

Le Continuous Testing, Attentes Versus Réalités



Soukaïna Alaoui



Julien Pessarossi-Langlois



Co-auteur Maxime Bardou

Sommaire

INTRODUCTION

- 1. Définition et contexte de mise en place**
- 2. Méthodes de test et outils employés**
- 3. Mesure de la Qualité**

CONCLUSION

INTRODUCTION

Le test est souvent

- Délaissé
- Relégué en bout de chaîne, notamment en organisation Agile
- Souvent mis en place suite à une « mauvaise expérience »



Conséquences :

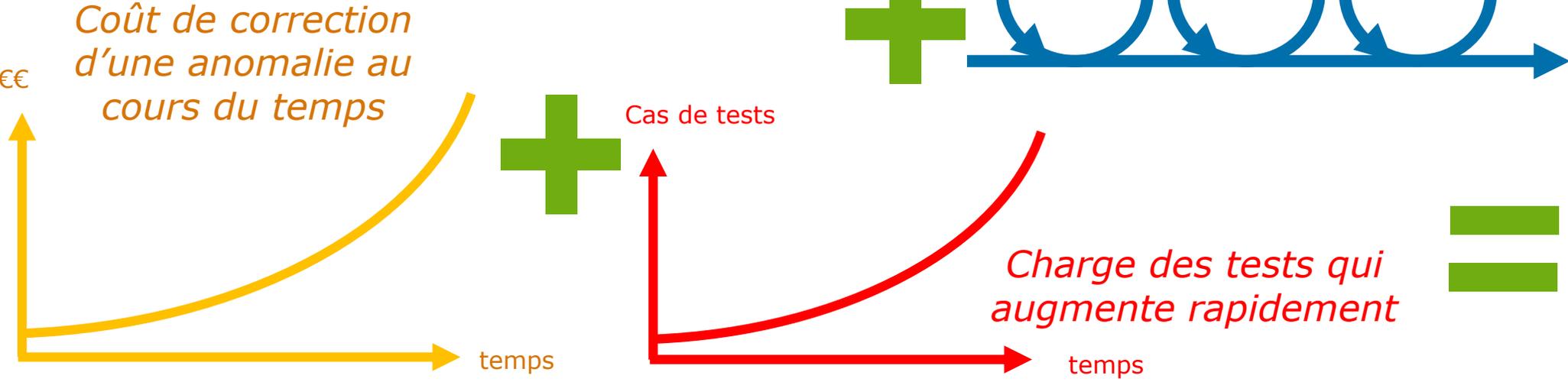
Lors de la mise en place du test, le testeur doit

- Rattraper « son retard »
- Adresser en même temps un Patrimoine de Non Régression et les Nouvelles Fonctionnalités



Pourquoi parler de Continuous Testing?

Sprints agiles qui s'enchainent continuellement



Trouver une organisation qui permette de suivre la cadence !



Le continuous testing est la réponse logique aux composantes suivantes :

- Le coût d'une anomalie augmente avec le temps
- L'Agilité entraine des modifications perpétuelles du système développé



DEFINITION ET CONTEXTE DE MISE EN PLACE

- Notre définition
- Contexte de mise en place



Notre définition

Organisation du test dans un projet Agile pour adresser le bon niveau de tests aux bons moments afin d'avoir une vision claire de la santé du système et réduire l'impact des anomalies en les découvrant au plus tôt, et ainsi contribuer au mieux à la réduction du Time to Market

Une définition basée sur notre expérience, qui sera challengée par nos projets



Contexte de mise en place

Principalement les organisations Agile

- Nous partirons d'une situation « idéale »
- Description pas à pas
- Nous y confronterons nos expériences projets
- Mise en avant des avantages du Continuous Testing



La réalité des projets rattrape toujours la théorie et nécessite de revoir nos positions !

Méthodes de test et outils employés

- Création et gestion du Patrimoine
- La Nécessité des tests automatisés



Création et gestion du Patrimoine



Création et Gestion du Patrimoine

Nous préconisons l'utilisation du Model Based Testing

- Model Based Testing : une technique qui utilise les modèles comme source de patrimoine de test
- Ses principaux avantages : faciliter ...
 - La communication en gagnant en clarté
les modèles sont utilisés comme base de travail pour discuter du système et comprendre les règles de gestion
 - La validation de la couverture
identifier visuellement les éléments manquants, les non-sens fonctionnels, etc...
- Gains
 - Une modélisation d'un système permet de générer différents niveaux de test et différentes couvertures (Fonctionnels, métiers, end to end (E2E))
 - Facilite la mise à jour des cas de test
 - Facilite les process d'analyse de risques et d'estimation de l'effort de test

Le juste effort de test, pour la juste qualité !



Sans MBT vs Avec MBT

Patrimoine de test sans MBT

Couverture réelle : **Inconnue**

Validation : **nécessite une lecture de l'ensemble par le PO**



Le testeur qui sait pertinemment que son travail ne sera pas relu pour validation



Le PO qui n'a absolument pas le temps de parcourir l'ensemble des cas de test

Sans MBT vs Avec MBT

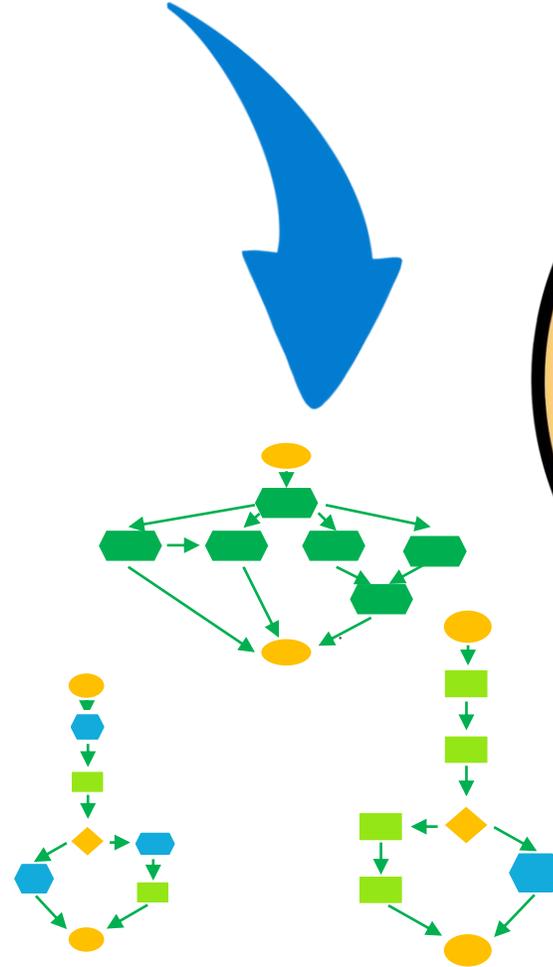
Patrimoine de test avec MBT

Couverture réelle : **calculée par les outils en fonction de la couverture choisie**

