

Telecom  
valley

Soirée du  
<Test Logiciel>

REX création et diffusion d'un jeu pour  
le test

Marc Hage Chahine

Mardi 6 décembre 2022

sogeti  
Part of Capgemini



# Votre présentateur en quelques lignes



## Activité publique

- Animateur et créateur du blog de La taverne du testeur
  - Ecriture d'articles
  - Animation de Webinaires
  - Accompagnement des « invités »
  - 1000 vues par jour en moyenne
- Ecriture d'articles pour des magazines
  - Programmez!
  - QA matters, SQ mag
- Participation à l'organisation d'événements
  - JFTL
  - STLS
- Conférencier
  - STLS, JFTL

## Activité professionnelle

- Expert test méthodes et outils
  - Audits
  - Coaching
  - Accompagnement
  - Formation test
  - Formation Agile
  - Organisation d'événements
  - Animation de communautés
- Formateur
  - Conception d'ateliers
  - Cours à l'université et école d'ingénieur





# Sommaire

- 1 Genèse
- 2 Création des 1001 tests
- 3 Les règles
- 4 REX les 1001 tests
- 5 On continue!
- 6 Questions

1.

Genèse

## 2018: Idée Sandrine Domagala

S'inspirer du 1000 borne pour former sa nouvelle équipe au test

Quelques échanges et prototype fait: pas le temps ni le budget pour aller plus loin

Développement de la « gamification »

Plus de notion de points

« jeux de cartes »

Serious games...

Mais pas de vrai jeu  
=> frustration de mon « Moi » joueur

Mise en forme d'une théorie de création d'un jeu pour le test

Article dans Programmez

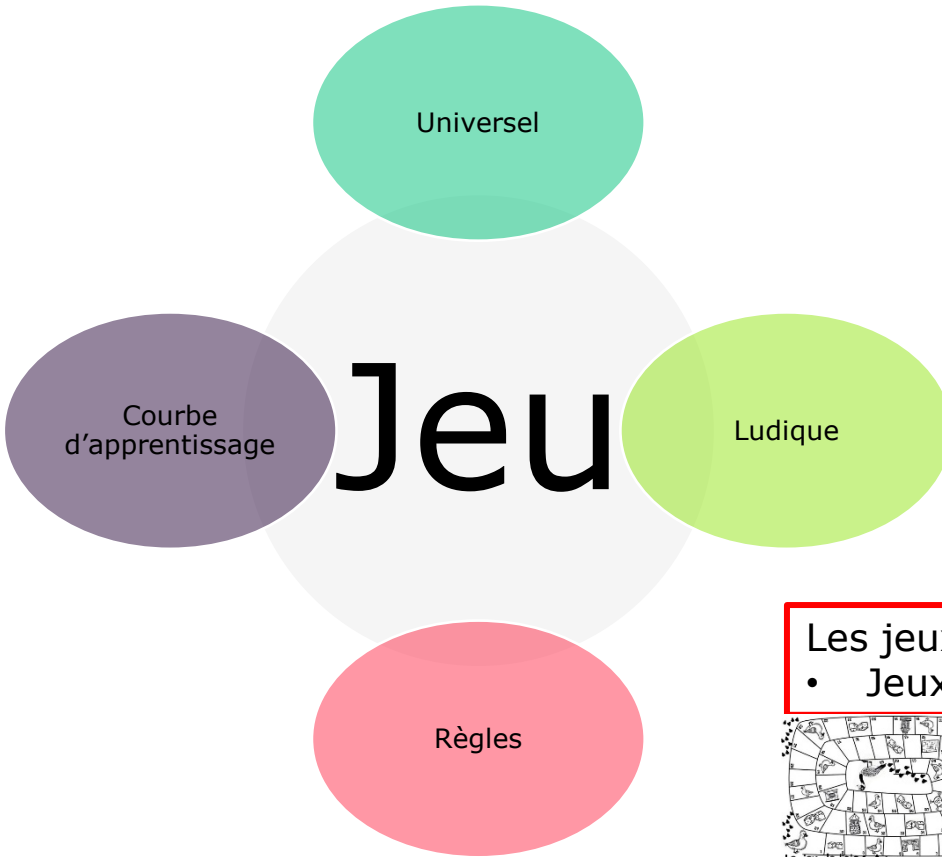
Présentation à des communautés de pratiques et à la STLS



Passer de la théorie à la pratique

Reprise de l'idée de 2018  
Pousser le concept et proposer un « vrai » jeu

# Genèse: qu'est ce qu'un jeu ?



## Universel:

- Tout âge
- Tout public (même certains animaux!)

