

Méthodologie de production logicielle

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



Anthony LABARRE



Anthony LABARRE

Responsable Direction Technique et QA



MGDIS

25 ans d'expérience à MGDIS, dont 18 en tests et qualité logicielle

50 techniciens et Ingénieurs

- Test agile et qualité du logiciel
- Homologation et Test automatisé
- Devops / SRE
- Support Editeur
- Système d'information
- Sécurité des systèmes d'information

Certifié ISTQB Niveau Fondation

Certifié ISTQB Agile Tester

Certifié ISTQB Niveau Gestionnaire Avancé

Certifié ITIL

Certifié REQB

Membre MforTest

Chargé d'enseignement sur Qualité logicielle



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

labarre-a@mgdis.fr



Que va-t-on voir dans ce cours?

- **Sensibilisation** sur la **qualité** de production des logiciels
- Apport d'un **premier niveau de connaissances** et de culture du test pour
 - aujourd'hui : construire vos projets
 - Demain : travailler en équipe
- Faire découvrir le **métier de Testeur**
- Apporter des **méthodes, des techniques, des outils**



Vos questions sont les bienvenues tout le long de ce cours.



Les Bugs, une histoire sans fin...

9/9

0800 1000 1300 (032) MP - MC 033 PRO 2 2.130476415 2.130676415 { 1.2700 9.037 847 025 9.037 846 995 const 1.2700 9.037 847 025 9.037 846 995 const

Relays 6-2 in 033 failed special sped test in Relay 2145
Relay 2145
Relay 3370

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.
1600 instant startd.
1700 closed down.

IPHONE

Bug des lignes qui apparaissent sur l'écran des iPhone 14 Pro : on avance

Un correctif est en passe d'être déployé et vous pourrez bientôt le recevoir sur votre mobile.

Publié le 14 janvier 2023 à 18:00
Par iPhon.fr



Arianespace a détruit sa fusée Vega-C en raison d'un bug, une catastrophe pour l'Europe



NASA/Bill Ingalls/Cover Images/Sipa



Les Bugs, une histoire sans fin...

Impôts. Déclaration de revenus: d'un million de contribuables, un bug

Les déclarations de revenus de nombreux contribuables n'ont toujours pas été traitées. La Direction générale des impôts s'explique sur ce retard.

Facebook corrige un bug responsable de plantages sur son application 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

Covid-19: un bug informatique perturbe la délivrance de "QR codes"

Source AFP





Impact

- Image de l'entreprise
- Economique
 - Pertes financières directes (cout de gestion du bug et de son rétablissement, mobilisation des équipes...)
 - Pertes financières indirecte (retard de livraison, cout de communication, réorientation du projet, désorganisation...)
- Juridique
- Environnementale
- Santé
- ...

"Le mouton à cinq pattes."



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 ?



Isabelle Gobetier

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 inconsistante
- 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 ça 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 Complexé

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	Le test aurait pris 30 minutes
En phase de maintenance	19 heures	

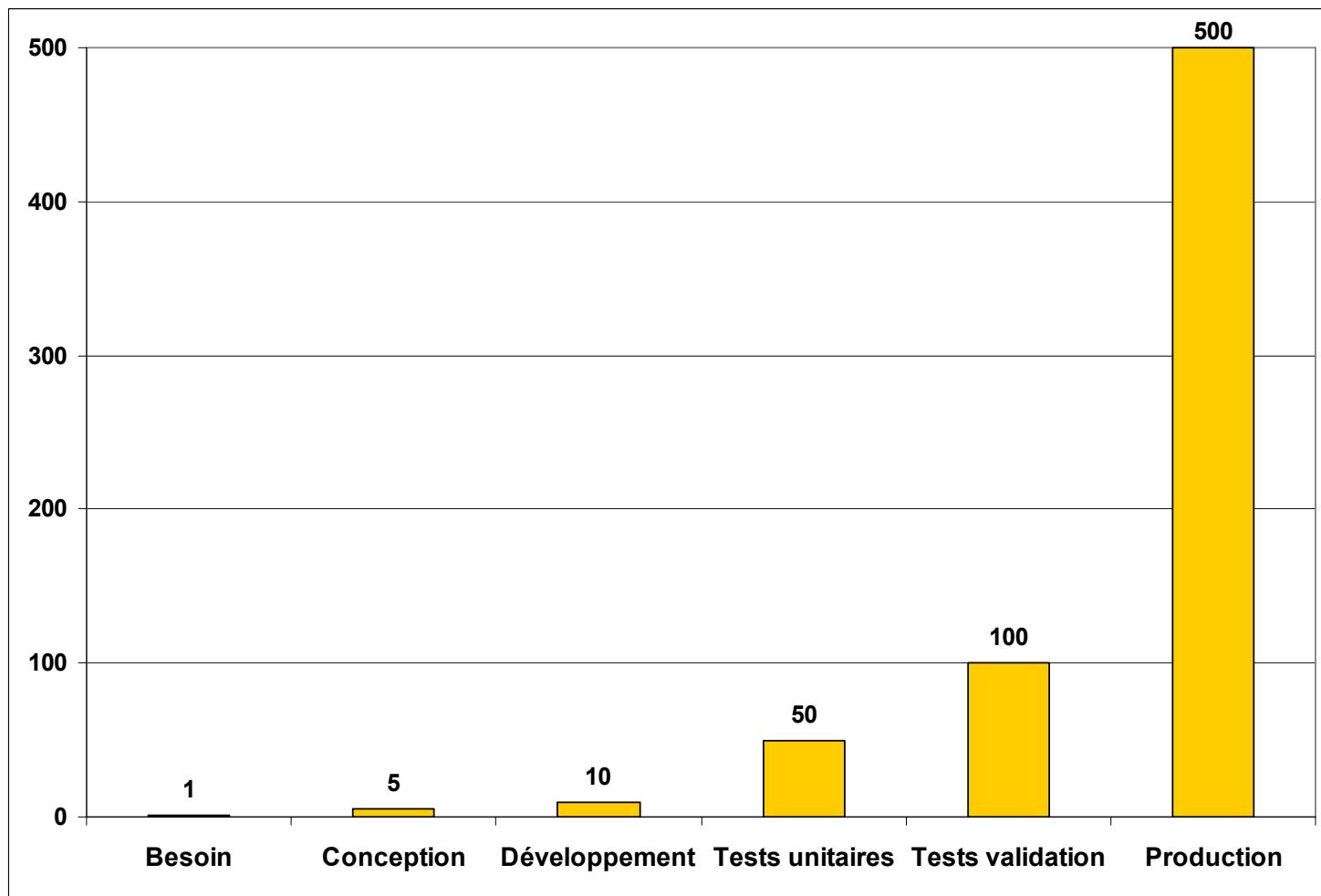
Cette anomalie n'a un impact que sur l'économie de l'entreprise. On ne compte pas ici les coûts non chiffrables comme la non satisfaction du client!!!
Des situations pires peuvent exister provoquant notamment des dégâts matériels et humains



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)

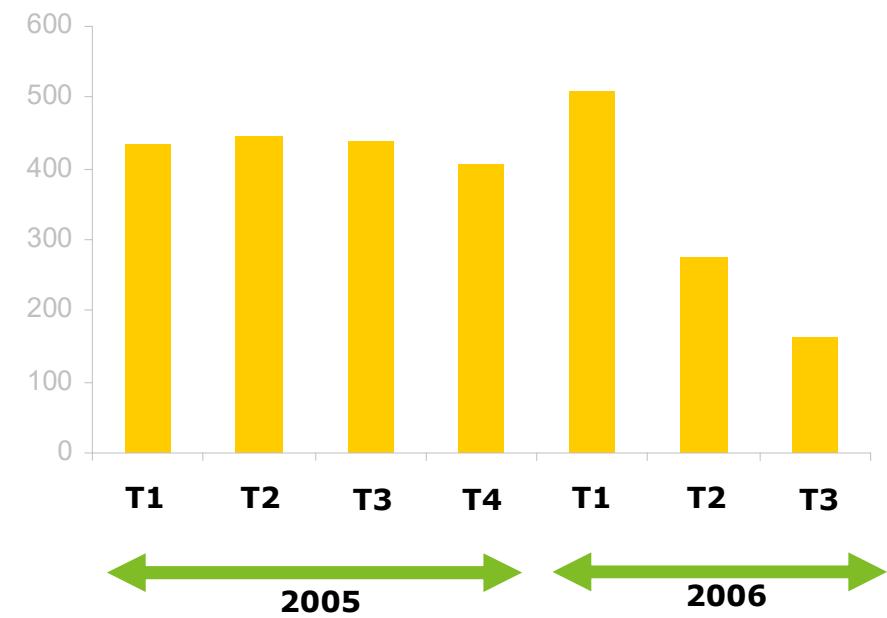
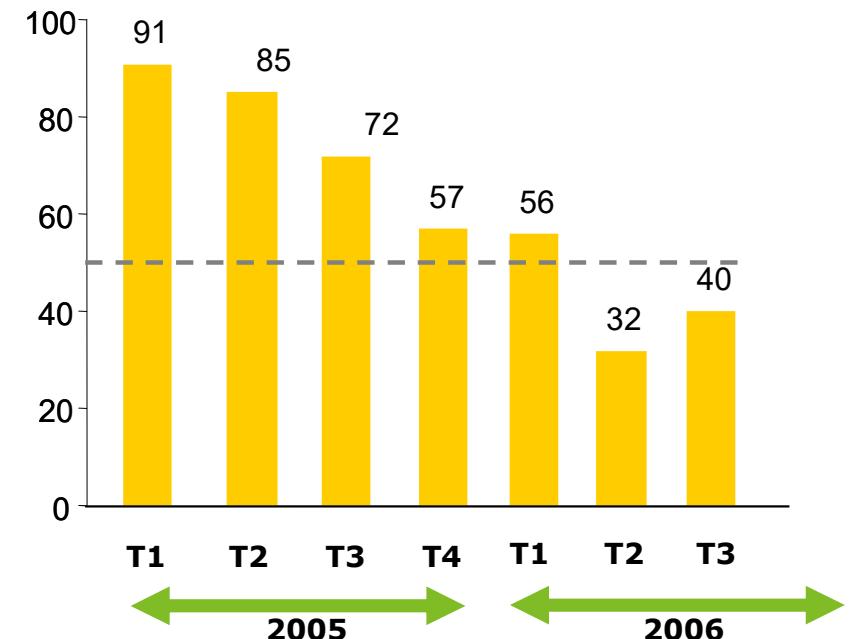
Exemple à MGDIS

% 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

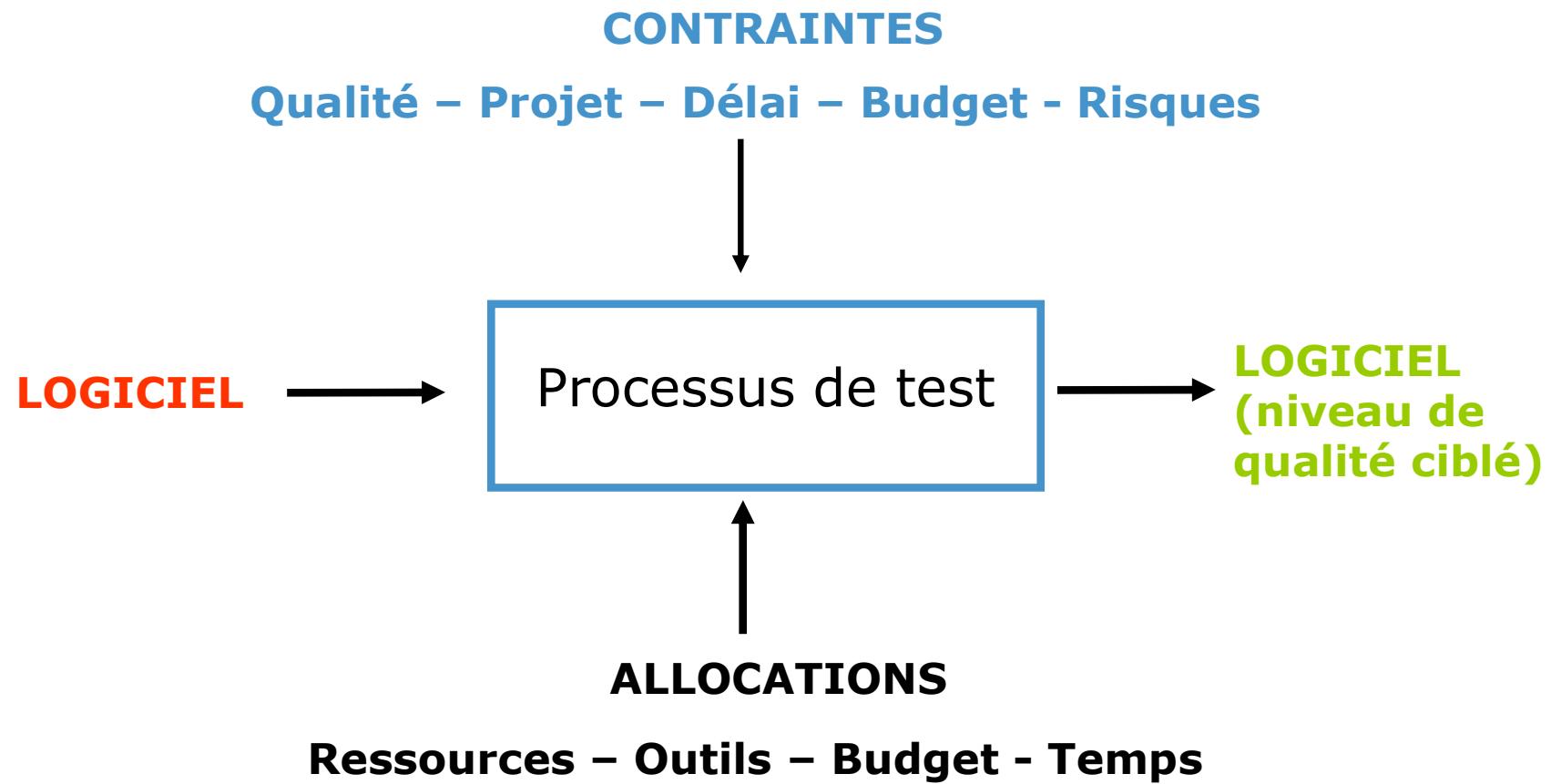
■ Objectif n°4:

RESPECTER LA LOI

- le Code de propriété intellectuelle
- le Code pénal
- la loi n° 98-389 du 19 mai 1998 relative à la responsabilité du fait des produits défectueux
- loi du 6 janvier 1978 dite « Informatique et libertés ».
- 20 juin 2018 sur la protection des données personnelles
- Référentiel général de sécurité (RGS)
- Référentiel général d'interopératblité (RGI)
- Référentiel général d'accessibilité pour les administrations (RGAA)