Validation : **nécessite une validation visuelle des modèles**



Le testeur content d'avoir une validation qui évite des erreurs lors de la phase d'exécution



Le PO qui est sûr que l'ensemble du système sera couvert

Une création optimisée, en deux temps

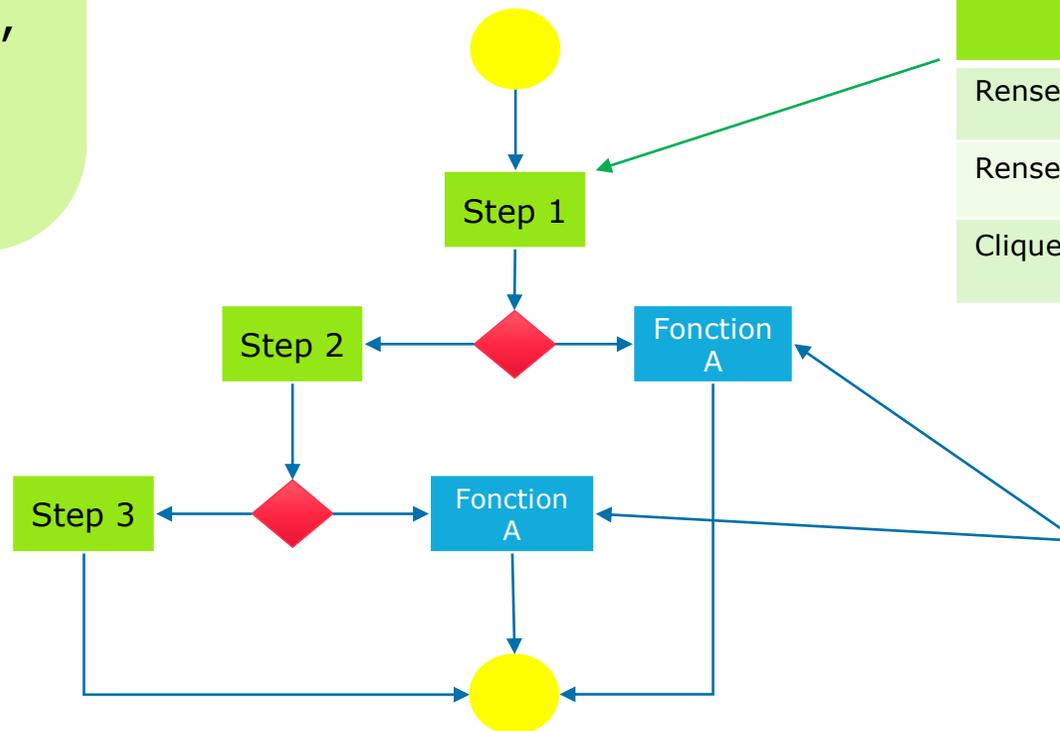
Conception de test :

- Partie la plus chronophage = rédaction du détail des cas de test
- Si un cas de test est jugé inadéquat, autant de temps perdu

Avec le MBT, on a une validation des règles de gestion via le modèle

Cette validation a lieu avant la rédaction du détail des steps

Description	Résultat attendu
Renseigner ID valide	ID renseigné
Renseigner MDP valide	MDP renseigné
Cliquer sur Valider	Au clic, le formulaire est validé, le user est connecté



Un step ou une partie de modèle appelé dans plus cas ne sera renseigné qu'une fois, l'outil se charge de concaténer les infos



Exemple de problématiques

Gestion du Patrimoine de test

- Comment créer différents niveaux de test ?
- Comment assurer la mise à jour de tous les cas de test ?
- Comment faire le lien entre documentation fonctionnelle et les tests ?
- Comment mesurer la couverture de test ?
- Comment gérer la gestion multi-projet ?