## Ludique:

- Fun
- Envie de faire une partie

## Règles:

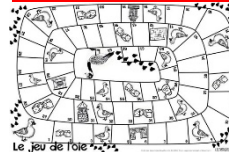
- Cadre défini

## Courbe d'apprentissage:

- On s'améliore avec l'expérience

## Les jeux inclus:

- Jeux de société, Jeux vidéos, Sports...





# Pourquoi un jeu pour le test ?

## Introduire au test

- Faire connaître ce métier et de susciter des vocations
- Aider des professionnels à bien débuter dans le test
- Faire comprendre à des non professionnels certains aspects de ce métier

## Développer des compétences spécifiques

- Compétences de planification
- Compétences de communication et de collaboration
- Compétences de recherche et d'observation
- Compétences de synthétisation....

## Donner une bonne image du test

- A travers des moments de partage

# 2.

Création des 1001 tests





# REX les 1001 tests: définition du public cible

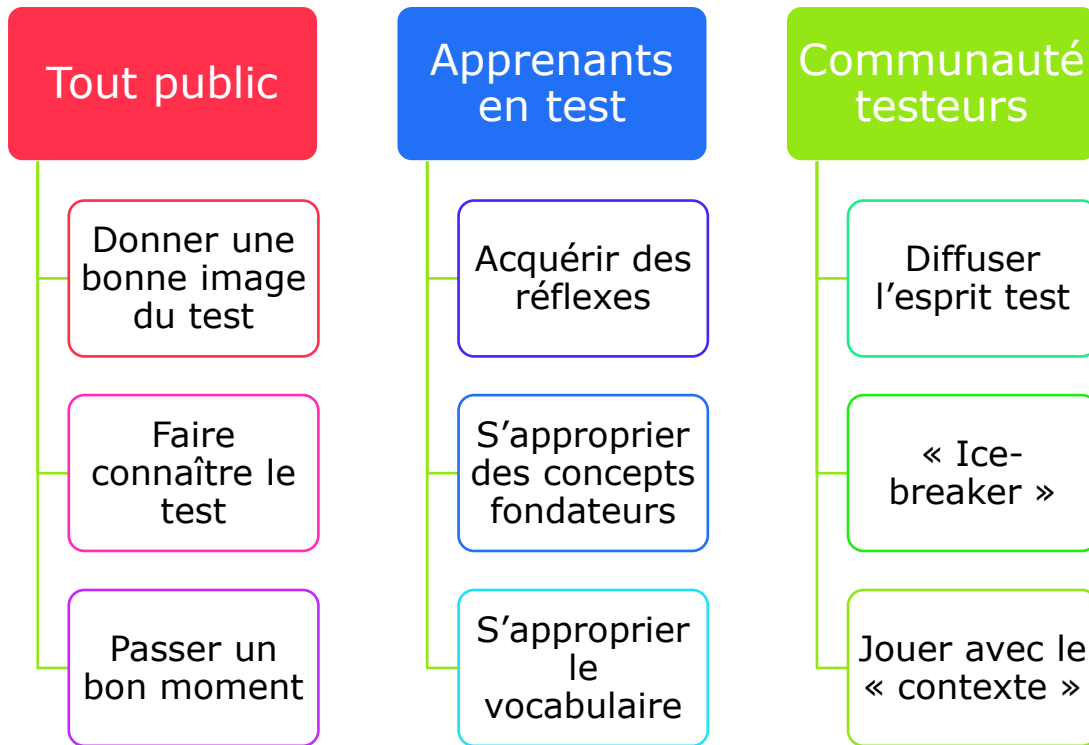
Tout public

Apprenants  
en test

Communauté  
testeurs



# REX les 1001 tests: définition du but



# REX les 1001 tests: conception du jeu

Inspiration

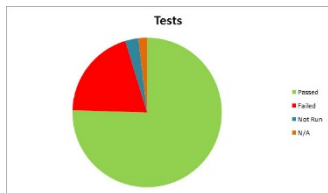


Sujet du test sélectionné

Notions à mettre en avant

Sélection des termes, cartes, points

Confrontation, ajustements, inédit



**Alerte Qualité !!**

## Jeu connu et à succès:

- A prouvé son efficacité
- Grand potentiel
- Points de vigilance: potentielles lenteurs, possible d'être « bloqué »

## Les campagnes de test:

- Représente bien le test
- Des tests à exécuter
- Une fin concrète

## Notions de base:

- Imprévus souvent rencontrés avec des solutions concrètes
- Pyramide des tests
- Diversité des tests
- Bonnes pratiques...

