked="" type="checkbox"/> Non <input type="checkbox"/>		Visa du testeur: OK



■ 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

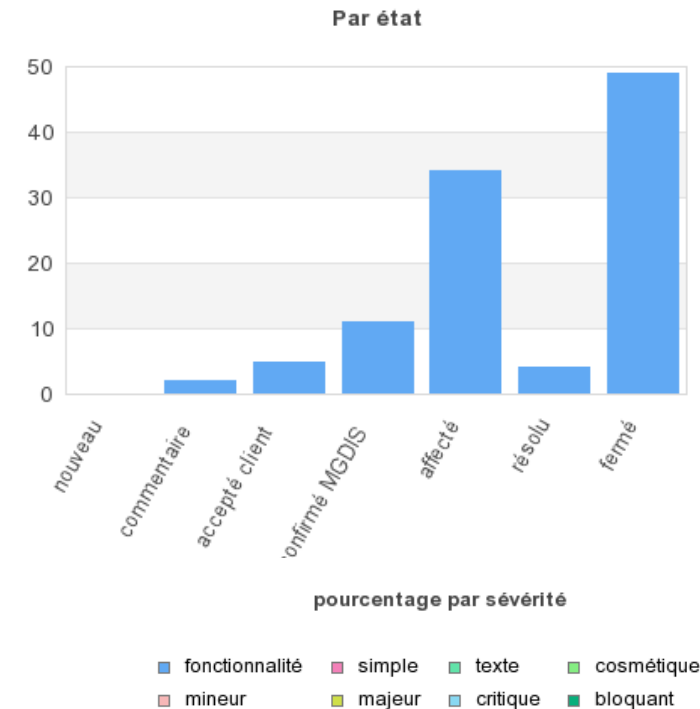
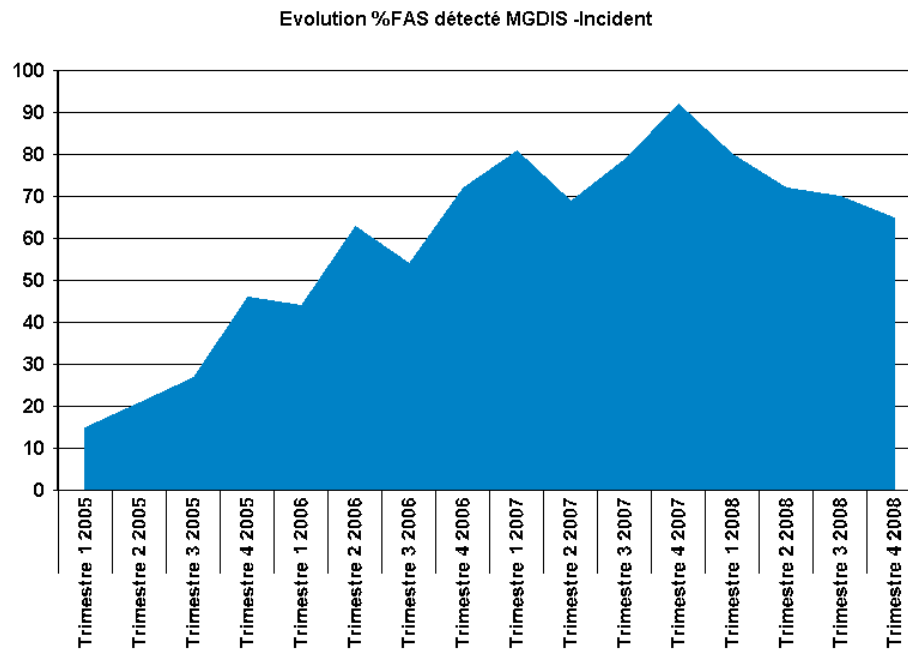
[Accueil](#) | [Mon affichage](#) | [Afficher les bogues](#) | [Rapporter un bogue](#) | [Historique des changements](#) | [Calendrier](#) | [Synthèse](#) | [Administration](#) | [Mon compte](#) | [Fermer la session](#)