Le PO écrit les règles de gestion et spécifications et crée le Back log



Chaque équipe doit écrire les cas de test selon son besoin et niveau de test



Ecriture des cas de test fonctionnels



Ecriture des cas de test métiers



Ecriture des cas de E2E

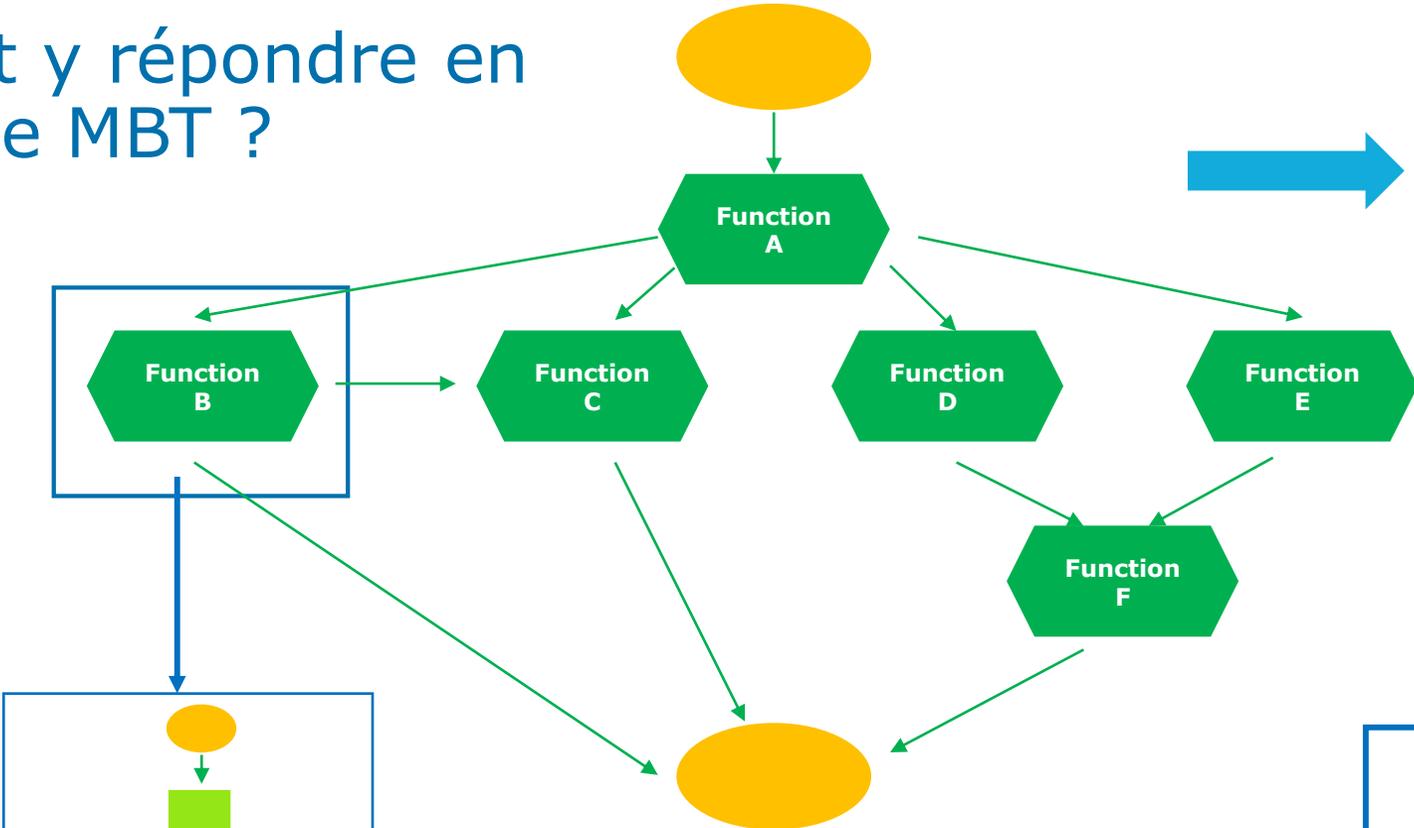


Scripting des cas de test fonctionnels

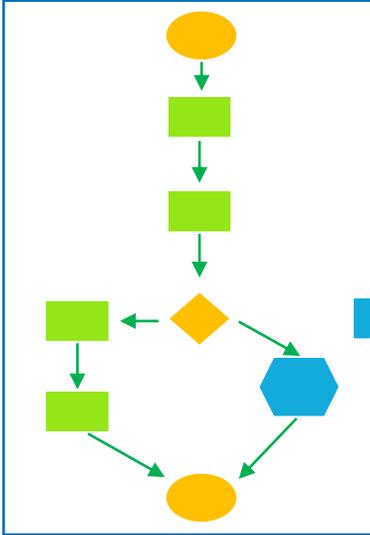


Scripting Des cas de test E2E

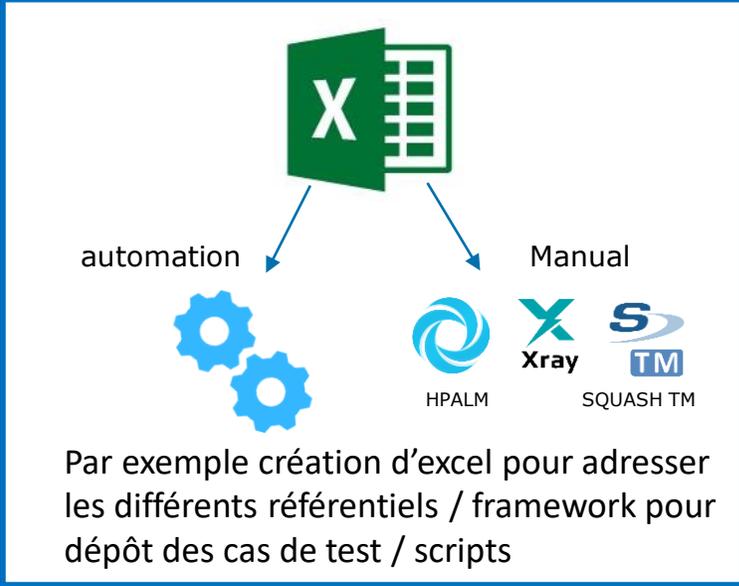
Comment y répondre en utilisant le MBT ?



- Macro model :
- Mapping de toutes les fonctionnalités
 - Génération des tests E2E
 - Manuels
 - Base script d'automatisation