## Conception:

- Nom et effet des cartes
- Nombre de chaque carte
- Règles
- Variantes
- Vocabulaire...

## Echanges:

- Echanges et présentations avec des testeurs / joueurs
- Adaptation des règles

# REX les 1001 tests: expérimentation du jeu / ajustements

Création de cartes  
« MVP »

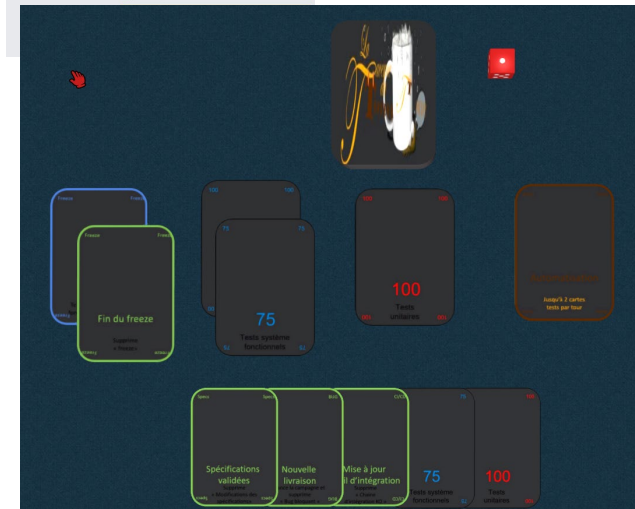
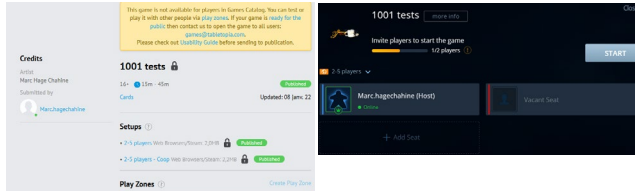
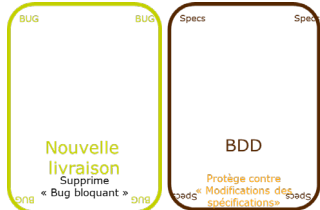
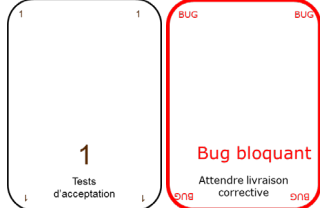
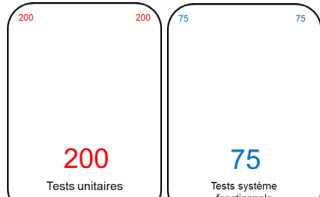
Numérisation du  
jeu sur tabletopia

Ateliers jeu de  
« base »

Ajustements

Ateliers  
« variantes »

Ajustements et  
validation finales



## Exemples d'ajustements:

- Nom des cartes
  - Effets de certaines cartes
  - Etudes de règles
- => Équilibrage du jeu