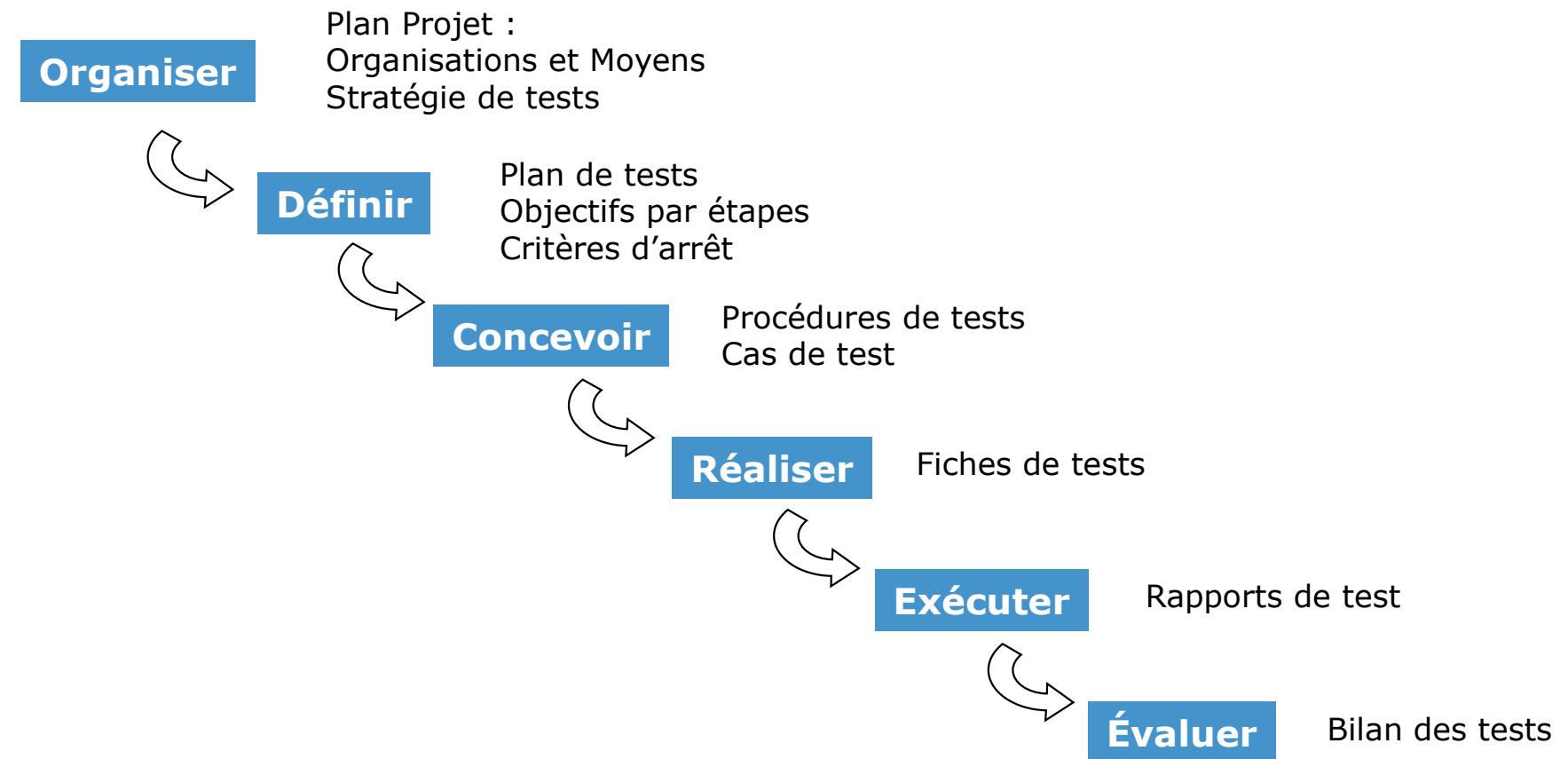


Le processus de tests





Les étapes du processus de tests





Quand tester ?

Expression du besoin

Analyse

Développement

Tests et intégration

Livraison

Formation

Validation

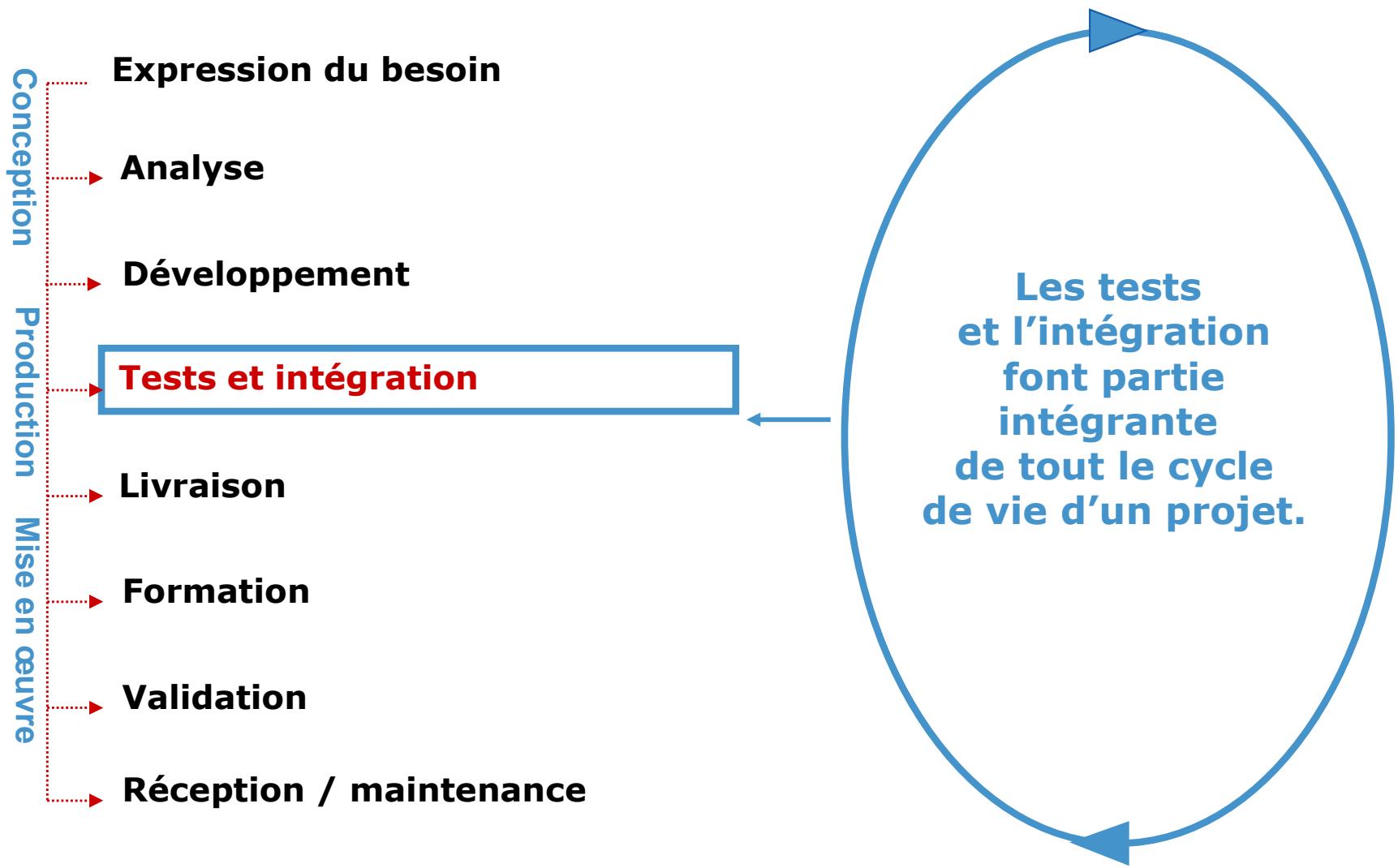
Réception / maintenance



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



Quand tester ?

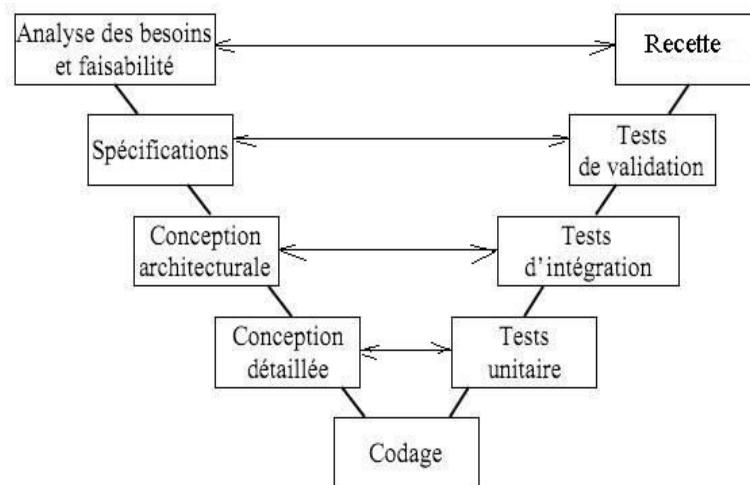




Les tests dans la gestion de projet

■ Les méthodes prédictives

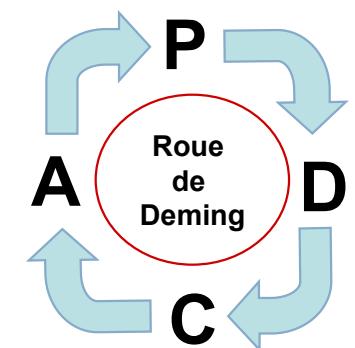
Cycle de développement en VV



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:	<input checked="" type="checkbox"/> Exigence	Etat:	VALIDATION (Afficher le flux de travaux)
Priorité:	<input checked="" type="checkbox"/> Important	Résolution:	Fini
Affecte la/les version(s):	Aucune	Version(s) corrigée(s):	2.14
Composants:	Aucune		
Étiquettes:	LTS_2017_Santé Prioritaire_EHPAD		
Lien d'épopée:	Outilage de la modélisation		
Sprint:	LTS 2017		
Intervention:	229 052		
Origine:	PO MGDIS		
Diffusion:	NON		

Personnes

Responsable:	Amiot David
Me l'affecter	
Rapporteur:	
Votes:	
Gérer les observateurs:	
Boichot Gildas	
Voter pour cette demande	
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:	<input checked="" type="checkbox"/> Test	Etat:	VALIDATION (Afficher le flux de travaux)
Priorité:	<input checked="" type="checkbox"/> Important	Résolution:	Fini
Affecte la/les version(s):	Aucune	Version(s) corrigée(s):	2.14
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		

Description

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

Détails du test

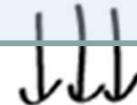
	Étape de test	Données de test	Résultats attendus
1	Lancer le record		Enregistrement actif
2	Créer un graphique	Libellé: test Mettre une variable	Création ok
3	Arrêter le record		un zip est créé sous " (CheminDuPoste) \ISOFlirecord\output"

Cas de test = Validation de réalisation

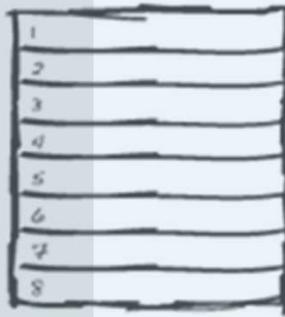


Les tests dans la gestion de projet

Alimente le backlog avec les Bugs et Idée d'amélioration



Product Owner



Carnet de produit

Consultants
Développeurs
Testeurs
Opérationnels



L'équipe

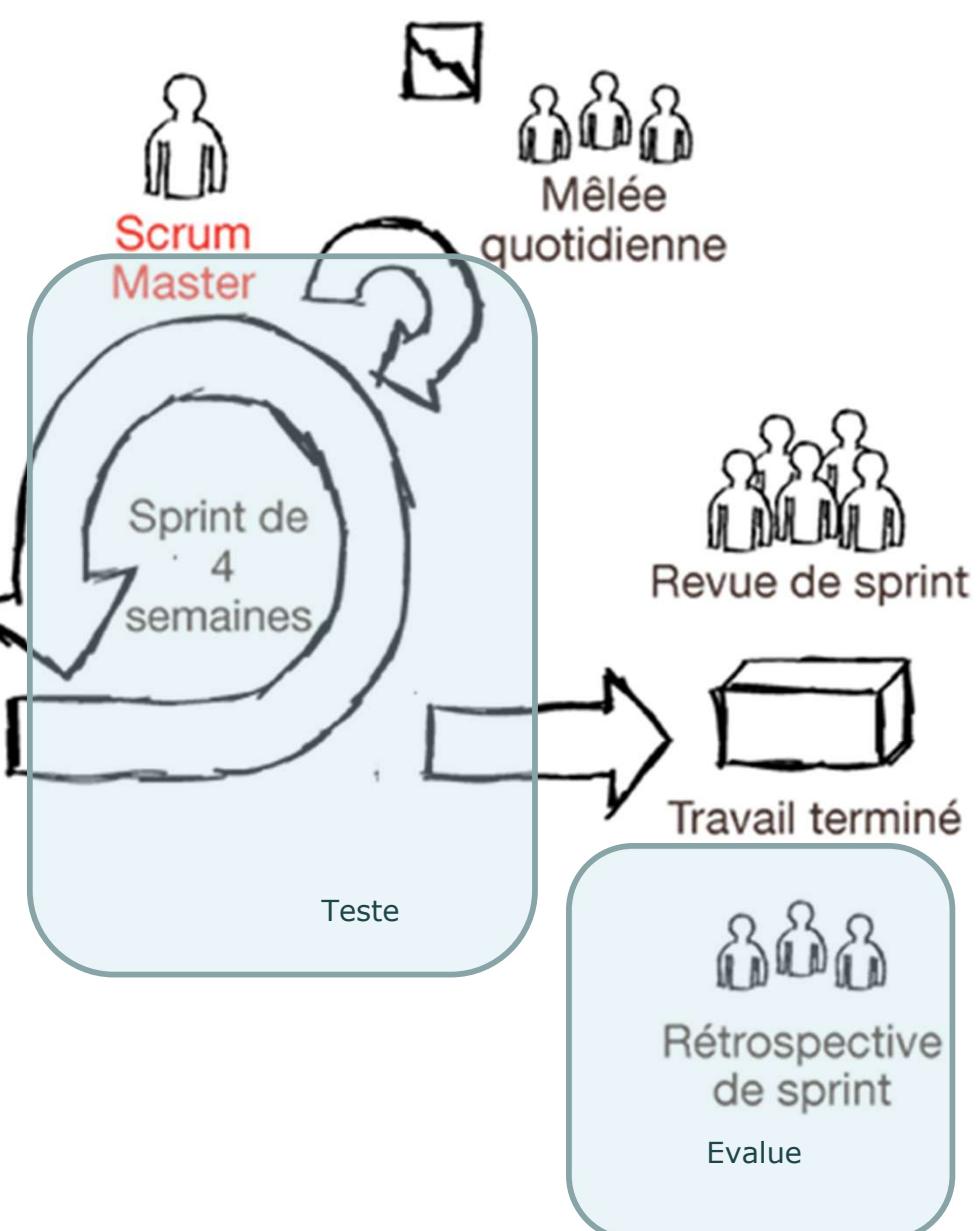
En commençant en haut de la liste, l'équipe choisit autant d'éléments qu'elle peut s'engager à livrer pour la fin du sprint.