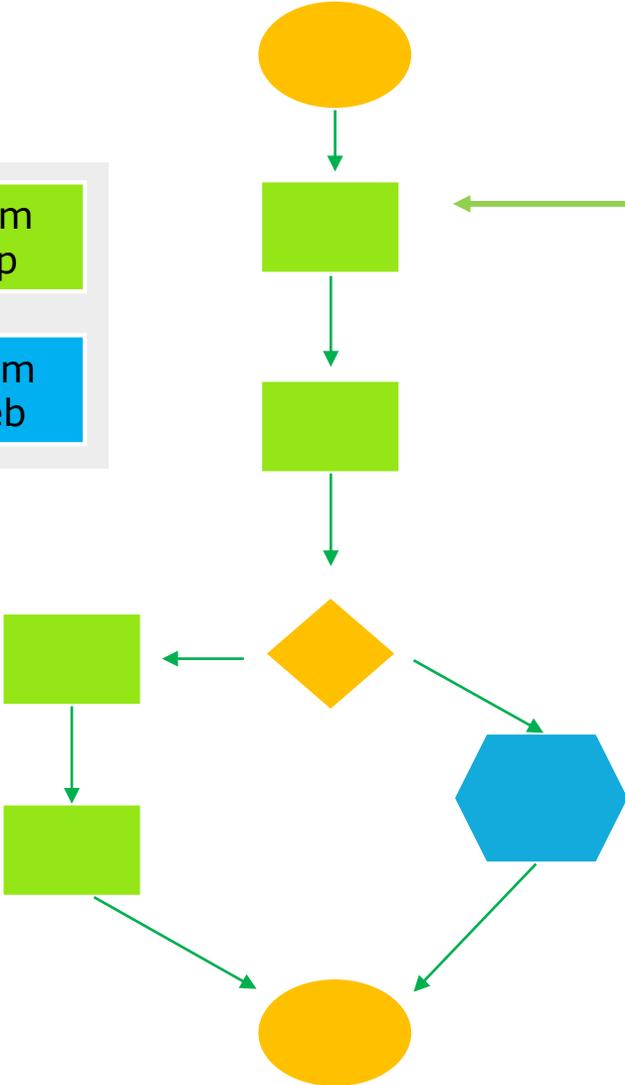


- Sous-model : Modélisation d'une fonctionnalité :
- Génération et optimisation de cas de test → cas de test Fonctionnels
 - Génération de toutes les combinaisons possibles → cas de tests Métiers
 - Génération pour
 - Tests manuels
 - Scripting d'automatisation



Comment y répondre en utilisant le MBT ?

Une utilisation du MBT qui nécessite une bonne organisation et une bonne communication !