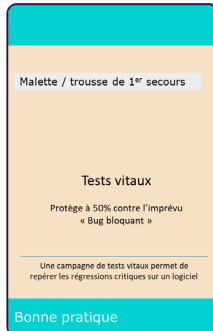
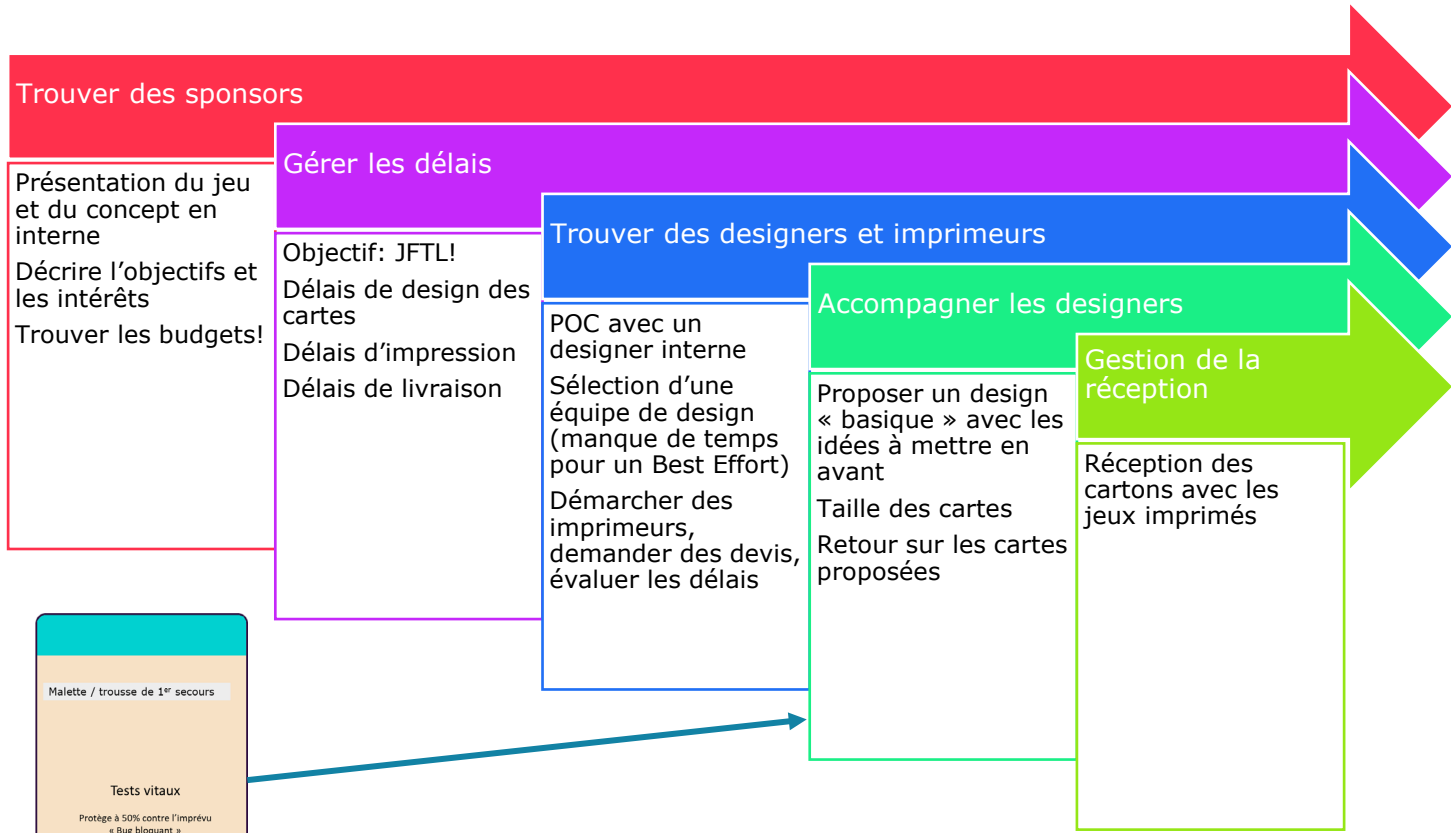
## Exemples d'ajustements:

- Explication de la raison des effets sur les cartes
  - Cartes « règles » récapitulative
- => Ajustements principalement ergonomiques!

## Variantes:

- Mode coopératif
- Mode itératif
- « A l'échelle »
- Adaptation de règles
- Test unitaires en 1<sup>er</sup>
- Produit critique...