Visité récemment: 0000098, 0000097

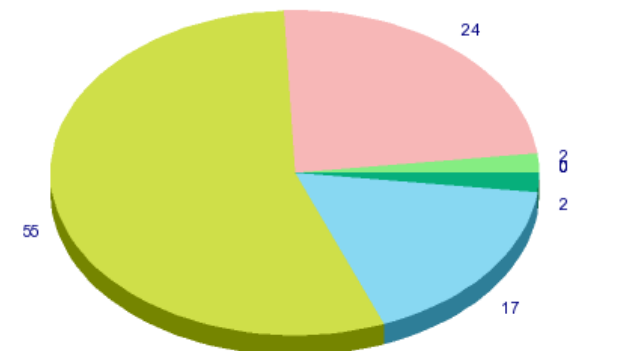
Visualiser les détails du résultat [[Aller aux notes](#)] [[Envoyer un rappel](#)] [<<] [>>] [[Historique du bogue](#)] [[Imprimer](#)]

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
Rapporteur	jamette-h				
Assigné à					
Priorité	normale	Impact	majeur	Reproductibilité	Aléatoire
État	accepté client	Résolution	ouvert		
Résumé	0000057: Perte visuelle des actions courriers dans le suivi de dossier (pb pack)				
Description	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				
Étapes pour reproduire	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				
Balises	Aucune balise n'est associée.				
Associer balises	(Séparer par ',') <input type="text"/> <input type="button" value="Balises existantes"/> <input type="button" value="Associer"/>				
CS installé	aucun				
Domaine	Subventions				
N° Intervention/FAS MGDIS	FAS 2013-1704				
Pack installé	8.20.11				
Résolution sur composant spécifique					
Résolution sur pack					
Pièces jointes					