Réunion de planification de sprint



Carnet de sprint

Participe au raffinement des exigences, estimations des tests, planification, allotissement des exigences





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 ...



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**

Une complexité technique

Trivial / Facile / Normal / Difficile / Galère !

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

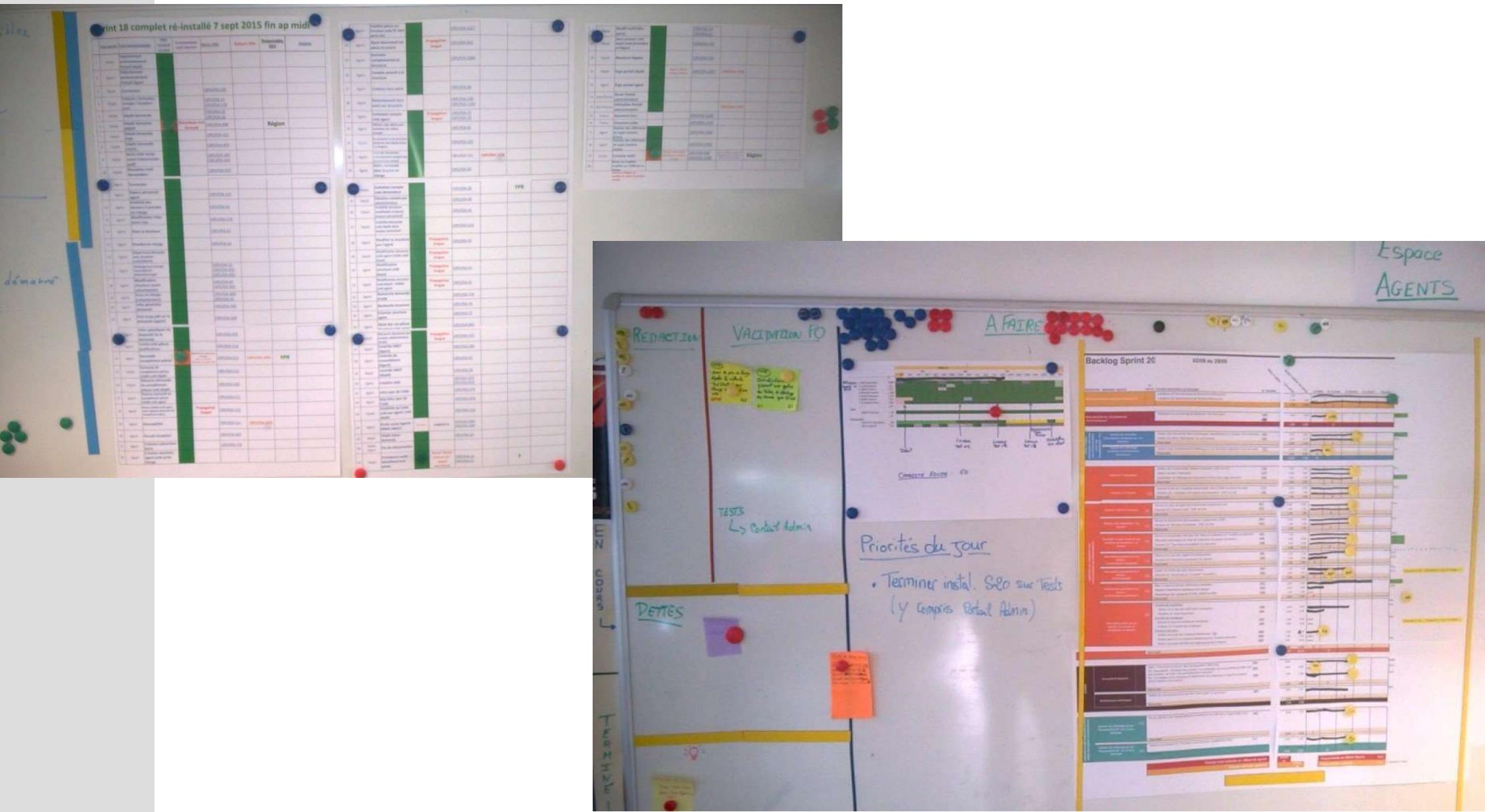
Les tests fonctionnels

Je tape le libellé en entier, une seule réponse apparaît

Je saisie partiellement le mot, une liste de choix apparaît

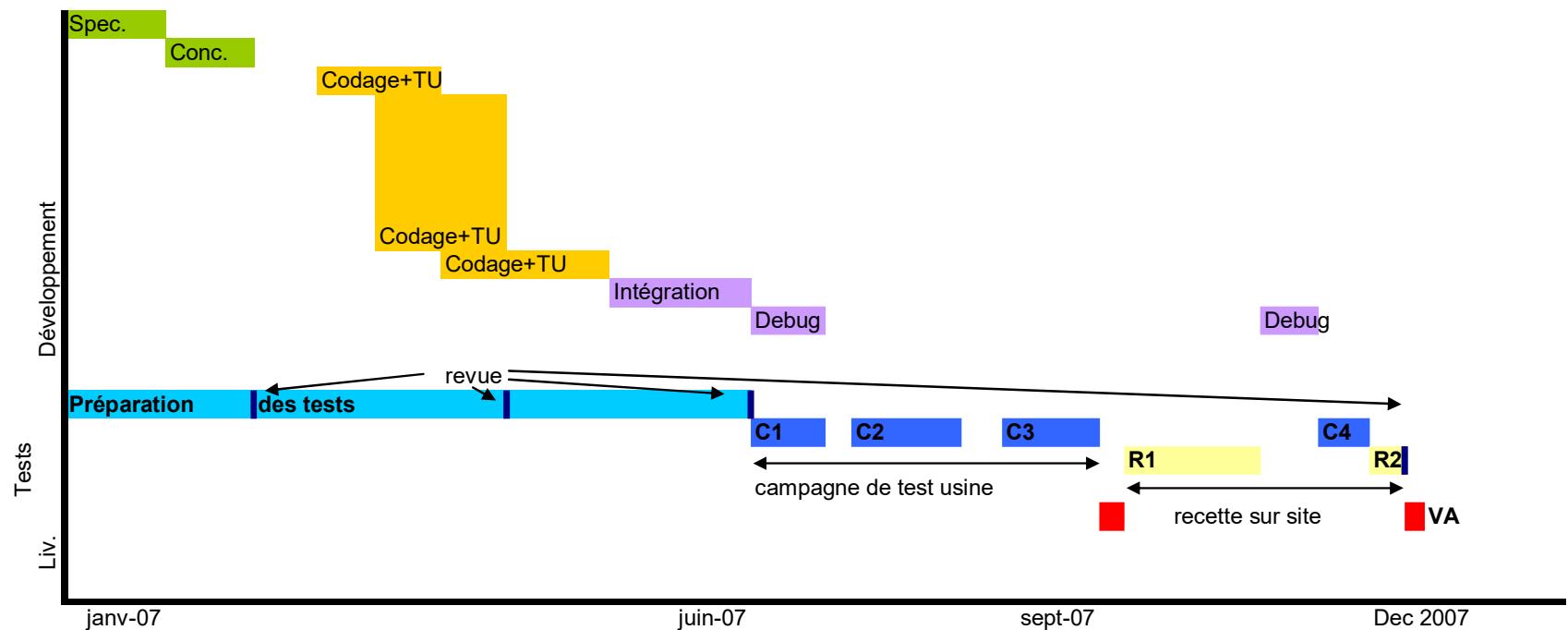
Je complète partiellement ma saisie, la liste de choix diminue et ne contient que ma demande...

>> Cours Tests logiciels

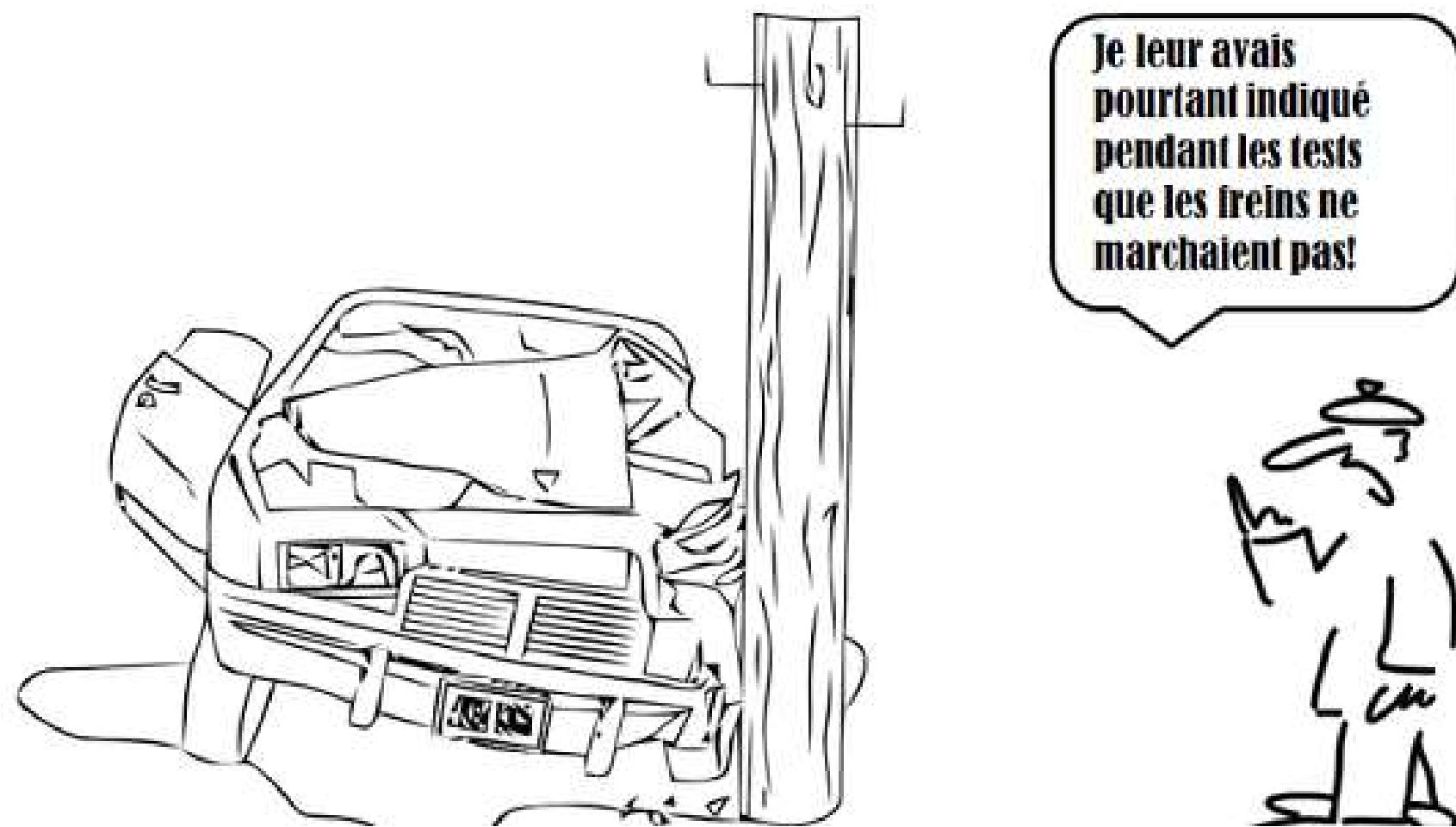


Le plan de test

Définition ISTQB : un document décrivant l'étendue, l'approche, les ressources et le planning des activités de test prévues. Il identifie entre autres les éléments et caractéristiques à tester, qui fera chaque tâche, le degré d'indépendance des testeurs, l'environnement de test, les techniques de conception des tests et les techniques de mesure des tests à utiliser, et tout risque nécessitant des plans de contingence. C'est un document reprenant les processus de planification des tests [d'après IEEE 829]
 Un plan de test définit donc ce que l'on va tester, comment on va le tester mais aussi ce qui ne va pas être testé. Une analyse des risques est également présente afin de décrire les limites de ces choix et leurs impacts sur la qualité.



En attendant ...





Une production de qualité ...