# REX les 1001 tests: design, impression, diffusion



# REX les 1001 tests: résultats

## Caractéristiques du jeu

- Parties de 15 minutes en moyenne
- 126 (dont 29 uniques) + 15 (règles) cartes
- Nombreux modes de jeu
- Forte rejouabilité
- Facilité de prise en main
- Peu de temps mort

## Coûts / délais

- Objectif JFTL tenu!
- Besoin de compétences externes (design)
- Besoin d'un sponsor (équipe externe + impression + aide à la priorisation)
- Besoin de temps (conception, expérimentation, accompagnement)
- Besoin de « testeurs » du jeu

## Autre

- Forte attente du milieu académique, interne et clients
- Très bons retours des bêta testeurs
- Souhait: utilisation de ce jeu comme goodies et outil de communication
- Communication interne: initiée
- Communication externe: articles + lors d'événements
- Atteinte des buts: à voir après la sortie



**BUG**

**Bug bloquant détecté**  
Il est impossible d'exécuter des tests tant qu'une nouvelle version n'a pas été livrée.

Un bug bloquant est une anomalie qui empêche totalement l'utilisation d'un logiciel ou d'une partie de celui-ci.

**Imprévu**



**1**

Les tests d'acceptation vérifient que le produit développé correspond au besoin.

Ce sont les tests « métiers »

**Campagne exploratoire de tests acceptation**



**100**

Les tests unitaires vérifient le fonctionnement d'une partie spécifique du logiciel, au niveau du code.

Ce sont les tests des « développeurs »

**Exécution de 100 tests unitaires**



**25**

Les tests de performances mesurent le temps de réponse, la consommation des ressources et la capacité du logiciel.

Ces tests nécessitent généralement des outils spécifiques.

**Exécution de 25 tests de performances**



**Projet prioritaire**

Immunise contre l'imprévu « Freeze »

Un projet prioritaire peut passer outre une politique de freeze.

**Bonne pratique**



**200**

Les tests unitaires vérifient le fonctionnement d'une partie spécifique du logiciel, au niveau du code.

Ce sont les tests des « développeurs »

**Exécution de 200 tests unitaires**



**Tests vitaux**

Protège à 50% contre l'imprévu « Bug bloquant »

Une campagne de tests vitaux permet de repérer les regressions critiques sur un logiciel.

**Bonne pratique**




**100**

Les tests fonctionnels vérifient que le comportement de l'application soit bien celui attendu.

D'après la norme ISO-25110, les tests fonctionnels sont des tests d'exactitude, de complétude et d'aptitude à l'usage.

**Exécution de 100 tests fonctionnels**



**1001 Tests**

**sogeti**  
Part of Capgemini

# 3.

Les règles et apports

# Les règles: objectifs de fun et de formation

## Règles (1/2)

Le but du jeu est de finir sa campagne de test aussi vite que possible, tout en assurant la qualité du produit.

### 2 à 5 joueurs

#### Mise en place :

Au début de la partie, mélanger le jeu et distribuer 5 cartes à tous les joueurs.

Les cartes restantes forment la pioche.

Choisir qui commence la partie.

S'il n'y plus de carte dans la pioche, mélanger les cartes de la défausse et constituer une nouvelle pioche avec.

#### Fin de partie :

La pose de la carte « test d'acceptation » met fin à la campagne et la partie.

Calculer les points (voir carte « Points ») et le gagnant est le joueur qui a obtenu le plus de points.



## Règles (2/2)

Une campagne n'est pas un long fleuve tranquille : vous devrez faire face à de multiples imprévus, comme l'apparition de bugs bloquants et vous devrez user des bonnes pratiques pour atteindre votre objectif.

### Pour arriver au bout de votre campagne, il vous faut respecter les prérequis suivants :

1. Avoir exécuté au minimum :
  - 400 tests unitaires
  - 200 tests fonctionnels
  - 50 tests de performances
2. Atteindre 1000 tests au minimum sans « test d'acceptation ». Ce prérequis ne s'applique pas pour le joueur ayant joué la bonne pratique « IA : sélection des tests »
3. Lorsque que les prérequis 1 et 2 sont atteints, jouer la carte « test d'acceptation » et la partie se termine

**1 type de test n'est pas équivalent à 1 autre type de test !**

## Points

### La qualité du travail fourni par l'équipe se mesure en points:

- 1 point par test jusqu'à 1001 tests
- Au-delà de 1001 tests, -1 point par test (ne pas trop tester)
- 50 points par bonne pratique
- 10 points par « imprévu » résolu
- 50 points par catégorie de tests effectués (fonctionnels, unitaires, performances, sécurité, adaptabilité, utilisabilité et acceptation)
- 100 points par prérequis atteint (maximum 300 points possible, voir carte « Règles (2/2) »)

### Analogie entre points et qualité :

- **0 - 1200 points** : Risque élevé d'anomalie
- **1200 - 1500 points** : Risque modéré d'anomalie
- **1500 points** : Confiance suffisante pour une mise en production
- **De 1500 - 2000 points** : Confiance accrue pour une mise en production
- **2000+ points** : Confiance optimale pour une mise en production



# Les règles: objectifs de fun et de formation

## Variantes (1/2)

### Gestion des bonnes pratiques

Au début de la partie, distribuer soit une partie soit toutes les cartes « bonne pratique » à quelques-uns ou l'ensemble des joueurs. Cette distribution peut être aléatoire ou au choix des joueurs.