■ 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

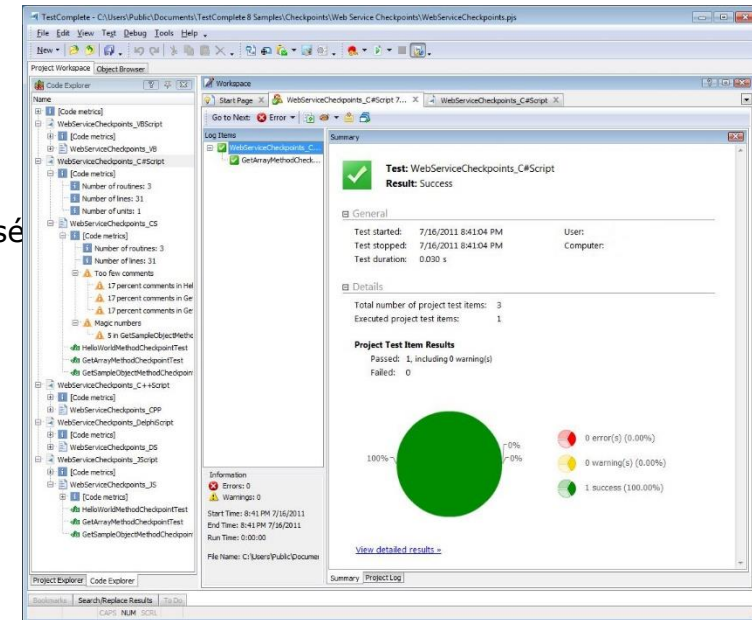
Exécution des tests

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

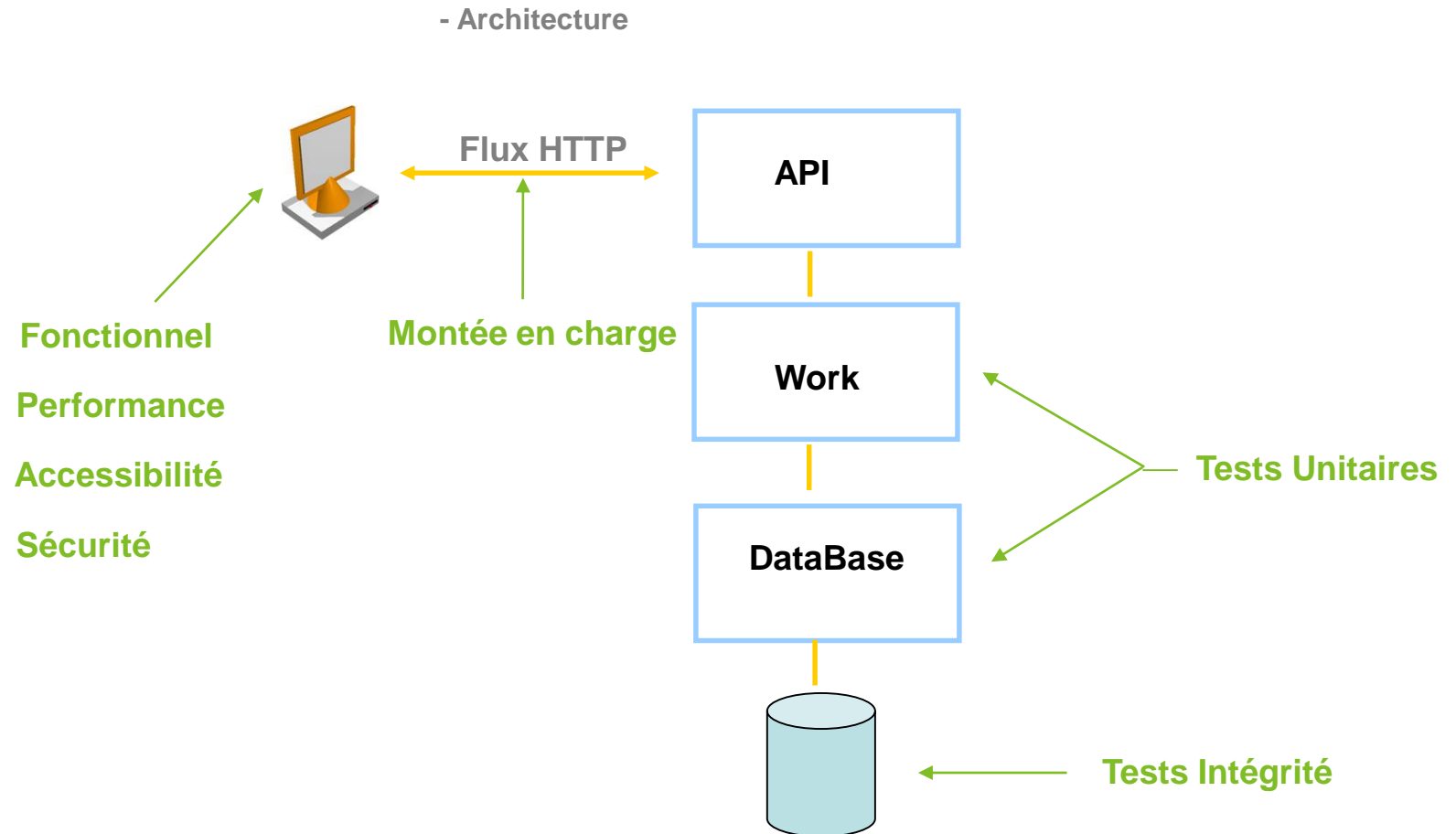
■ Les bonnes pratiques

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.