... En 7 étapes

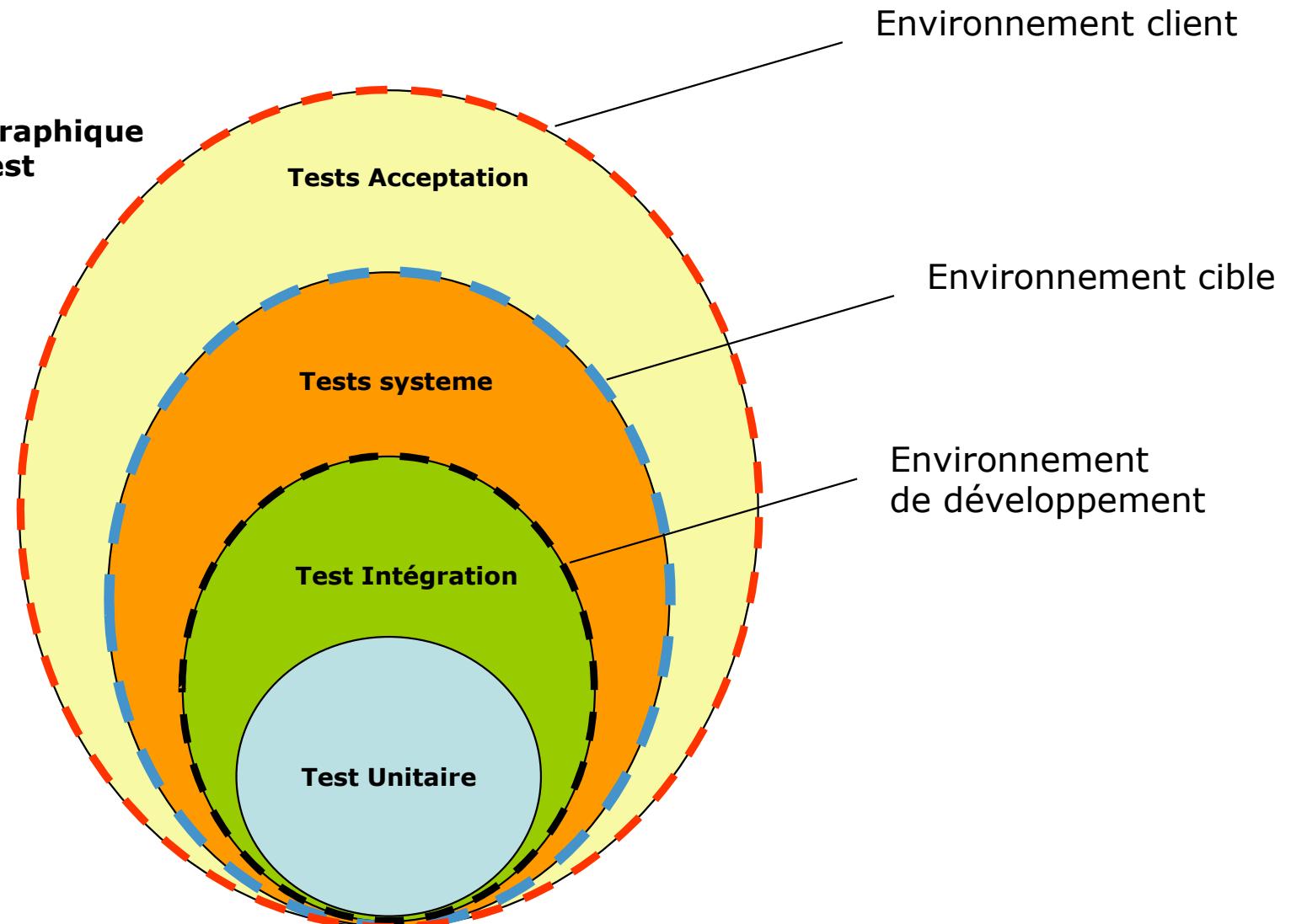
- Une stratégie de test
- Des bons testeurs
- Un référentiel d'exigences
- Des cas de test pertinents
- Un banc de test
- Des bons outils
- Une usine de production

1. Une stratégie de tests



Les niveaux de test

■ Visualisation graphique des niveaux de test

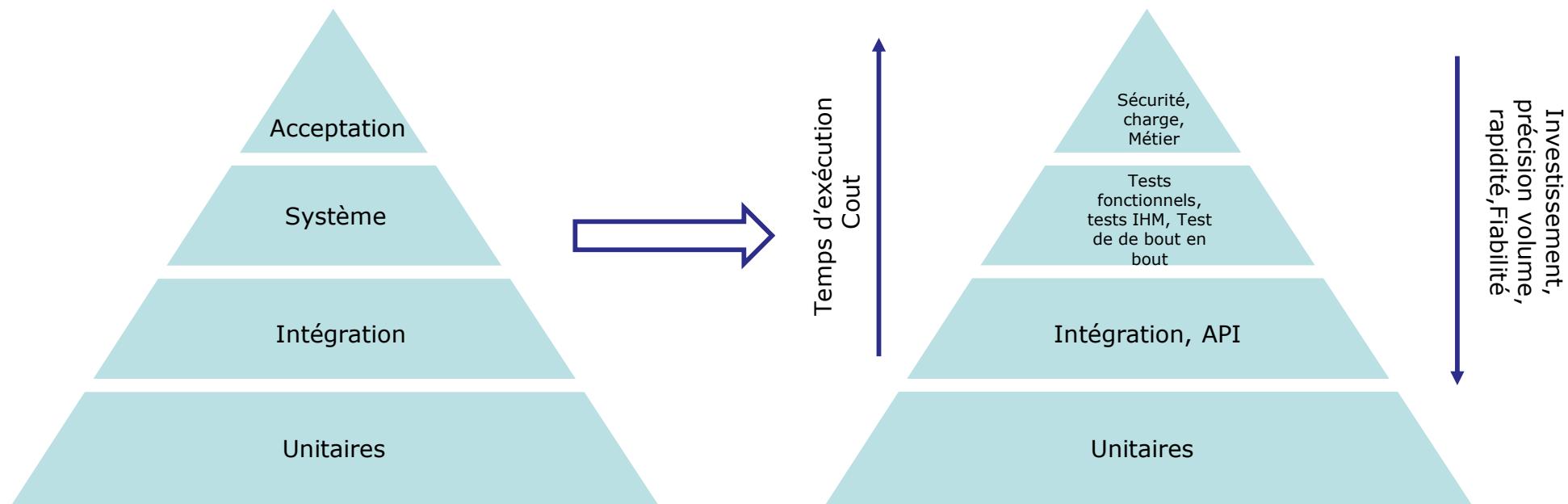




La pyramide des tests

On représente les niveaux en pyramide pour bien montrer où se situe l'effort de test

Excellent moyen d'illustrer la stratégie de test, l'importance de chaque niveau

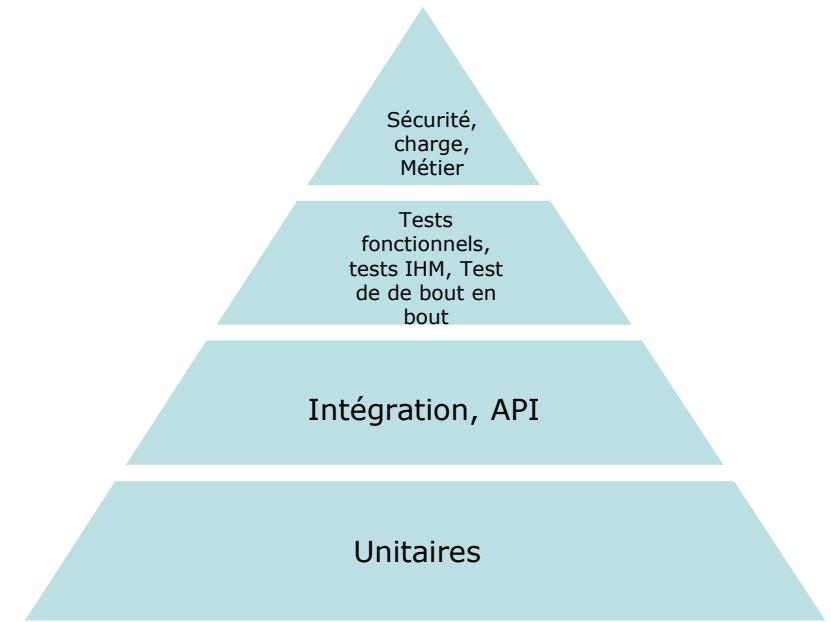
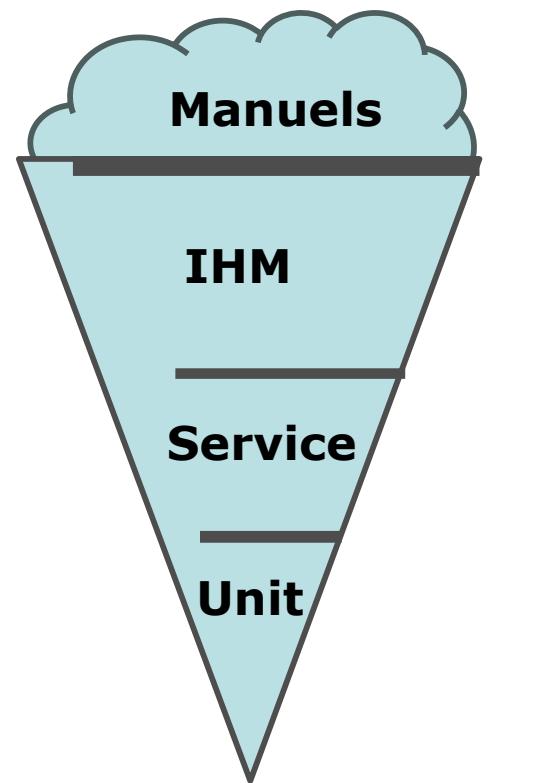




La pyramide des tests

Effet Ice Cream testing

On parle souvent d'inverser la pyramide de test

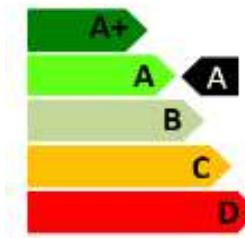




Les thèmes de test

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 simultanéinité
- 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 :

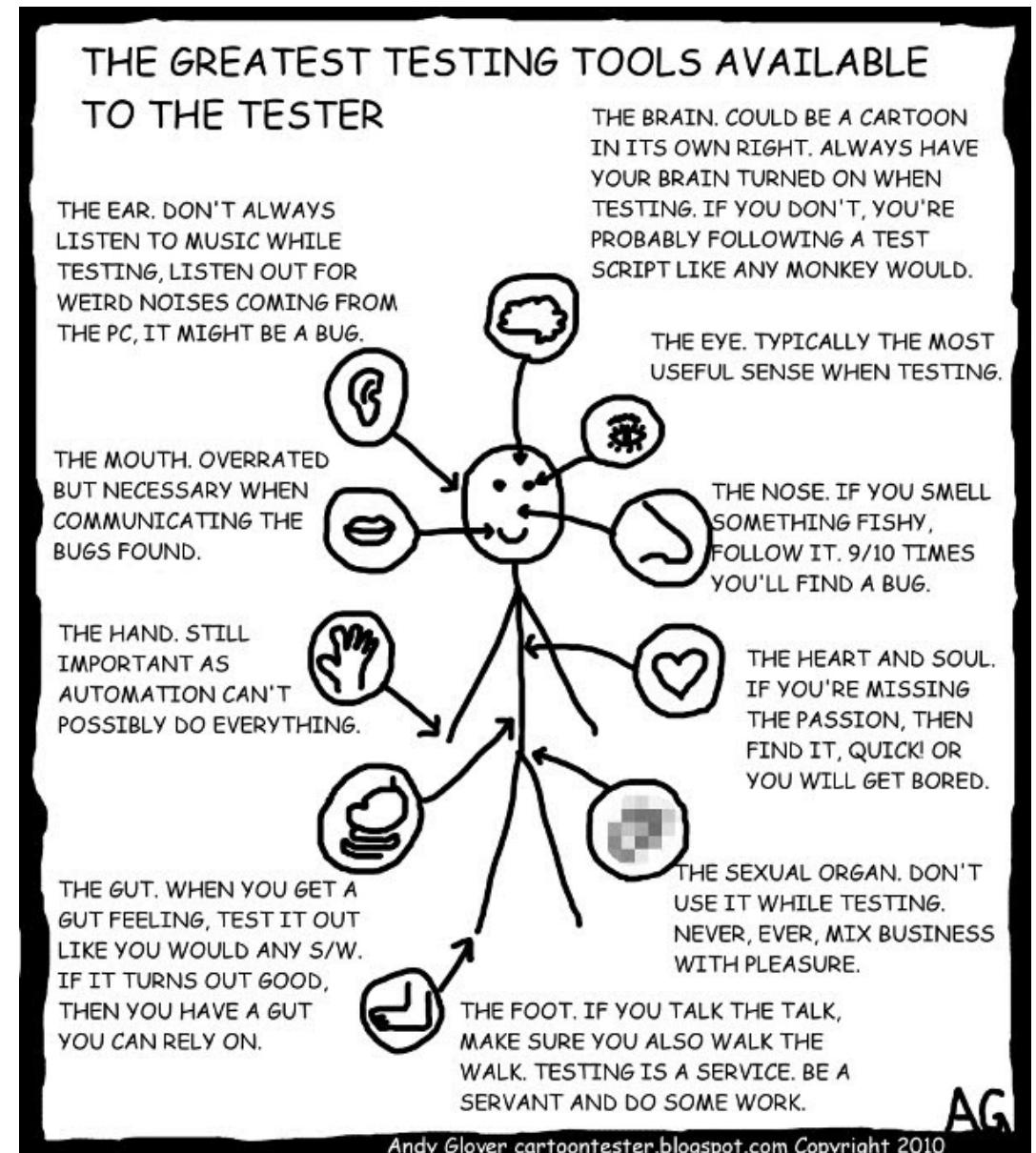
le contexte change, est ce que l'application fonctionnera toujours (exemple: passer de windows à linux)