### Réutilisation

Possibilité de piocher la première carte de la défausse plutôt que la première carte de la pioche.

### Tests unitaires first

Il doit toujours y avoir plus de tests unitaires que de tests fonctionnels sur son plateau.

### Produit critique

Seule une confiance maximale « rien ne peut nous arriver » (2000+ points) est acceptée pour mettre fin à la partie.

## Variantes (2/2)

### Back & Front

Mode de jeu en « 2 vs 2 » en 2 phases.

#### Phase 1 :

- Chaque joueur de l'équipe doit exécuter une campagne.
- Un joueur peut intervenir chez son coéquipier uniquement pour jouer des cartes « action corrective ».
- Quand tous les joueurs de l'équipe ont fini leur campagne, l'équipe passe en phase 2.

#### Phase 2 :

- L'équipe commence une campagne en coopératif.
- Les bonnes pratiques de l'ensemble des joueurs de l'équipe sont mises en commun.



## Itératif (1/2)

### Il peut être joué en mode « Classique » ou « Coopératif »

Ce mode est un enchaînement de 4 campagnes sur un produit qui devient de plus en plus complexe (incrément). Le nombre de test pour respecter le prérequis 1 (voir « carte « Règles (2/2) ») s'incrémente lui aussi (voir carte « Itératif (2/2) »).

Le gagnant est le joueur avec le plus de points.



### Règles particulières :

- A la fin d'une itération, chaque joueur conserve, sur son plateau, les cartes « bonne pratique » qu'il a joué et toutes les autres cartes sont mélangées. Une nouvelle itération peut ainsi commencer.
- Les points des joueurs sont calculés à la fin de chaque itération et additionnés aux scores précédents.

## Itératif (2/2)

Minimum de tests exécutés pour respecter les prérequis 1 et 2 (voir carte « Règles (2/2) ») et nombre de tours maximum pour le mode « Coopératif » (« Facile » / « Normal » / « Difficile ») :

#### Itération 1 :

- 200 tests unitaires - 100 tests fonctionnels
- Atteindre 400 tests au minimum
- Moins de 13/10/8 tours

#### Itération 2 :

- 300 tests unitaires - 150 tests fonctionnels
- 25 tests de performances
- Atteindre 600 tests au minimum
- Moins de 16/13/10 tours

#### Itération 3 :

- 400 tests unitaires - 175 tests fonctionnels
- 25 tests de performances
- Atteindre 800 tests au minimum
- Moins de 19/16/13 tours

#### Itération 4 :

- Prérequis par défaut du mode « classique »
- Moins de 23/19/16 tours

# Les règles: les types de cartes

## SPECS



Modification des spécifications  
Impossible d'exécuter des tests fonctionnels.

Les spécifications fonctionnelles décrivent le comportement du logiciel. Une modification force à revoir ses développements et tests

Imprévu

## Spécifications validées



Supprime l'imprévu « Modification des spécifications »

Des spécifications validées permettent au testeur de concevoir ses tests en connaissant le comportement attendu

Correctif

## Behaviour Driven Development



Immunité contre l'imprévu « Modification des spécifications »

Le BDD est une méthodologie permettant au métier, aux développeurs et aux testeurs de se synchroniser sur un besoin

Bonne pratique

# 25

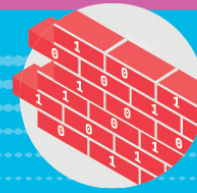


Les tests fonctionnels vérifient que le comportement de l'application soit bien celui attendu

D'après la norme ISO-25010, les tests fonctionnels sont des tests d'exactitude, de complétude et d'aptitude à l'usage.