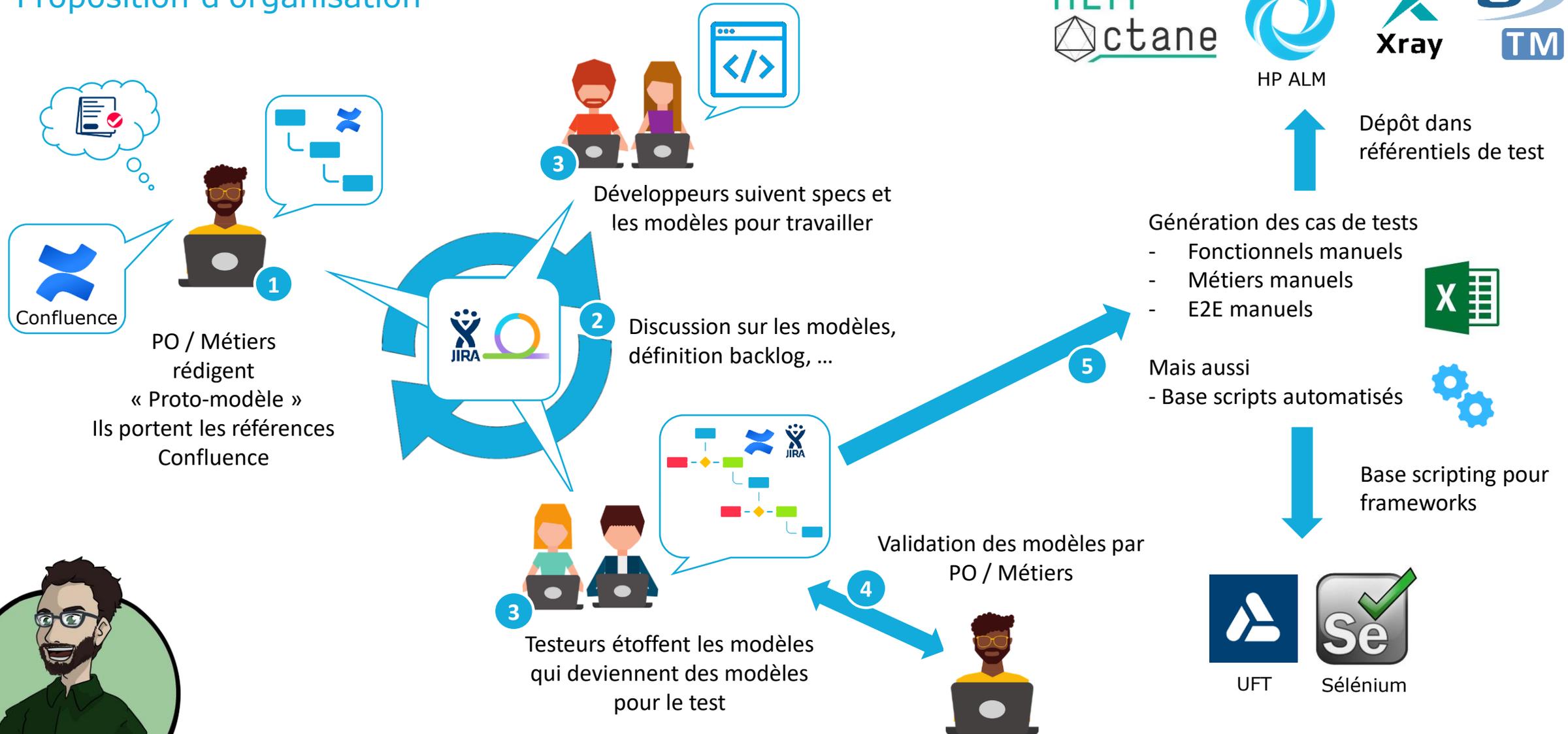


Team Application

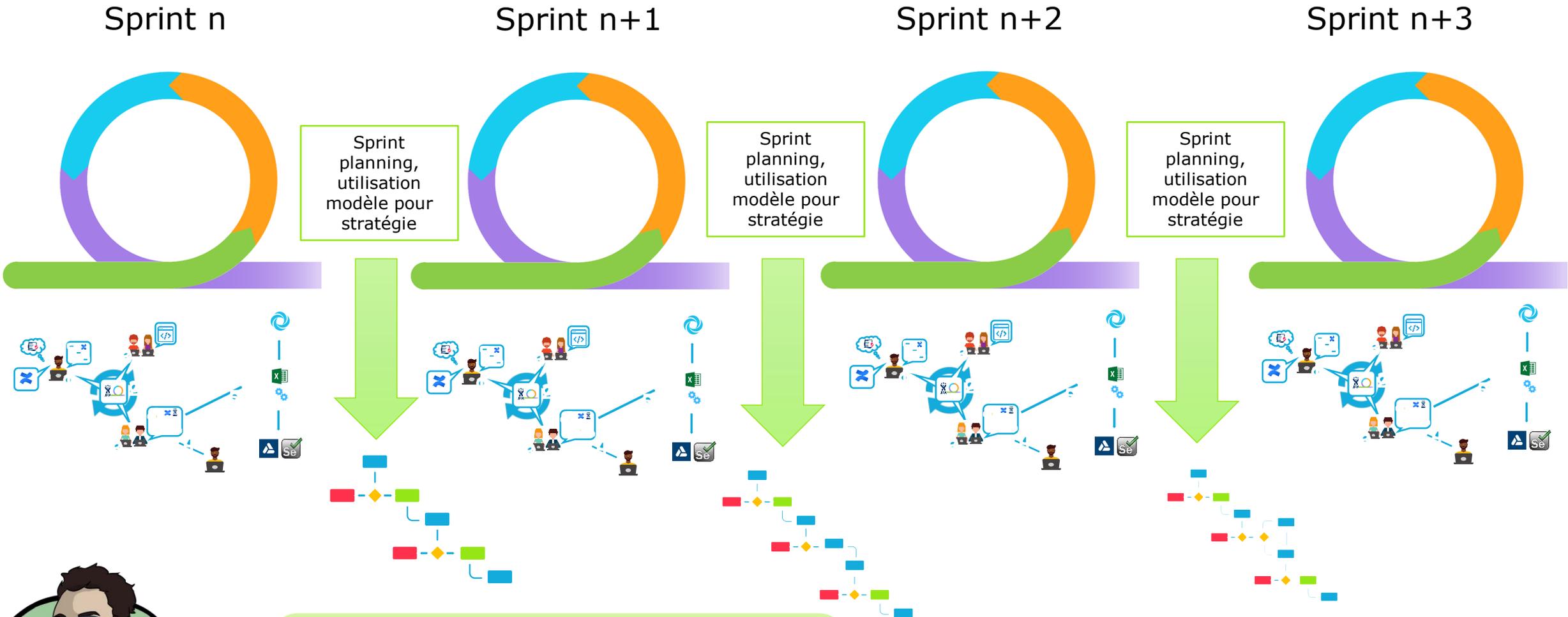
- Crée et entretien le modèle de l'application
- Quand l'application appelle une webview, le modèle appelle le sous modèle correspondant du projet web → ce modèle aura été créé par la team web. Au préalable, discussion entre les deux équipes pour comprendre les besoins de la team app (couverture, ...)
- Création et m à j du modèle web par la team web
- La plupart des outils MBT permettent de générer et stocker différents niveau de couverture suivant les besoins

Et ensuite ?

Proposition d'organisation



Le MBT dans une organisation Agile



Nos modèles permettent d'évaluer la qualité de chaque incrément puisqu'on aura une vision claire de ce qui aura été testé ! Ils permettront en plus de servir de support pour la stratégie de test du sprint suivant !

Les limites du MBT

Plus de 3 ans d'utilisation sur des projets différents par :

- Le sujet,
- La taille,
- La complexité,
- Le niveau d'implication des clients ...

L'utilisation du MBT peut se heurter à de nombreux « obstacles »

- Facile à mettre à jour,
- Tests générés très macro / peu détaillés



- Mise à jour complexe,
- Permet de générer une grande diversité de tests très complets et détaillés

Le MBT n'est pas une technique « miracle ».
Elle nécessite de bien définir en amont les règles pour son utilisation, et plusieurs bonnes pratiques !

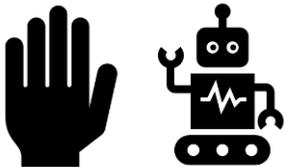


Où positionner le curseur ?