2. 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



Requirements
Engineering
Qualifications Board



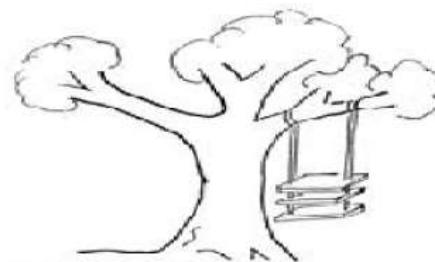
3. Un Référentiel d'exigences



Exigences

■ Les exigences de test

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



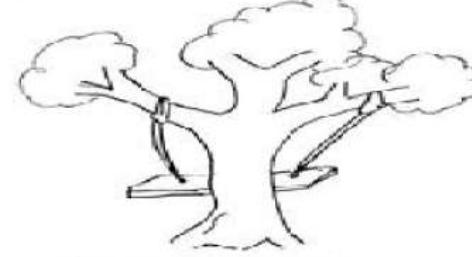
ce que demande l'utilisateur



ce qui est écrit dans le cahier des charges



ce que l'analyste a compris



ce que le programmeur a réalisé

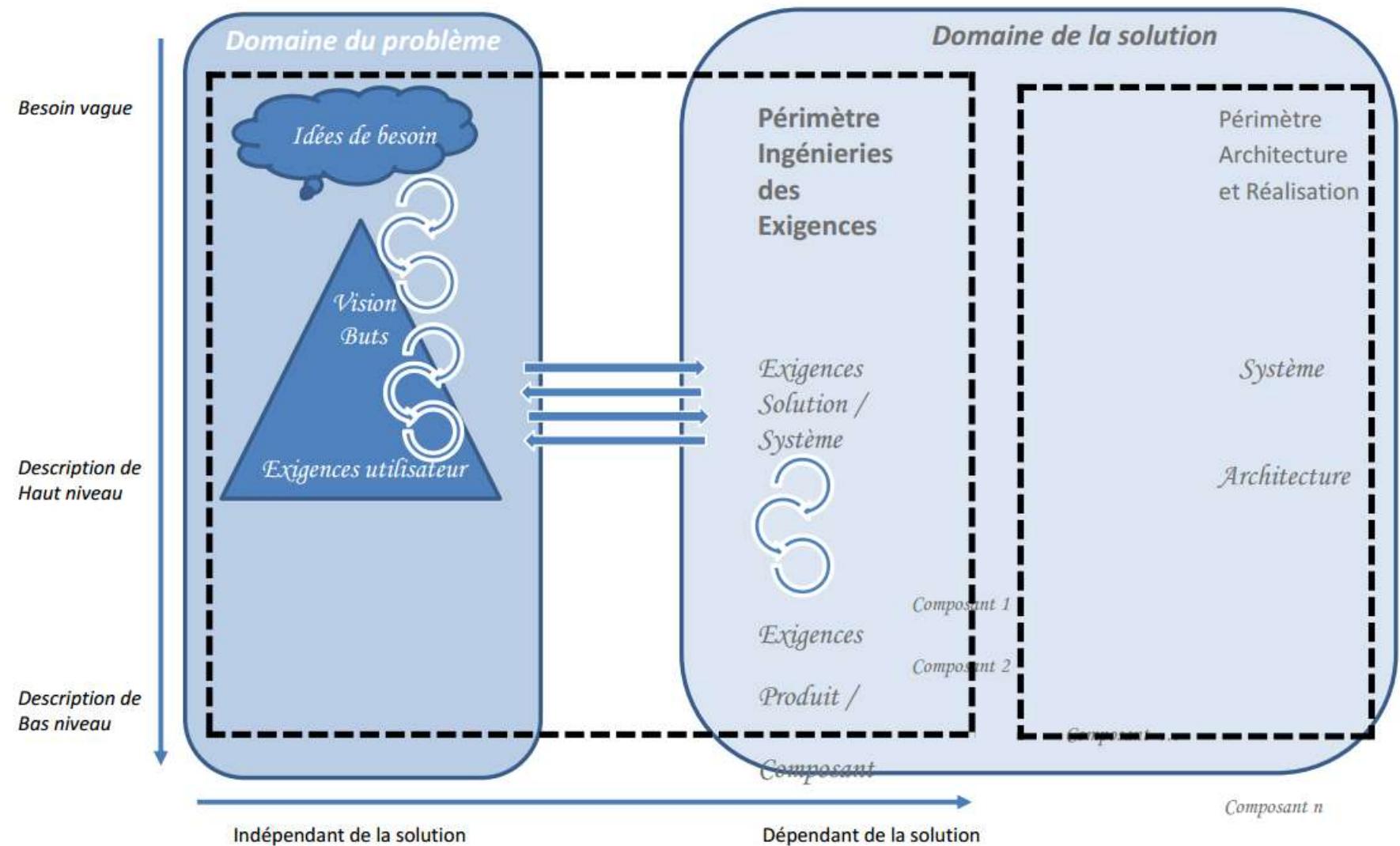


après la mise au point



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

Niveaux exigences :

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

Qualité des spécifications d'Exigences :

Complète
Cohérente
Modifiable
Traçable
Conforme au standard

Normes

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) – gerkhin

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





Exemple d'une Exigence

MGDIS Pilotage des aides - v9 / PLAID-71609

EAdm - Conditionner la création d'un Service Fait à certains types de paiement

Modifier Ajouter un commentaire Attribuer Plus Rédition Redressement Temps Gérer

Informations

Type:	Exigence	État:	VALIDATION (Afficher le workflow)
Priorité:	Mineur	Résolution:	Fini
Affecte la/les version(s):	Aucune	Version(s) corrigée(s):	9.2022.272-build236642
Composants:	Dématerrialisation et gestion des demandes de paiement	Niveau de sécurité:	Privé
Étiquettes:	Aucune		

Infos Documentation

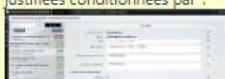
Equipe:	FTB
Lien d'épopée:	Demande de paiement - Gestion du cycle de vie des services faits
Nb Incident de test:	0

Description

EN TANT QU' administrateur
JE PEUX conditionner la création de services faits à certains types de paiement
AFIN QUE l'agent ne doive pas créer un Service Fait pour payer une avance (par exemple).

CA 1 : SI l'administrateur souhaite limiter la création de Service Fait à certains types de paiement **QUAND** il accède à au paramétrage d'un téléservice de paiement **ALORS** il peut sélectionner les types de paiement concernés.

RG 1-01 : Dans le paramétrage d'un téléservice de paiement, un nouveau champ de sélection multiple "Types de paiement exclus des services faits" s'affiche en dessous du champ "Dépenses justifiées conditionnées par".



RG 1-02 : Le champ "Types de paiement exclus des services faits" est non obligatoire et non renseigné par défaut à la création d'un téléservice de paiement.

RG 1-03 : Il permet de sélectionner un ou plusieurs de types de paiement parmi ceux actifs afin de ne pas pouvoir créer de services faits sur ces types de demandes.

RG 1-04 : Lors d'une duplication d'un téléservice de paiement, le paramétrage du champ "Types de paiement exclus des services faits" est dupliqué également.

RG 1-05 : Le champ "Types de paiement exclus des services faits" est associé à un tooltip d'aide :

Quand l'option "Gestion du Service Fait" est activée dans les sections du téléservice, cette option permet d'exclure la création de Service Fait sur certains types de paiement. Si aucun type de paiement n'est sélectionné, le Service Fait peut être créé sur tout type de paiement.

Personnes

Responsable:	Crestia Stéphanie
Me l'affecter	
Rapporteur:	GIGON Marie [X] (Inactif)
Votes:	<input type="radio"/> Voter pour ce ticket
Gérer les observateurs:	<input type="radio"/> Démarrer l'observation de ce ticket

Dates

Création:	05/mai/22 16:16
Mise à jour:	07/oct/22 09:17
Résolue:	09/sept/22 11:10

Suivi temporel

Estimé:	8j 5h
Restant:	0j
Consigné:	6j 1,5h
<input checked="" type="checkbox"/> Inclure les sous-tâches	

Agile

Afficher sur le Tableau



Exemple d'un référentiel d'Exigences

MGDIS – Pilotage des aides V9 : Référentiel des exigences

Demande de financement - Relations entre deux demandes PDA9-646

Demande de financement - Suivi PDA9-28

Demande de financement en CPO PDA9-509

Demande de paiement PDA9-34

Dossier de financement PDA9-32

Gestion des Instances PDA9-36

Gestion des Tiers PDA9-35

Gestion des chatbots PDA9-116

Demande de financement - Suivi

Espace Agents – Accéder à une demande de financement PDA9-198

Consulter et modifier les informations d'une demande de financement PDA9-241

Consulter la fiche de synthèse d'une demande de financement PDA9-209

Consulter les informations générales d'une demande de financement avec une contribution pour modification en cours PDA9-199

Consulter les informations générales d'une demande de financement PDA9-218

Consulter les minimis associés à une demande de financement PDA9-357

Modifier les informations générales d'une demande de financement PDA9-231

4. Des cas de tests pertinents



Fiche de test

Elle permet de recenser:

- L'objectif du test
- L'exigence rattachée
- Les prérequis...

Elle détaille:

- L'action à réaliser (étapes, pas de test)
- Le jeu de donnée
- le résultat attendu

En phase de test :

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 et ce que l'on obtient

- le résultat obtenu
- le résultat du test (ok ou ko)
- le numéro de la fiche anomalie
- le testeur

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



Quelques notions

Pas de test: une action simple dont le résultat est un élément vérifiable entraînant un changement d'état

- Pas de test
- Pas de test
- Pas de test
- Pas de test

- Cas de test : Ensemble de pas de test (15) validant tout ou partie d'une fonctionnalité
- Cas de test
- Cas de test
- Cas de test
- Cas de test

- Scénario : Ensemble de cas de test couvrant une partie des fonctions de l'application
- Scénario
- Scénario
- Scénario

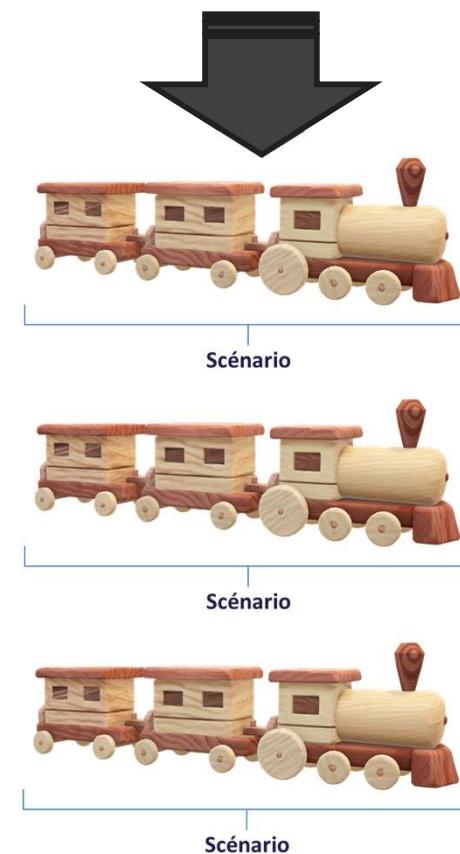
Campagne de tests :
tests du lots



Un scénario de test :

- Est un ensemble de cas de test mi bout à bout
- Permet de tester une fonctionnalité de bout en bout ou un cas d'utilisation

Campagne





Exemple de cas de test

Espace Cas de test

Screenshot of the 'Espace Cas de test' interface showing a tree view of scenarios and steps:

- PDA-Bibliothèque de scénarios
- PDA-Exécution TAR
- PDA-FT
 - 0 - Gestion lot
 - 1 - Gestion des paiements
 - Declaration
 - Instruction
 - 3.14 - Instruction de la proposition de paiement
 - 3.14.01 - Calcul du montant proposé au paiement
 - 3.14.01.01 - EA - Calcul d'un montant proposé au paiement**
 - 3.14.01.02 - EA - Renseigner le montant proposé directement
 - 3.14.02 - Contrôle de complétude d'instruction
 - 3.14.02.01 - EA - Contrôle progressif de la complétude d'instruction
 - 3.14.02.02 - EA - Contrôle de la complétude d'instruction sur choix de l'avis du s
 - 3.14.02.03 - EA - Informations obligatoires de la section Avis d'instruction non re
 - 3.14.02.04 - EA - Contrôle de la complétude d'instruction - proposition de paiem
 - 3.14.02.05 - EA - Contrôle de la complétude d'instruction sur domiciliation banca
 - 3.14.02.06 - EA - Contrôle de la complétude d'instruction - domiciliation bancaire
 - 3.14.02.07 - EA - Contrôle de la complétude d'instruction-Conformité des pièces
 - 3.14.02.08 - EA - Contrôle de la complétude d'instruction-demande de pièces co
 - 3.14.03 - Pièces justificatives de paiement
 - 3.14.03.01 - EA - Ajouter une pièce justificative
 - 3.14.03.02 - EA - Générer un document depuis la section pièces justificatives
 - 3.14.03.03 - EA - Supprimer une pièce justificative de paiement
 - 3.14.03.04 - EA - Télécharger une pièce justificative de paiement
 - 3.14.04 - Validation de la proposition de paiement
 - 3.14.05 - Informations financières associées au paiement
 - 3.30 - Gestion des instances délibérantes
 - 3.31 - Gestion des instances consultatives

Screenshot of the '3.14.01.01 - EA - Calcul d'un montant proposé au paiement' test step details:

Created on: 16/06/2021 09:09 (bousquet-f)
Modified on: 17/06/2021 17:00 (bousquet-f)

Actions: Renommer, Imprimer

Informations, Pas de test (selected), Paramètres, Pièces jointes, Exécutions

Reduire, Ajouter un pas de test, Appeler un cas de test, Copier, Coller, Supprimer

#	Ex.	PJ	Actions	Résultat attendu
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sur l'espace Agent, ouvrir le menu dossier de la demande d'aide et cliquer sur le sous-menu Paiements	<p>Le tableau contenant la demande de paiement est affiché dans la section liste des paiements avec les colonnes suivantes :</p> <ul style="list-style-type: none"> libellé : libellé saisi lors de la création statut : ici en cours d'instruction type : celui renseigné lors de la création montant prévisionnel : non obligatoire date prévisionnelle : non obligatoire montant à payer : non obligatoire montant payé : non obligatoire <p>En bout de ligne 2 icônes :</p> <ul style="list-style-type: none"> crayon pour édition de la demande de paiement poubelle pour suppression de la demande de paiement
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cliquer sur l'icône d'édition de la demande de paiement	Ouverture du formulaire de la demande de paiement : libellé de la demande en haut de page avec le statut en cours d'instruction
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Dans la section Calcul du montant de la proposition de paiement : saisir un montant des dépenses justifiées de 2€ , un taux de 30% et cliquer sur calculer	Le champ Montant proposé est renseigné : 0.60€
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cliquer sur enregistrer dans la section calcul du montant de la proposition de paiement	Le calcul est enregistré et le montant est reporté dans le champ montant proposé de la section Avis d'instruction

Afficher 50 éléments :



Exemple d'une campagne de test

Espace Campagnes

2 - EA Envoi demande pour redirection à un téléservice

Créé le : 22/04/2021 11:21 (robin-h)
Modifié le : 22/04/2021 11:25 (robin-h)

Relancer Suites de tests Renommer

#	Emplacement	Ref.	Test	Imp.	Suite de tests	Statut	% succès	Utilisateur	Dernière exécution
1	PDA-FT		▶ 2-1 P1 EA - Accès à la création d'une contribution pour redirection (PLAID-47501 RG 1-01, 3, 4, 6)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	28/04/2021 15:32
2	PDA-FT		▶ 2-2 P1 EA - Crédit d'une contribution pour redirection sur un téléservice (CA1 et 2 PLAID-47534)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	21/06/2021 14:17
3	PDA-FT		▶ 2-3 P1 EA Etape personnalisation du courriel contribution pour redirection (CA3 PLAID-47534)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	30/04/2021 09:50
4	PDA-FT		▶ 2-4 P1 Vérification de l'envoi de la contribution pour redirection (CA4 PLAID-47534)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	30/04/2021 09:58
5	PDA-FT		▶ 2-5 P1 EA Accès à la contribution pour redirection (PLAID-47534 CA6 et PLAID-47501 RG 1-08)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	29/04/2021 09:54
6	PDA-FT		▶ 2-6 P1 EA Accès à une demande ayant une contribution pour redirection (PLAID-47534 CA5)	F	-	succès	100 %	Guylaine-Clémentine MALIEDJIE TASSING (tassing-g)	30/04/2021 17:26



Exemple de cas de test

Action	Résultat attendu
Cliquer sur le bouton Création de compte	<p>- Le champ Objet du courriel est obligatoire et renseigné par défaut avec la valeur suivante :</p> <p><i>Espace Usagers - Activation de votre compte</i></p> <p>- Le champ Contenu du courriel est renseigné par défaut avec la valeur suivante :</p> <p><i>Bonjour {{user.name.honorificPrefix}} {{user.name.givenName}} {{user.name.familyName}},</i></p> <p><i>Nous venons de recevoir une demande de création de compte à votre nom sur l'espace de la collectivité.</i></p> <p><i>[...]</i></p>



Construire les cas de test

Il existe plusieurs techniques de conception de cas de test

- Tests passants
- Tests aux limites
- Test basés sur les états
- Tests basés sur les incidents....