Exécution de 25 tests fonctionnels

# 200



Les tests unitaires vérifient le fonctionnement d'une partie spécifique du logiciel, au niveau du code

Ce sont les tests des « développeurs »

Exécution de 200 tests unitaires

## IA: sélection des tests



Atteindre le nombre minimum pour chaque type de test permet de jouer la carte « test d'acceptation »

Une sélection des tests permet de diminuer le nombre total de tests à exécuter. L'IA peut aider à dans ce choix

Bonne pratique

# 25



Les tests unitaires vérifient le fonctionnement d'une partie spécifique du logiciel, au niveau du code

Ce sont les tests des « développeurs »

Exécution de 25 tests d'adaptabilité

# 25



Les tests d'utilisabilité vérifient la facilité de prise en main de l'application, l'ergonomie, l'accessibilité à des personnes ayant des déficiences visuelles ou auditives

Ces tests sont particulièrement importants dans le cadre de la RGAA.

Exécution de 25 tests d'utilisabilité

# 25



Les tests de sécurité vérifient la capacité à protéger les données, gérer les droits et assurer les authentifications

Ces tests sont particulièrement importants dans le cadre de la RGPD

Exécution de 25 tests de sécurité

# 1



Les tests d'acceptation vérifient que le produit développé correspond au besoin

Ce sont les tests « métiers »

Campagne exploratoire de tests acceptation

# 25



Les tests de performances mesurent le temps de réponse, la consommation des ressources et la capacité du logiciel