La Nécessité des tests automatisés

La nécessité des tests Automatisés

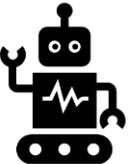


Tests fonctionnels
Prototypes
Simulations

Tests exploratoires
Tests d'utilisabilité
Tests d'acceptance



Quadrants
des test
agiles

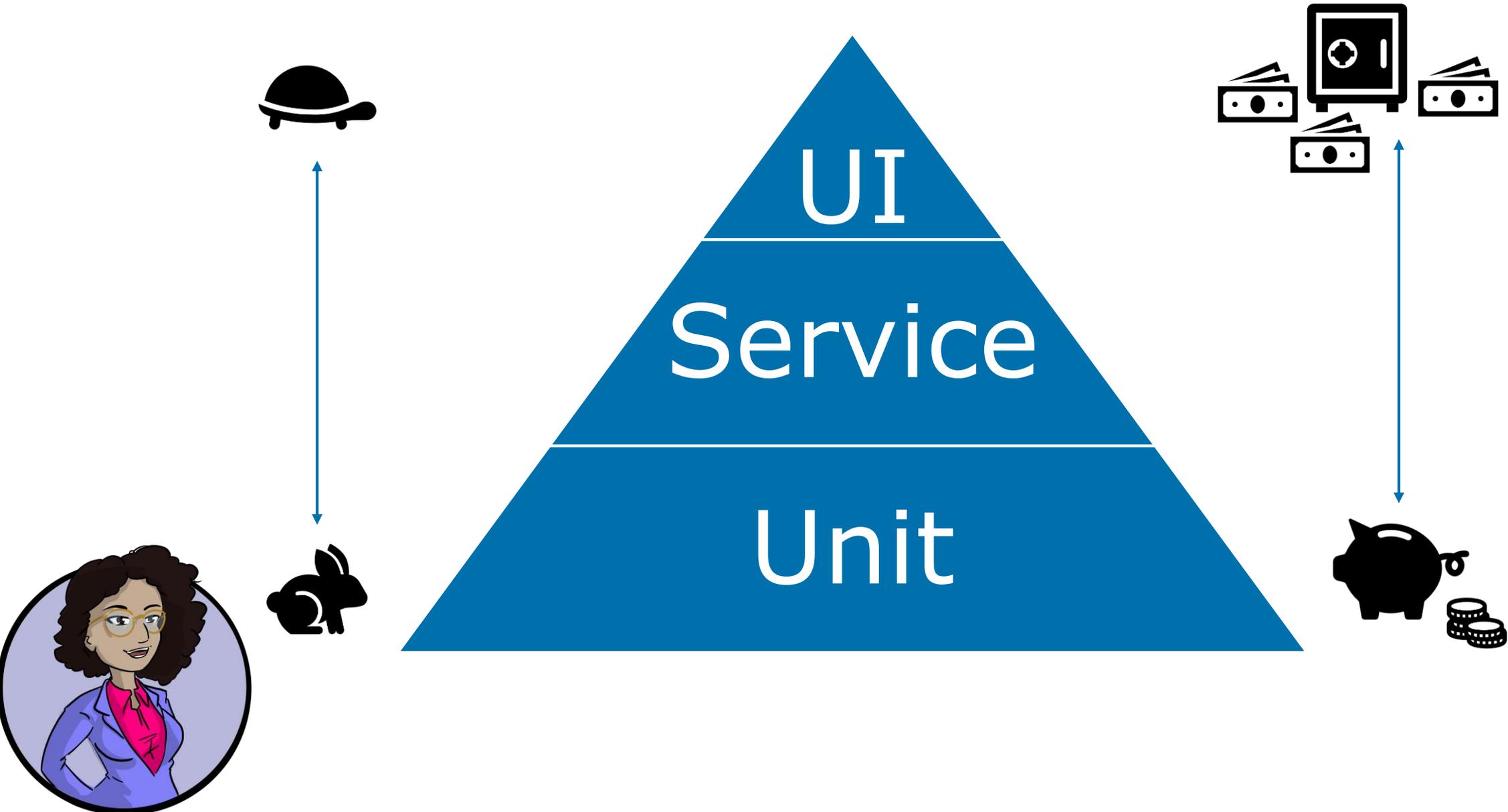


Tests unitaires
Tests de composants

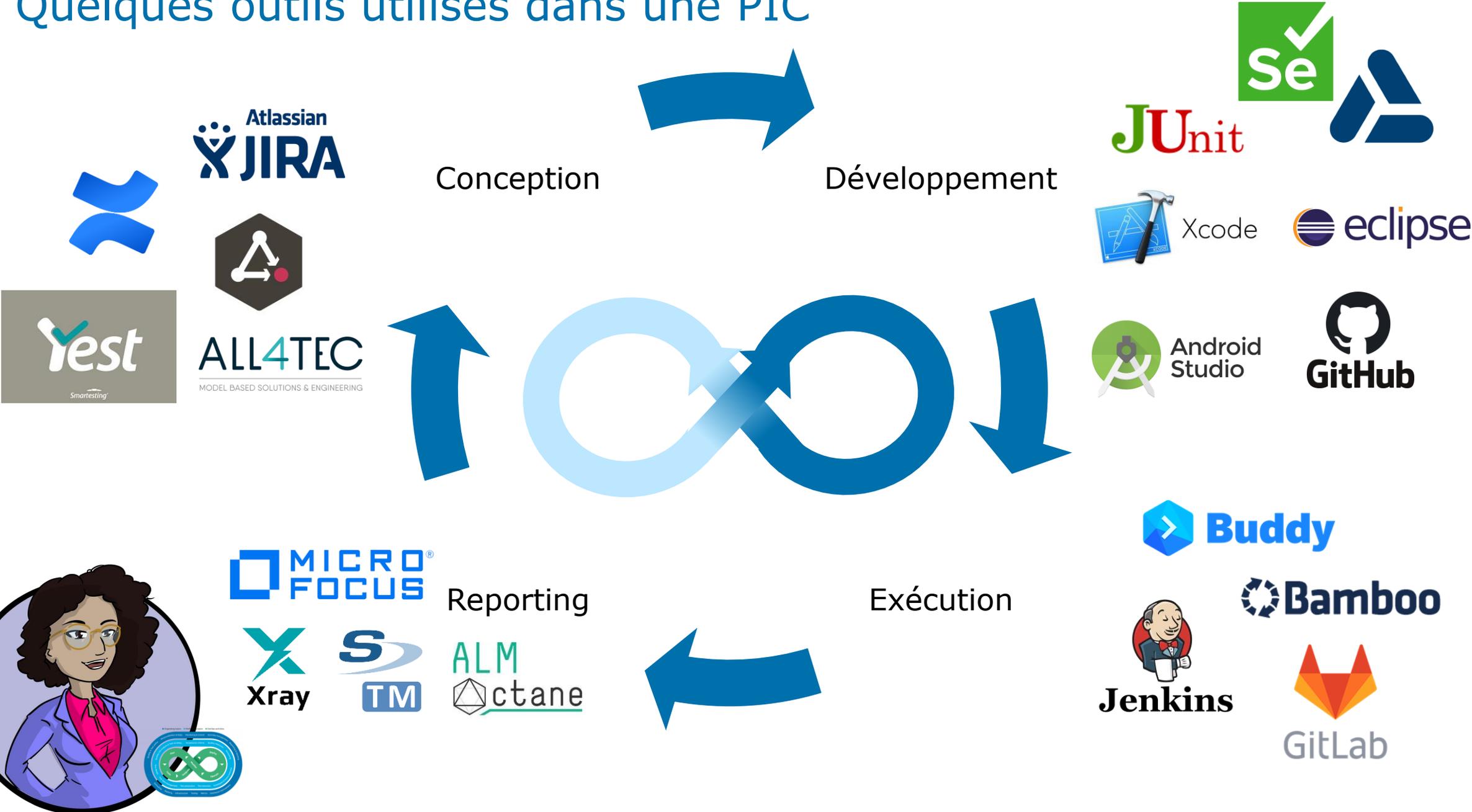
Tests de performance
Tests de charge
Test de sécurité
Tests de « ité »



La nécessité des tests Automatisés



Quelques outils utilisés dans une PIC



Challenges face à l'automatisation



Cout d'investissement
Cout de maintenance
Cloisonnement d'environnement de test
Compatibilité entres les plateformes de test et du produit à tester