La construction des cas de test doit souvent être évaluée au regard du budget et des risques. Une analyse de risque vient ici cibler les efforts de test à réaliser.

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:

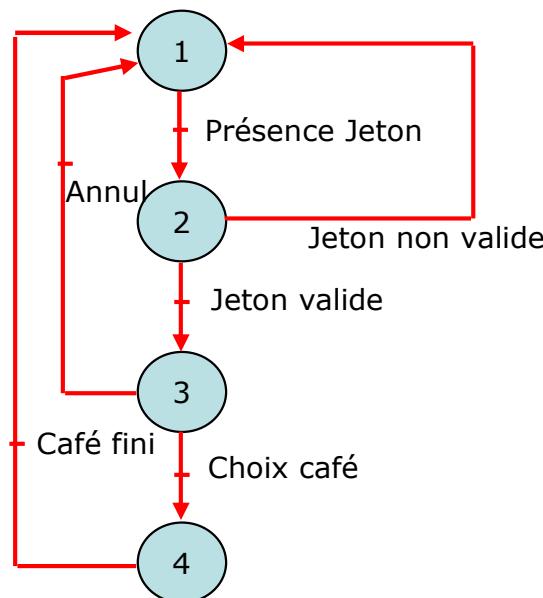
1. Identifier les différents états du système et les conditions des transitions entre états
2. Etablir la matrice de transitions, puis le graphe d'état
3. Identifier les circuits indépendants à couvrir par des cas de test de telle sorte à couvrir le graphe: **Nb de circuit = nombre d'arcs – nb états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

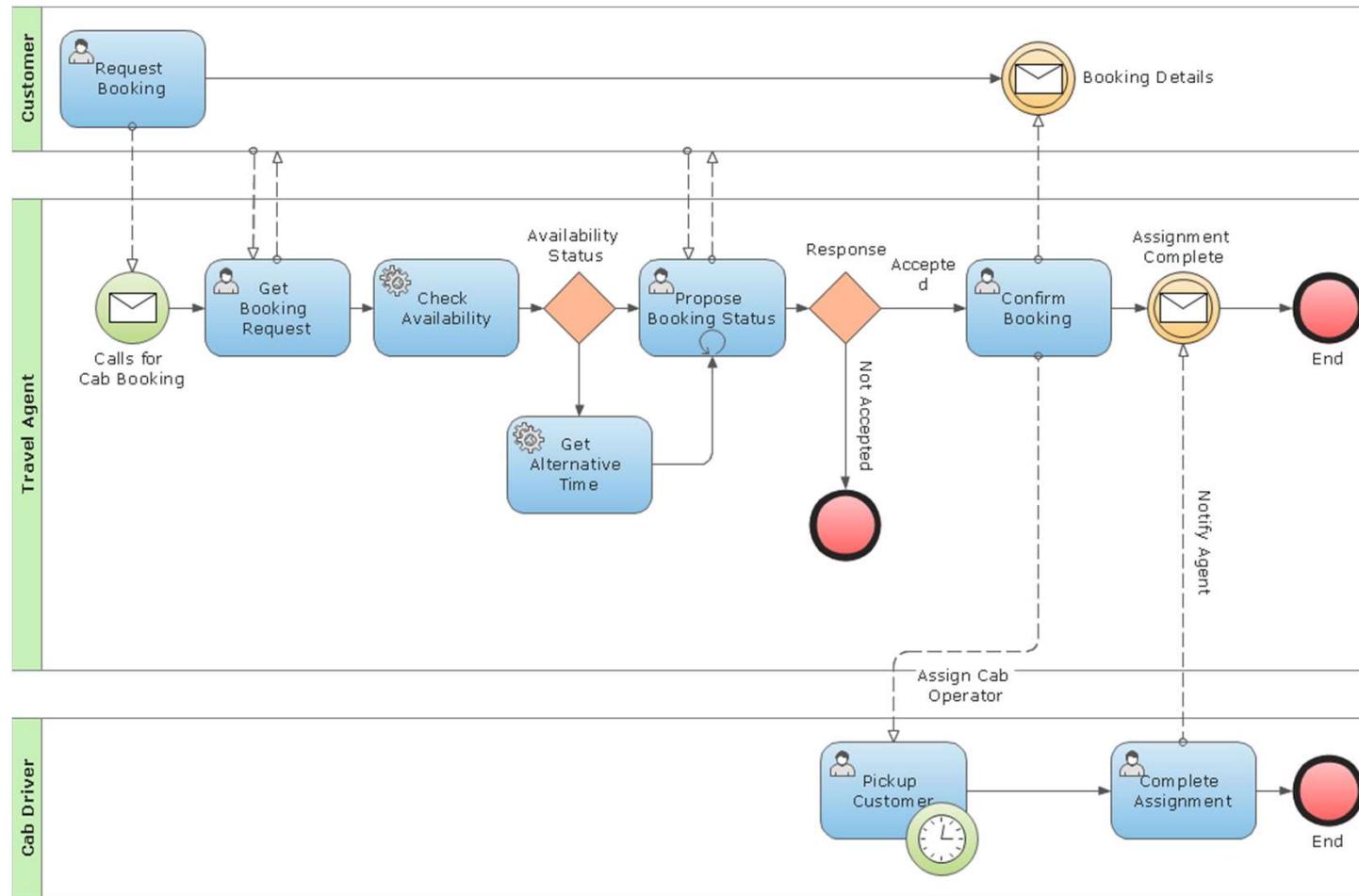
	1	2	3	4
1	X	Présence Jeton		
2	Jeton non valide	X	Jeton valide	
3	Annulation		X	Choix café
4	Café fini			X

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

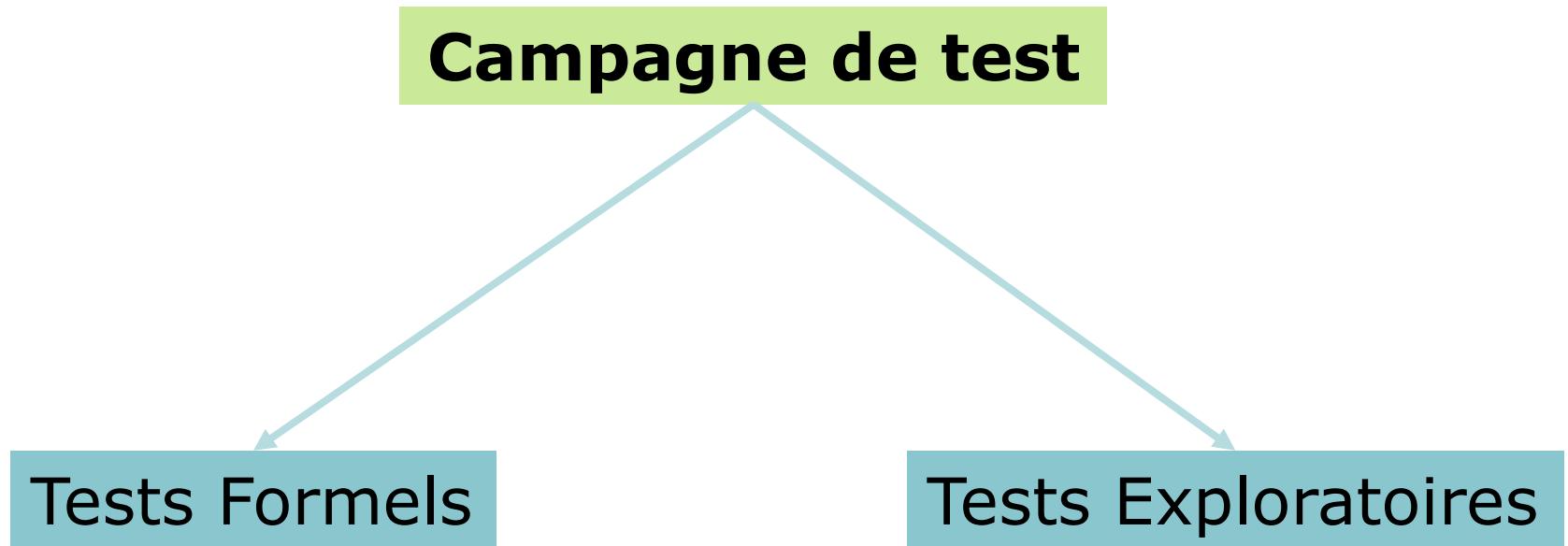
→ Cela permet d'avoir un retour rapide sur la compréhension (spécification par le visuel)

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





Exécution des tests

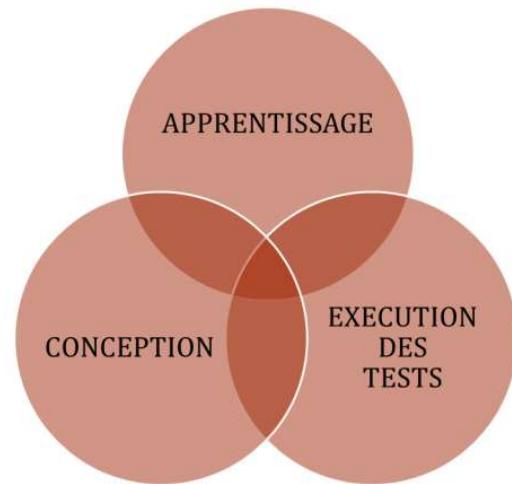




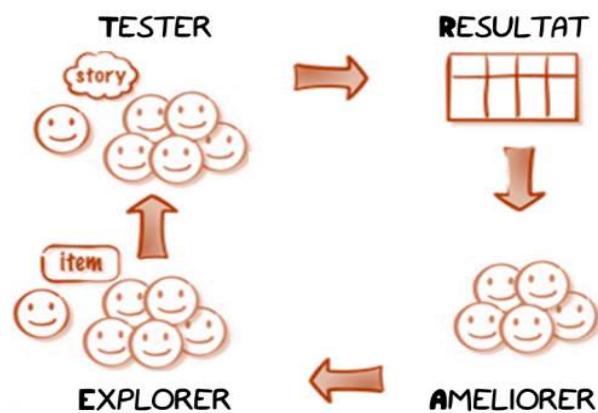
Exécution des tests

Tests Exploratoires

Run new tests as you think of them!



le principe du test est de **découvrir**
le logiciel, dans le même temps le **testeur conçoit les tests et les exécute.**
Il est intéressant pour rechercher des défaillances et donner une tendance de qualité.



Méthode agile centrée sur le **résultat**
pas de conception et peu de documentation à prévoir en amont
Facile à mettre en œuvre à chaque sprint, pas de cadre précis
Peu couteuse en temps et en argent

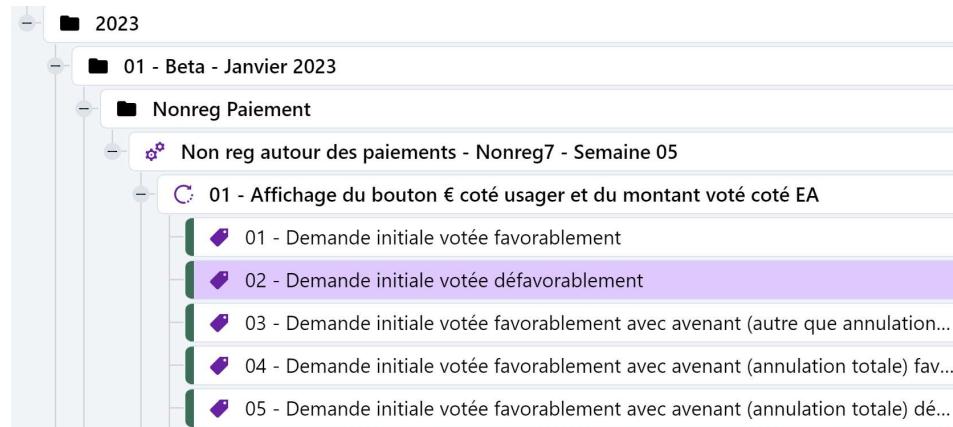


Basée et dépendant des **compétences du testeur**
le testeur laisse libre court à sa **créativité** et son bon sens pour établir de manière autonome les tests qui lui sens pertinents d'un point de vue utilisateur.
Difficile de référencé précisément le scénario établi, la zone de couverture



Exécution des tests

Tests Formels



Le test est écrit avant l'exécution (et même la conception). Il permet de décrire ce que l'on va faire, comment on va le faire. C'est un test structuré

Il est intéressant pour donner un niveau de confiance sur le produit



Suivi de l'avancement et de la couverture de test
Plus facile pour distribuer le test sur plusieurs testeur et éviter les collisions de test



Nécessite de tout décrire et organiser avant de tester
Nécessite une certaine technique de test



Indicateur de mesure

Un des principaux indicateurs de mesure:

- Couverture de code par les tests
- Couverture des exigences par les tests
- Avancement des test
- ...