Ces tests techniques nécessitent généralement des outils spécifiques

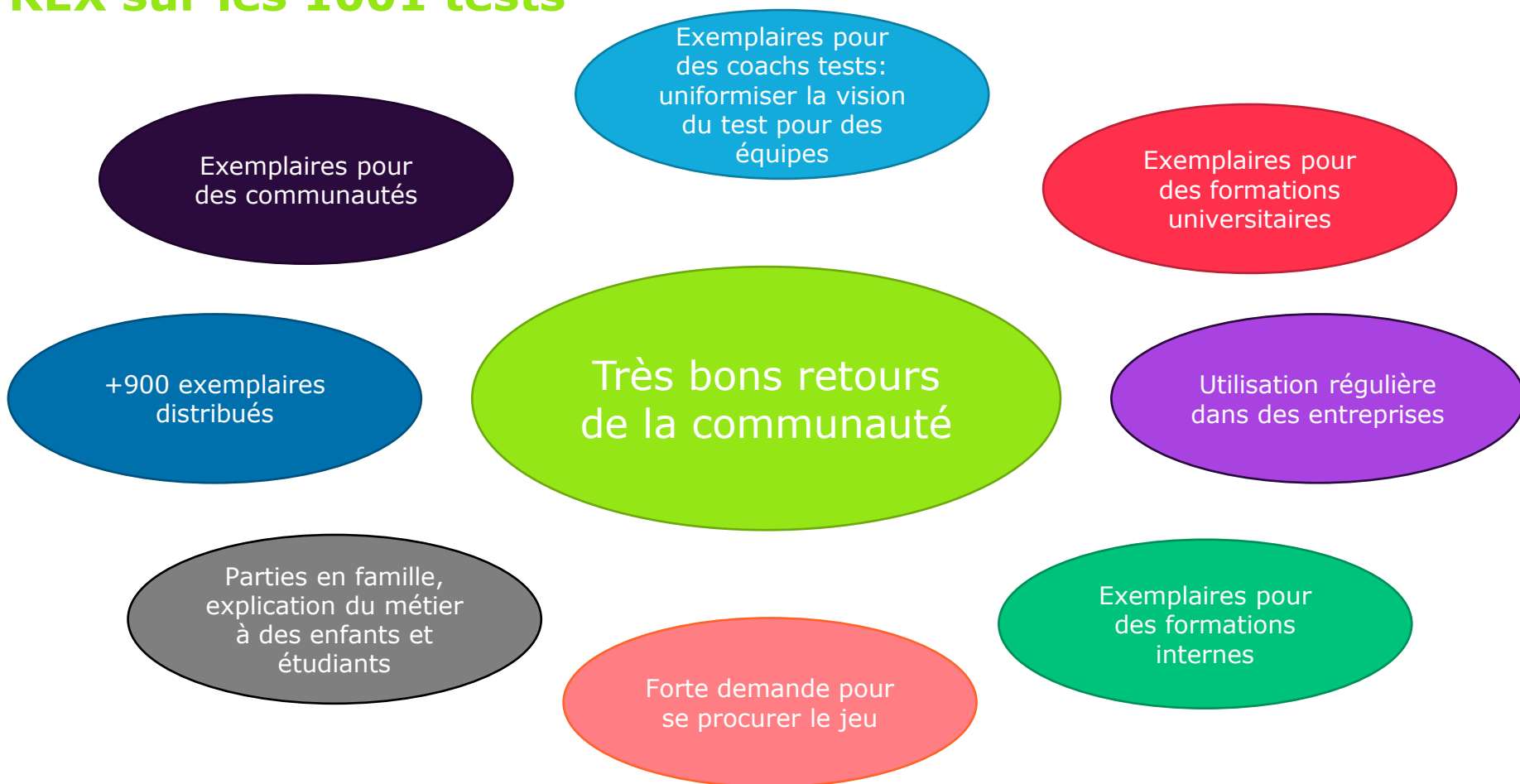
Exécution de 25 tests de performances

# 4.

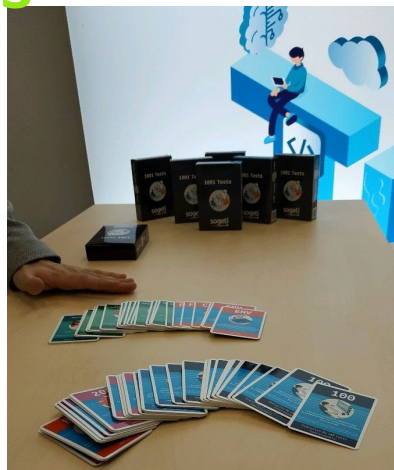
## REX les 1001 tests



# REX sur les 1001 tests



# REX sur les 1001 tests



**6.**

On continue!

# On continue!

## Sur le fond

- **Traduction en anglais**
- Déjà plusieurs demandes (Amadeus, L'Oréal, interne Cap, LVMH...)
- Mise en avant en interne
- Présentations
- Formations
- Welcome pack ?
- Communication interne groupe
- Outil de communication de Sogeti
- Evènements public
- Direct client
- Une extension à venir ?

## Sur la forme

- Imaginer d'autres moyens de diffusion
  - Commande en ligne ?
  - Virtualisation (sur tablettopia)
- Quelques améliorations ergonomiques
  - Boite
  - Identifier les « régressions »...
- Mise en place lors d'événements (ex: QX Day, événements clients)
- Mise en place d'ateliers en clientèle et en interne
- Animer des communautés / Ice breaker

## Autres

- Imaginer d'autres jeux ?
  - Prototype jeu TMAP
- Accompagner des initiatives moins proches du test (ex: craftsmanship) ?

### Behaviour Driven Development



« Immunise contre l'imprévu » Modification des spécifications »

Le BDD est une méthodologie permettant au métier, aux développeurs et aux testeurs, de se synchroniser sur un besoin.

Bonne pratique

# 25



Les tests unitaires vérifient le fonctionnement d'une partie spécifique du logiciel, au niveau du code.

Ce sont les tests des « développeurs »

Exécution de 25 tests d'adaptabilité

# BUG



**Bug bloquant détecté**

Il est impossible d'exécuter des tests tant qu'une nouvelle version n'a pas été livrée.

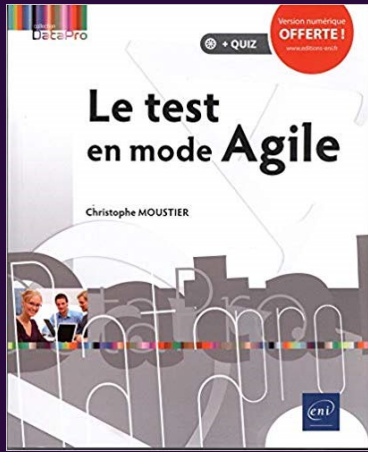
Un bug bloquant est une anomalie qui empêche totalement l'utilisation d'un logiciel ou d'une partie de celui-ci.

Inprévu

# Questions ?

## MERCI de votre attention

### Un peu de lecture ?



24



Security Classification



Édité par le Comité Français des Tests Logiciels