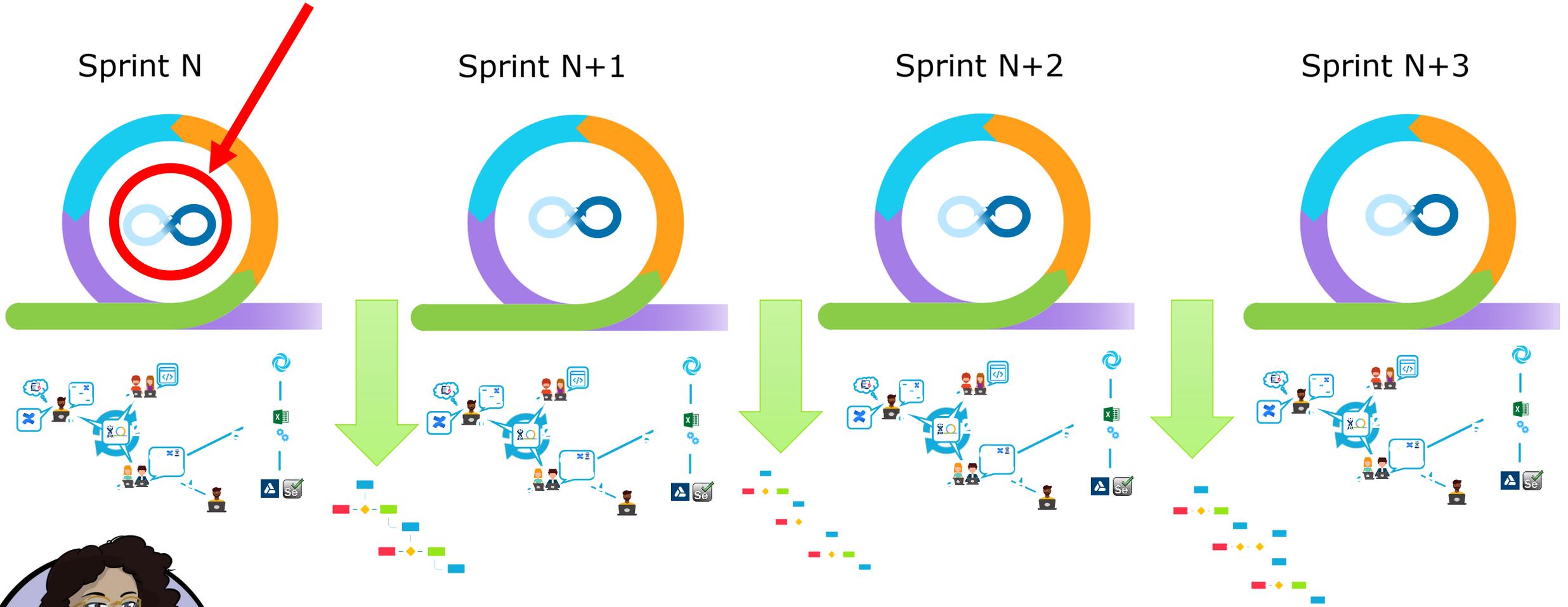
On ajoute la PIC à notre organisation

Sprint N

Sprint N+1

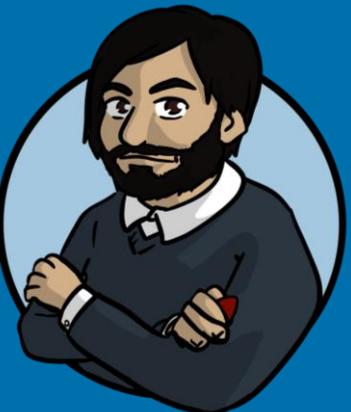
Sprint N+2

Sprint N+3



On a donc un patrimoine de test toujours à jour, grâce aux modèles, et qui facilite la sélection et l'enrichissement des cas de test automatisés pour la PIC

Mesure de la qualité



Mesure de la qualité

Afin d'assurer et mesurée la qualité, il est nécessaire de mettre en place un suivi.

Celui-ci doit permettre d'assurer,

- Une communication claire et transparente
- Le suivi du projet :
 - Avancement des tests
 - Qualité des incréments

On a tendance à faire la guerre aux KPI dans les organisations Agile, donc c'est un sujet sensible!



Il faudra s'attacher à choisir des KPIs pertinent parmi la multitude que l'on peut imaginer !

Suivi
d'avancement
de l'exécution

Couverture
des TU

Statuts des
tests

Robustesse
des scripts
autom'

Backlog
d'anomalies

Taux de
rejeu

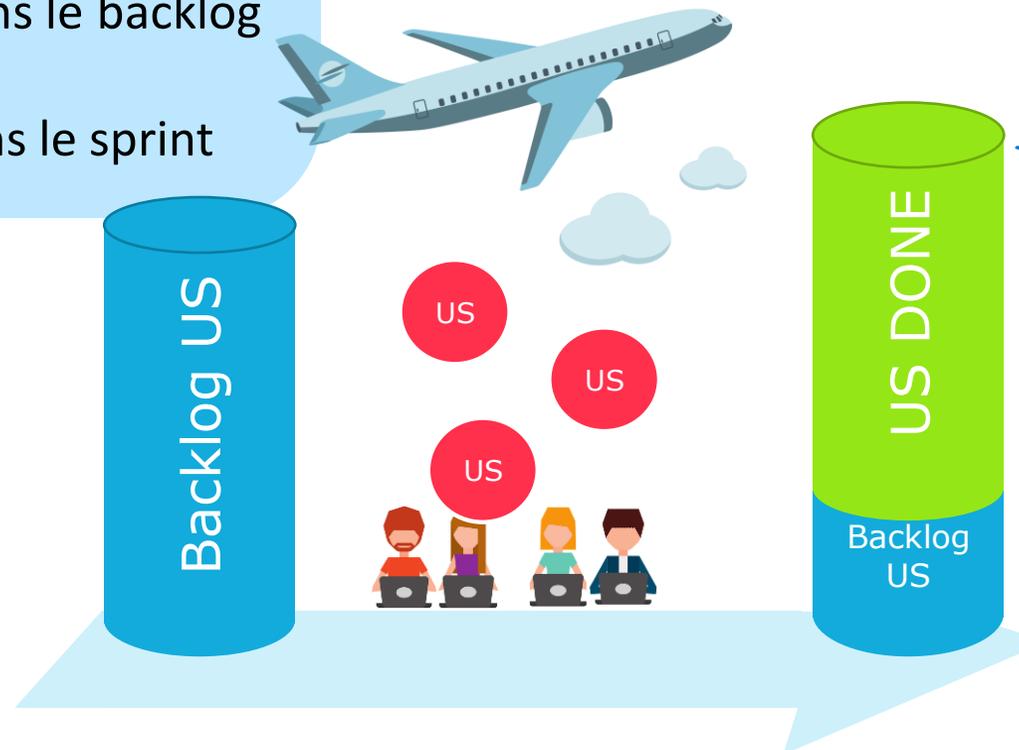
Criticité,
périmètres

...