Attention 100% de couverture de code ne signifie en aucun cas un bon fonctionnement de votre application

- Elle peut ne pas répondre au besoin
- Elle doit être vérifier aux limites, en erreur

- Exemple:

```
function sqrt(num) {
    return Math.sqrt(num);
}
module.exports = sqrt;
```

Le test

```
const assert = require('assert');
const sqrt = require('../sqrt');

describe('sqrt', function() {
    it('should return the square root of a positive number', function() {
        assert.strictEqual(sqrt(4), 2);
    });
});

it('should return NaN for negative numbers', function() {
    assert(isNaN(sqrt(-4)));
});

it('should return 0 for 0', function() {
    assert.strictEqual(sqrt(0), 0);
});
```

Résultat et couverture:

```
$ ./node_modules/.bin/istanbul cover ./node_modules/mocha/bin/_mocha

sqrt
  ✓ should return the square root of a positive number

  1 passing (6ms)

=====
Writing coverage object [C:\test\res\coverage\coverage.json]
Writing coverage reports at [C:\test\res\coverage]
=====

===== Coverage summary =====
Statements 100% 3/3 )
Branches   100% 0/0 )
Functions   100% 1/1 )
Lines      100% 3/3 )
```



Résultat 100% et pourtant nous avons 2 autres cas à tester

5. Un banc de test



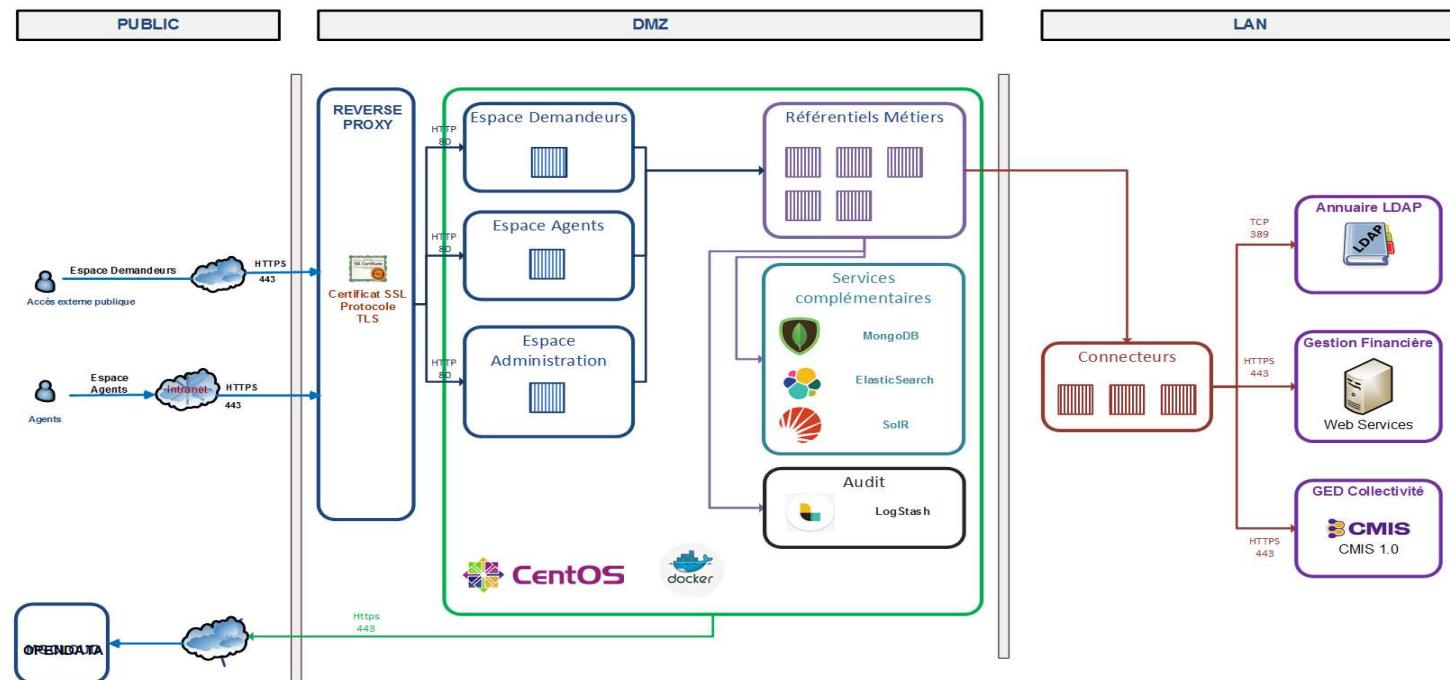
Le Banc de test

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



Outilage de tests

Comment choisit on un outil de tests ?

Facilité d'utilisation

(Enregistreur, Langage de développement, Mots clés)



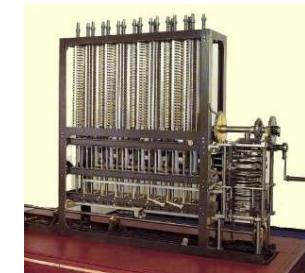
Intégration entre les outils
(Suite logicielle, interopérabilité, ALM)



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

Des Outils, mais pour faire quoi?

Management
des tests

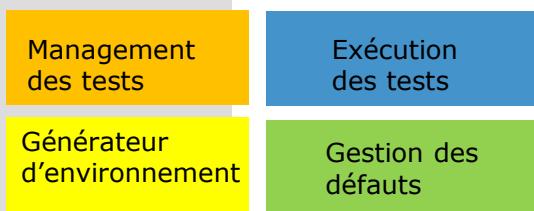
Exécution des tests

Générateur
d'environnement

Gestion des défauts



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...

>> Cours Tests logiciels

Squashtm

Espace Exigences

Bibliothèque Recherche

Exigence : 3. Synthèse territoriale

Créé le : 21/02/2013 15:36 (labarre-a)
Modifié le : 07/06/2013 12:38 (labarre-a)

Informations Pièces jointes

Renommer Supprimer Crée une nouvelle version

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 sélectionnée 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 : [\(Cliquer pour éditer...\)](#)

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		Affichage des actions d'une commune ayant 0 actions	manuelle
2	Mobilité - Territoire		Affichage des actions d'une commune sélectionnée	manuelle
3	Mobilité - Territoire		Affichage des actions du canton sélectionné	manuelle
4	Mobilité - Territoire		Affichage des informations d'une action	manuelle
5	Mobilité - Territoire		Affichage des informations d'une action à 0	manuelle
6	Mobilité - Territoire		Affichage du graphique avec 1 année à 0	manuelle
7	Mobilité - Territoire		Passage au détail des aides	manuelle
8	Mobilité - Territoire		Retour à la sélection du territoire	manuelle

Afficher 50 éléments :

Historique des modifications



Outilage de tests

Management
des tests

Exécution
des tests

Générateur
d'environnement

Gestion des
défauts

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





Outilage 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 Atlassian JIRA web interface. The title bar reads "Issue Navigator - Atlassian JIRA". The main area is titled "Issue Navigator" and displays a table of issues. The table has columns for Key, Summary, Assignee, Reporter, Status, Resolution, Created, Updated, Due, Votes, and Backlog Order. There are 123 matching issues listed, with the first few rows visible. Each row contains a small thumbnail image, a key identifier (e.g., JRA-568, JRA-14076), a summary description, the assignee (e.g., Unassigned, Brad Baker, Chris Mountford), the reporter (e.g., Primoz Prislan, Brad Baker, Jack Low), and various status indicators (e.g., Resolved, Fixed, Open). The interface includes a search bar at the top and navigation links like "Dashboards", "Projects", "Issues", "Agile", and "Administration".



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



>> Cours Tests logiciels

■ 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

Visité récemment: 0000098, 0000097

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

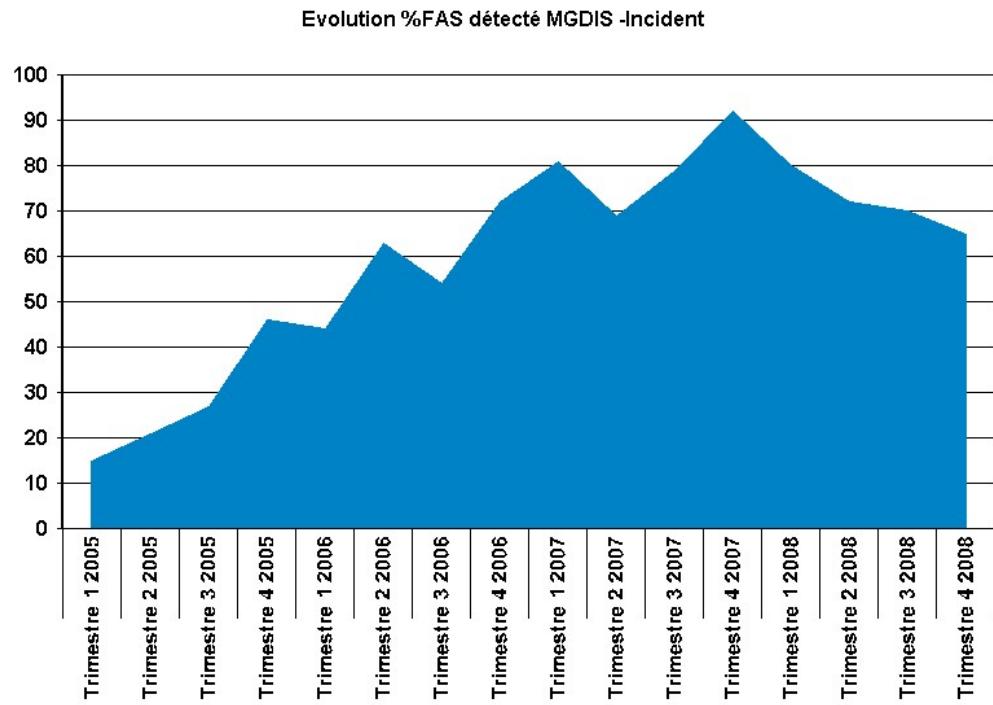
Bogue # Aller

Visualiser les détails du résultat [Aller aux notes] [Envoyer un rappel] [Historique du bogue] [Imprimer]

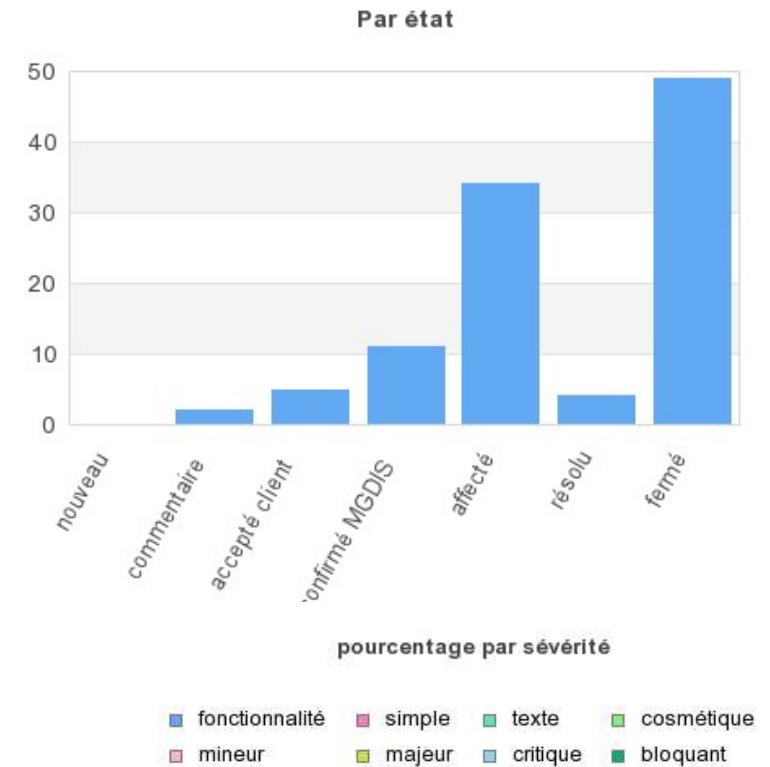
Identifiant	Projet	Catégorie	Afficher l'état	Date de soumission	Dernière mise à jour
0000057	CRAU - Projet 25531 - PROGOS Aides Web	[Tous les projets] Anomalie	public	2013-10-08 13:18	2013-10-08 13:32
Rapporteur	jamette-h				
Assigné à					
Priorité	normale	Impact	majeur	Reproductibilité	Aléatoire
État	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 icone "masquer editions", 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 réinitialisement 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 icone "masquer editions", elles réapparaissent				
Balises	Aucune balise n'est associée.				
Associer balises	(Séparer par ',')	Balises existantes	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					

>> Cours Tests logiciels

■ Exemple de synthèse dans Mantis



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





Outilage de tests

Management
des tests

Exécution
des tests

Générateur
d'environnement

Gestion des
défauts

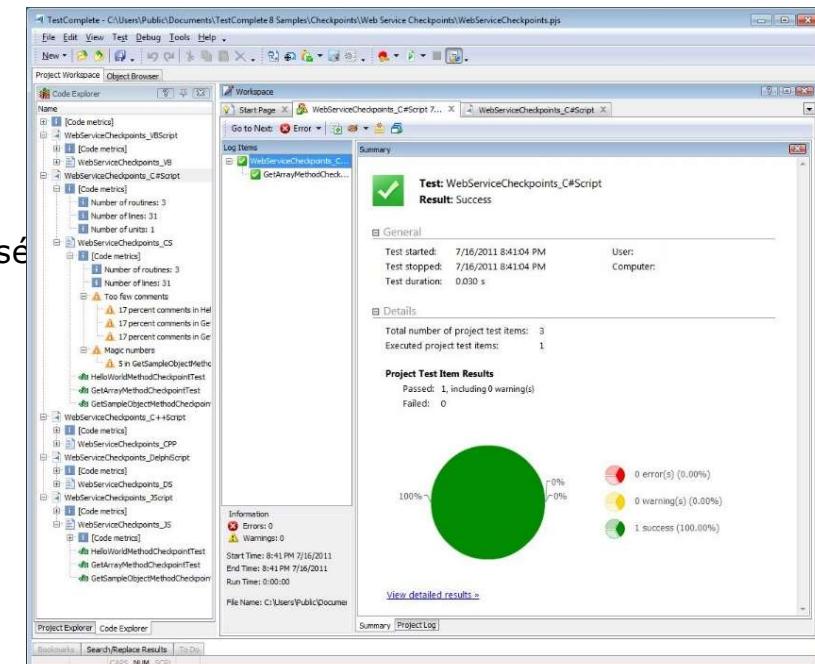
Exécution des tests

Objectifs

Jouer et rejouer le scénario 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)