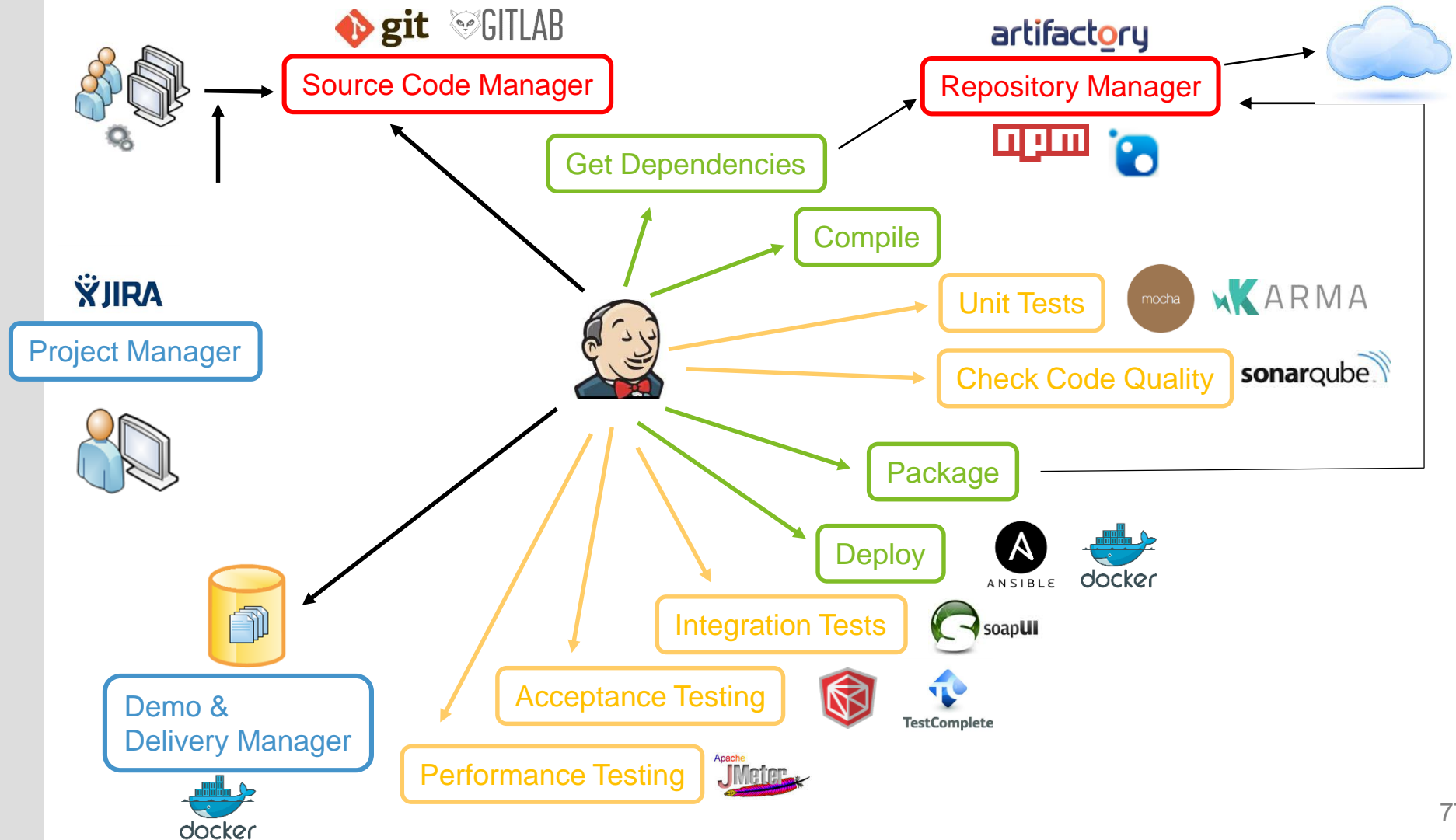
Focus sur l'automatisation les tests

■ A chaque tiers son test



7. Une usine pour produire

➔ 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



Performances: Test nominal des principales requêtes de création de tiers et aides

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_polo_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