Suivre les US – les apports

Les données liées aux US = la clef d'une bonne organisation Agile !

- Le nombre de US planifiées avant le sprint
- Le nombre d'US « done » en fin de sprint
- Le nombre d'US qui reste dans le backlog en fin de sprint
- Le nombre d'US rajoutée dans le sprint



On définit ainsi :

- La maîtrise de l'estimation des Story point
- La livraison d'un produit fonctionnel

$$= \frac{\text{US done}}{\text{Total US}}$$


L'objectif étant d'identifier les problèmes éventuels

Seulement les US ?

Si suivre les US n'est pas suffisant,
on peut chercher d'autres KPIs

- Nbr anomalies relevées,
- Nbr anomalies rejetées,
- Aléas,
- ...

En évitant le trop plein de KPIs !



« Le testeur doit toujours garder à l'esprit les éléments qui lui permettent d'être en maîtrise de sa capacité à faire en accomplissant ses tâches de test »

Pour nous le principal KPIs du testeur est la couverture de tests en rapport avec la stratégie et l'analyse risque produit

La première étape :
Faire une ARP entre
PO et Devteam, et
utiliser les outils
pour maîtriser les
besoins



Couverture de test
identifiée ?

Couverture des cas
de test automatisé ?

Juste les
derniers cas de
Julien à extraire
du modèle

Yes !

Oui !



Conception des test
?

Modèles finis, juste
besoin de générer les
cas de test, et de les
injecter dans le
référentiels de test !

Campagne de test
créée ?

Et je pourrai créer
les campagnes
dans la foulée !



Sprint N



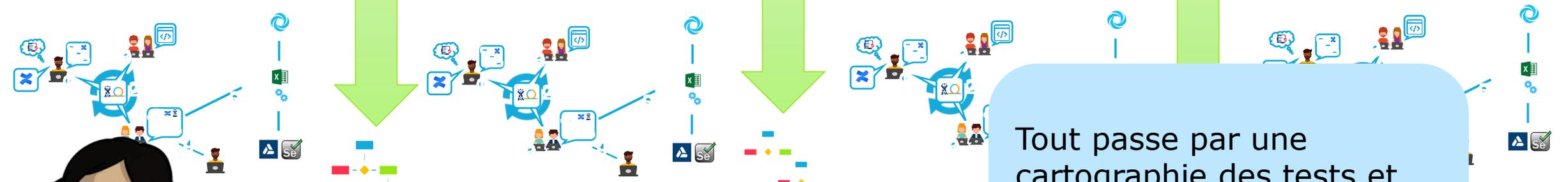
Sprint N+1



Sprint N+2



Sprint N+3



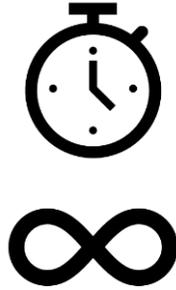
On apporte une vision claire de la qualité de l'application pour savoir quoi améliorer et aider à la prise de décision des PO

Tout passe par une cartographie des tests et une maîtrise totale de son patrimoine de test manuel et automatisé

Conclusion

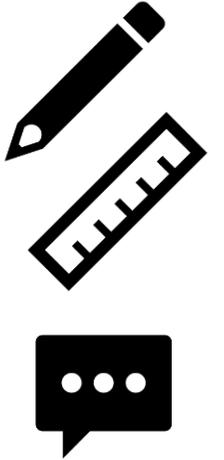
TESTER

- Au plus tôt
- Tout le temps



CONCEVOIR

- Facile à MAJ
- Facile à mesurer
- Facile à communiquer



AUTOMATISER

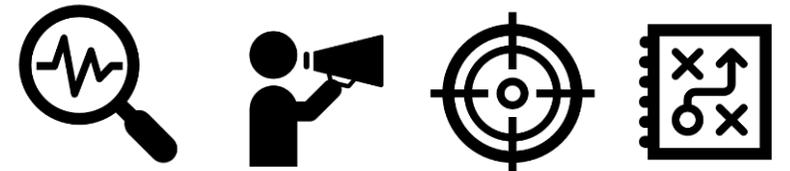
- Selon le besoin
- Pour réduire temps du feedback qualité



MESURER

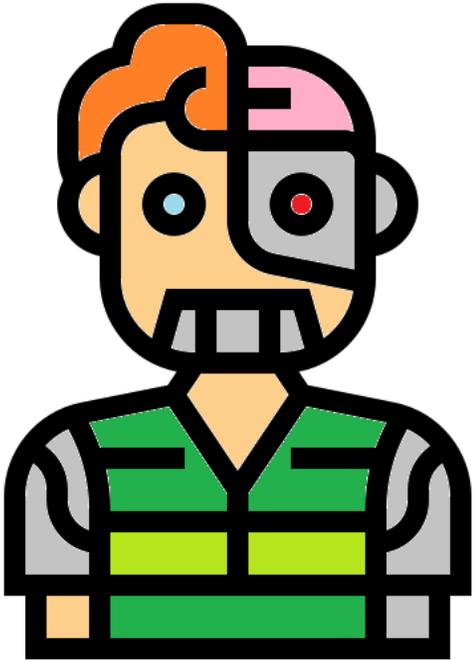
Savoir quoi mesurer :

- Pour mieux communiquer
- Pour mieux cibler l'effort
- Pour aider à la prise de décision



Conclusion

AI & QA



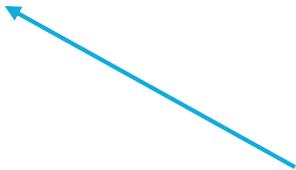
CONCEVOIR



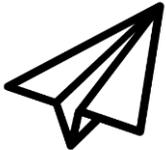
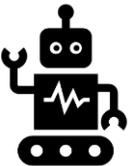
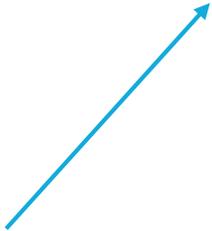
MESURER



TESTER



AUTOMATISER



Merci de votre attention

Vos questions