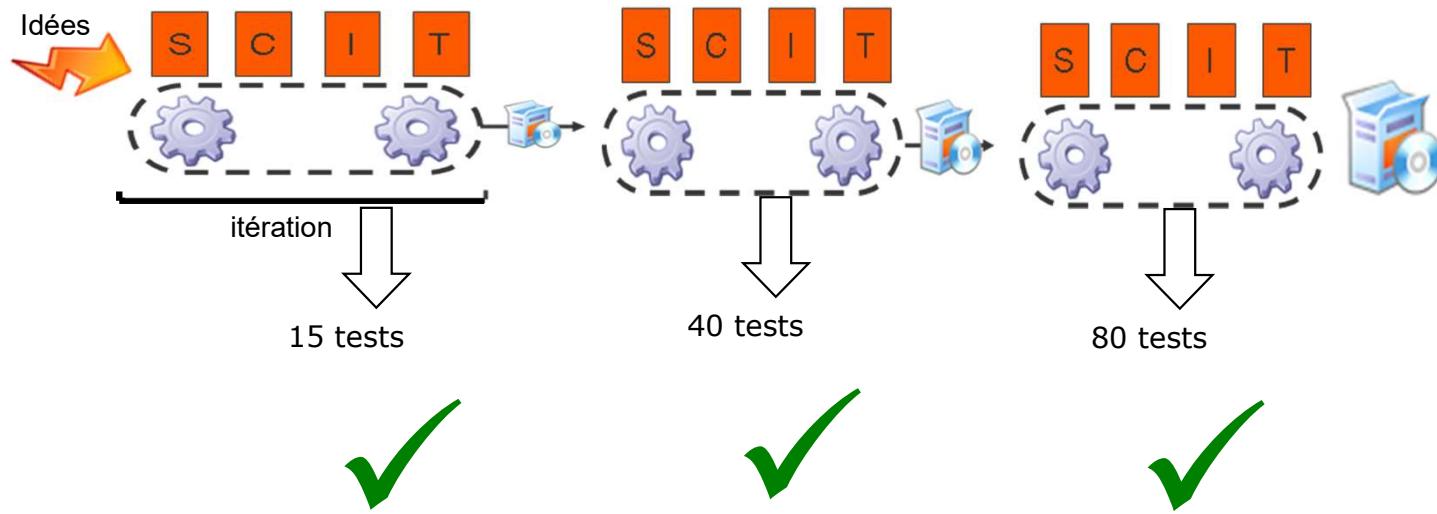
■ Prévisibilité et cohérence

- > 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

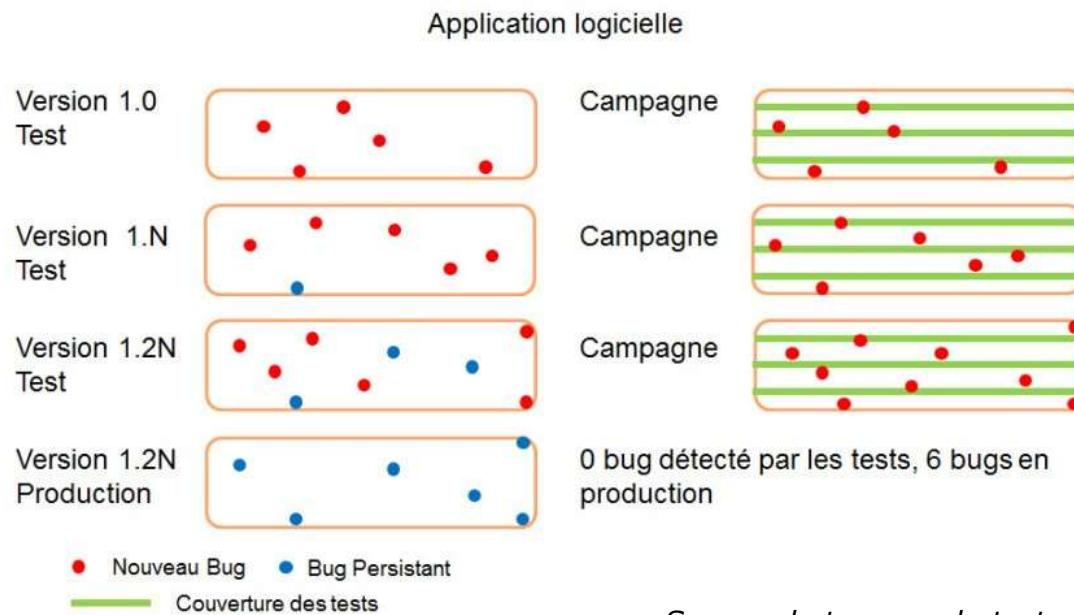
■ Productivité

- > 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

Bénéfice de l'automatisation des tests sur des cycle agiles



Paradoxe du pesticide



Source: *la taverne du testeur*

Paradoxe emprunté au monde agricole. Il nous explique qu'à faire toujours les mêmes tests, on ne découvre pas de nouveaux défauts. Il nous rappelle également que tout miser sur les tests automatisés n'est malheureusement pas une solution mais surtout que ce qui est important dans un logiciel c'est ses utilisateurs. Il faut les écouter, comprendre ce qu'ils attendent et s'adapter à leurs demandes. Pour cela de bons

- 1 Explorer les logs
- 2 Travailler avec les utilisateurs
- 3 Créer un test sur découverte d'un bug important (Rien de pire que ce bug revienne)
- 4 Faire évoluer les campagnes (retirer certains cas, en ajouter d'autres...) afin de s'adapter au logiciel et à ses potentielles faiblesses.
- 5 Faire des tests exploratoires

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

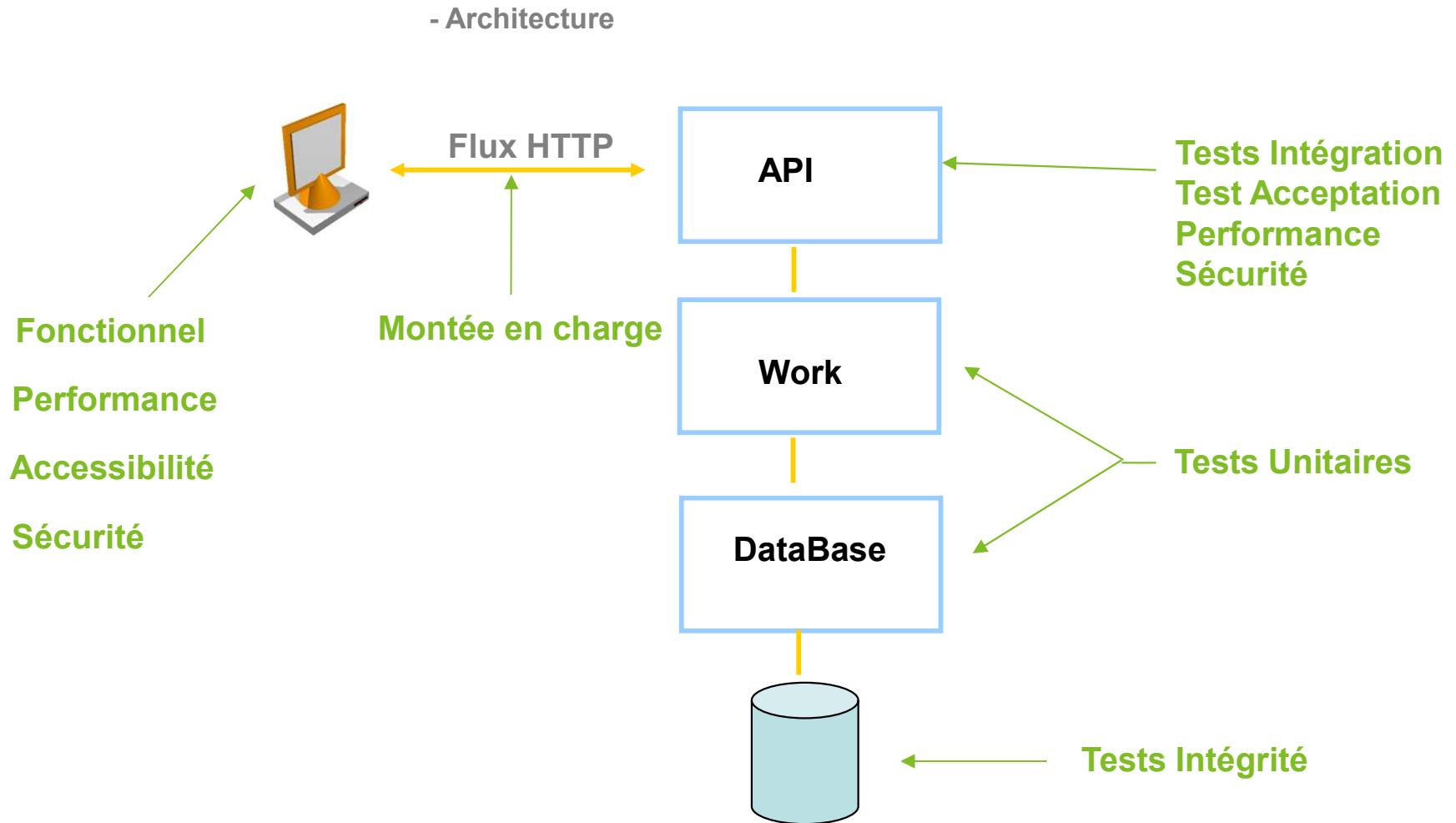
■ Les bonnes pratiques

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





Outilage de tests

Comment choisit on un outil de tests ?

Facilité d'utilisation

(Enregistreur, Langage de développement, Mots clés)



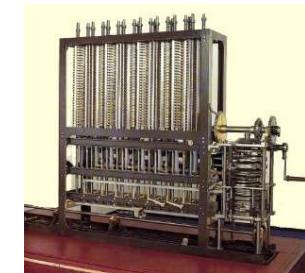
Intégration entre les outils
(Suite logicielle, interopérabilité, ALM)



Maturité (obsolescence de certains outils, projet open source)



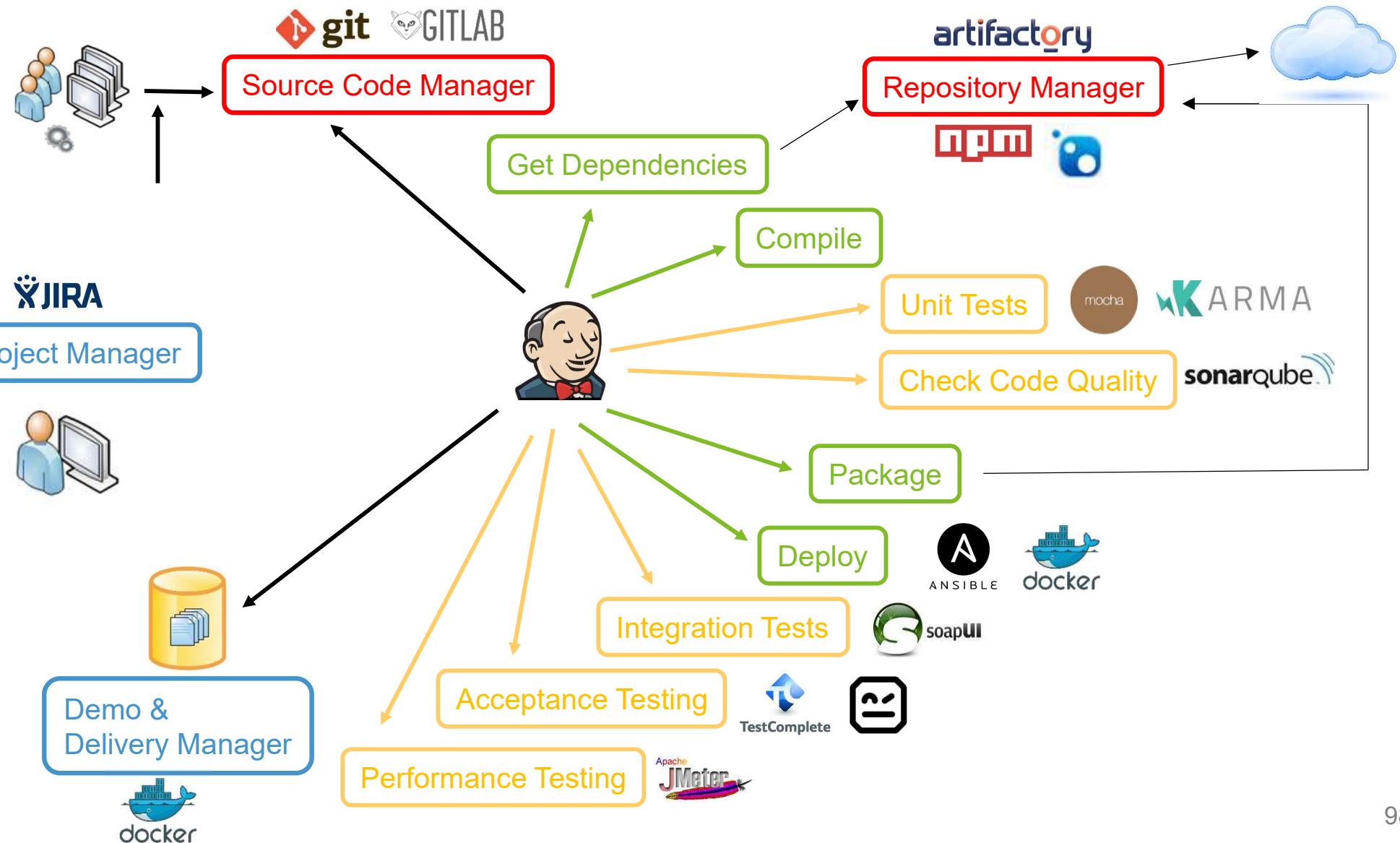
Cout (Rentabilité ne veut pas nécessairement dire Moins Cher, on cherche le ROI)



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

7. Une usine pour produire

→ Pour construire, il faut une usine de Production logicielle



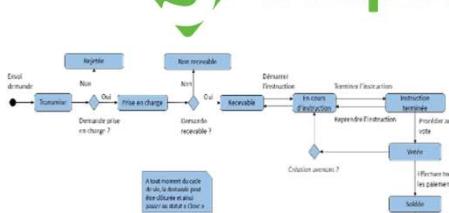
Usine de tests



Exécution de scénarii de tests utilisateur



Gestion des tests de montée de version



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



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



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_COMpte	ESPACE_AGENT_GLOBAL	RATTACHEMENT	soapui	ZAP	jmeter		
PORTAIL_DE_LA_HAUTE_EDUCATION	Sep 03 02:15	No Changes	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

→ Tester en condition d'exploitation

Intégrité de l'environnement + configuration

Monitoring

Alerting / AlertManager

Critical alerts	Other alerts	High Memory Nodes	High CPU Nodes	High Storage Nodes	High HTTP 5xx Nodes	Admin status Nodes	Tenant status Nodes
1	15	2	0	0	0	0	10

Alerts for Astreinte

Time	alertname	job	description	level	class	exported_type
2021-01-13 18:15:28	portail_aides_tenant_status_response_time_warning	demo-prod-sddc.mgcloud.fr	Problème de temps de réponse 2532ms du statut tenant du service account-management du tenant prod-agents.	Warning	astreinte	
2021-01-13 18:11:25	redis_message_queue_waiting_warning	demo-prod-sddc.mgcloud.fr	Le nombre moyen de 4.63k messages en attente pour le clef bq:webhooks:waiting sur la dernière heure de Redis est important.	Warning	astreinte	
2021-01-13 17:33:28	portail_aides_tenant_status_response_time_warning	demo-prod.mgcloud.fr	Problème de temps de réponse 3269ms du statut tenant du service account-management du tenant croccitanie-agents.	Warning	astreinte	
2021-01-13 14:37:28	portail_aides_tenant_status_response_time_warning	demo-prod.mgcloud.fr	Problème de temps de réponse 3509ms du statut tenant du service account-management du tenant croccitanie-demandeurs.	Warning	astreinte	

Système de contrôle et d'alerte

100

→ Tester la sécurité : Pentest

Objectif: Tester la vulnérabilité d'un système informatique et la capacité de résistance offerte par l'application

Le scan de vulnérabilité peut faire partie du pentest, mais ce test d'intrusion va bien au delà puisqu'il démontre l'exploitation des failles

Le pentester doit se mettre dans la peau de l'attaquant pour mettre en évidence un chemin d'attaque

L'audit de sécurité n'est pas un test d'intrusion



Attention, sans consentement, ce test est illégal et peut entraîner des poursuites pénales.

Méthodologie

White box

Grey Box

Black Box

Démarche

1. Cadrage du scénario d'audit
2. Exploration de la cible
Reconnaissance
Collecte d'information
3. Recherche de vulnérabilité
Défaut
10 Owasp
Configuration
4. Exploitation



CommitStrip.com