caTesTdrale

Table of Contents

Atelier de sensibilisation et de réflexion sur l'apport des Test Unitaires Automatisés

Pitch

- Venez participer à un atelier agile et ludique pour faire des tests unitaires sans toucher à du code!
- Vous êtes développeur: venez sans votre pc et repartez avec une pratique vitale pour vous!
- Vous êtes PO/MOA: venez comprendre pourquoi vos devs vous réclament du temps pour leurs tests
- Vous êtes manager: Venez comprendre pourquoi il faut aider vos équipe à faires des TUA. Pour eux, pour vous, pour l'entreprise!
- Vous êtes curieux: venez prendre quelques bonnes idées et arguments

Objectif

L'atelier vise à sensibiliser sur l'intérêt d'investir dans la mise en place de tests unitaires automatisés.

Il peut-être utilisé, de manière plus général, pour montrer l'importance d'investir dans de la qualité.

Préparation et déroulement

Public: Tout public (Devs, MOA/PO, Managers...). Pas de code on vous dit!

Durée: 30 mn

La durée peut variée en fonction de la manière de dérouler l'atelier et du temps consacré au debriefing.

Pour l'animateur

Salle

Des tables pour construire les châteaux. Il faut qu'elles soient stables et de préférence, qu'elles ne soient pas en contact des tables d'une autre équipe.

Prévoir 2 à 3 personnes par table.

Préparation avant l'atelier

Imprimer la page ChateauCarte.pdf sans mise à l'échelle et sans toucher aux marges. Vérifier que les "fentes" (traits en noir) fassent bien la largeur d'une carte.

Faire autant de copies que nécessaire (cf. "Nombre de carte et de harnais nécessaire)

NOTE	Pour gagner du temps, on peut inciser les fentes avec un cutter au préalable.

Matériel pour l'atelier

- Harnais précoupés ou pas (ChateauCarte.pdf)
- Paires de ciseaux (une pour 2 groupes)
- Cartes à jouer (cf. "Nombre de carte et de harnais nécessaire), si possible usagées

Le tableau ci dessous indique le nombre de cartes nécessaires en fonction de la hauteur du château. Les équipes devraient arriver à faire des châteaux de 5 à 6 étages.

h = Hauteur du château de cartes

Table 1. Nombre de carte et de harnais nécessaire

h	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Nb cartes	7	15	26	40	57	77	100	126	155	187	222
Nb paquets de cartes (54)	1	1	1	1	2	2	2	3	3	4	5
Nb Harnais	3	6	10	15	21	28	36	45	55	66	78
Nb feuilles de harnais	1	1	1	2	2	2	3	4	4	5	6

Nb cartes = $(3\ h^2+h\ )/2$

Nb harnais=$(\ h^2+h\ )/2$

Déroulement de l'atelier

Discours d'introduction

Le développement informatique est un assemblage de code écrit, d'assemblage de librairie et de communication avec d'autres application tout en utilisant diverses techologies. La moindre erreur lors de la construction de cet ensemblage complexe et c'est le bug. Les applications sont des immenses châteaux de cartes et c'est ce que l'on se propose de construire.

Pour cela, on va expérimenter deux approches. La première sera la construction classique d'un château de carte. La seconde propose de mettre en place des tests unitaires.

L'objectif sera de former le plus haut château de carte. Un étage est comptabilisé s'il est complet.

Mis en place

Montrer ce que l'on appelle un test (un harnais) et faire une démo de sa mis en oeuvre.

On peut opter pour deux formats différents:

1. Séparer les équipes avec certains faisant des tests et d'autres pas. La séparation peut être selon le volontariat où e fonction des tables où ce sont placés les gens.
2. Dérouler l'exercice pour tout le monde sans test puis rejouer l'exercice avec les tests. Il est intéressant de noter la hauteur obtenue à chaque itération pour pouvoir comparer les résultats entre les deux approches.

Distribuer le matériel si ce n'est pas déjà fait et déclencher le chronomètre pour une itération d'1 minute.

A la fin de l'itération, faire un rapide tour de l'avancement de chaque équipe.

Faire au moins 5 itérations. Il est probable que les équipes sans tests jettent l'éponge avant cela.

On peut continuer "pour le fun" en proposant aux équipes sans test de commencer à les mettre en oeuvre ou on peut les faire rejoindre une équipe faisant des tests. La compétition continue alors entre les équipes.

Pour appuyer la prise de consiscience des avantages des tests automatisés, on peut proposer de faire un refactoring sur notre château:

1. retourner les cartes formant les plateaux (aucune modification des tests nécessaire).
2. enlever les cartes représentant des "têtes" (légère modification des tests)
3. changer les cartes avec des formats plus ou moins large (nécessité de refaire les tests).

Debriefing

A l'issue de l'atelier, il est bien de prendre le temps de faire un débriefing.

On peut commencer par demander la manière dont les personnes ont vécues l'atelier et faire le parallèle avec ce qu'ils vivent dans leur travail.

L'idée générale qui se dégage est la notion d'investissement que représentent les tests. Voici quelques éléments sur lesquels on peut discuter:

- Sans tests, on va (beaucoup) plus vite au début mais cela ne dure pas. On peut faire le lien avec la courbe décrite en extreme programming et qui cherche à avoir un coût des évolutions constant dans le temps là où il est généralement exponentiel.
- Quel risque prend t-on sur l'avenir du code à développer sans test ?
- L'absence de tests donne un sentiment de challenge. Il est nécessaire d'être extrêment concentré pour mettre au point une solution. Avec les tests le travail est plus simple et plus rébarbatif. Cela peut donner l'impression d'avancer moins vite mais le progrès est constant.

- Imaginons que l'on nous fournisse un château de cartes de 3 étages déjà fait et sur lequel on nous demande d'ajouter un étage. Préféreriez vous partir d'un château de 3 étages sans tests (partir d'un code sans tests) ou uniquement des tests coupés/pliés (les tests sans le code) ? Qu'est ce qui a le plus de valeur ?
- En cas de destruction du château (restructuration), combien de temps cela prend il à le reconstruire avec/sans tests ?

Ressources

- Version pdf des instructions
- Harnais PDF ou XLSX