

FORMATION QUALITE LOGICIELLE

TESTEUR DANS UNE ÉQUIPE AGILE



Ecole de la Qualité Logicielle

Testeur dans une équipe agile

Le Test en contexte Agile	p.3
L'inversion de la pyramide des tests	p.5
La stratégie de tests en mode agile	p.6
Les Quadrants de Marick	p.13
Le Manifeste du Testeur Agile	p.15
Être testeur agile :	
Les compétences complémentaires	p.20
Les difficultés culturelles	p.28
Les risques pour le testeur	p.29
Le rôle du testeur dans l'équipe	p.30

Le Test en contexte Agile (1/2)

- **Les tests unitaires et de composants** (TU & TI – Tests d'Intégration) forment la base de l'édifice du développement Agile et permettent d'asseoir la qualité du code.
- Automatisés, nombreux, et régulièrement maintenus, ils participent à la stabilité applicative.
(« Doing the software right »)
- **Le test IHM** concerne la partie haute du test itératif : on s'assure de l'ergonomie et des performances de l'IHM. La forme de cette pyramide hérite d'un constat : dans un mode manuel, cela coûte très cher de faire du test IHM. **On visera donc l'automatisation.**

Le Test en contexte Agile (2/2)

- **Les tests fonctionnels** permettent de vérifier en permanence la conformité aux besoins exprimés par le client et implémenter au sein de l'équipe agile (« *Doing the right software* »). Mais il s'agit aussi de maintenir la qualité globale d'un édifice construit et livré de façon incrémentale, via leur utilisation en tant que TNR.

Les tests fonctionnels doivent être conçus en vue de leur automatisation.

- **Les TNR** : Automatisés, ils permettront de s'assurer en permanence de la qualité de la conformité aux besoins de l'utilisateur.

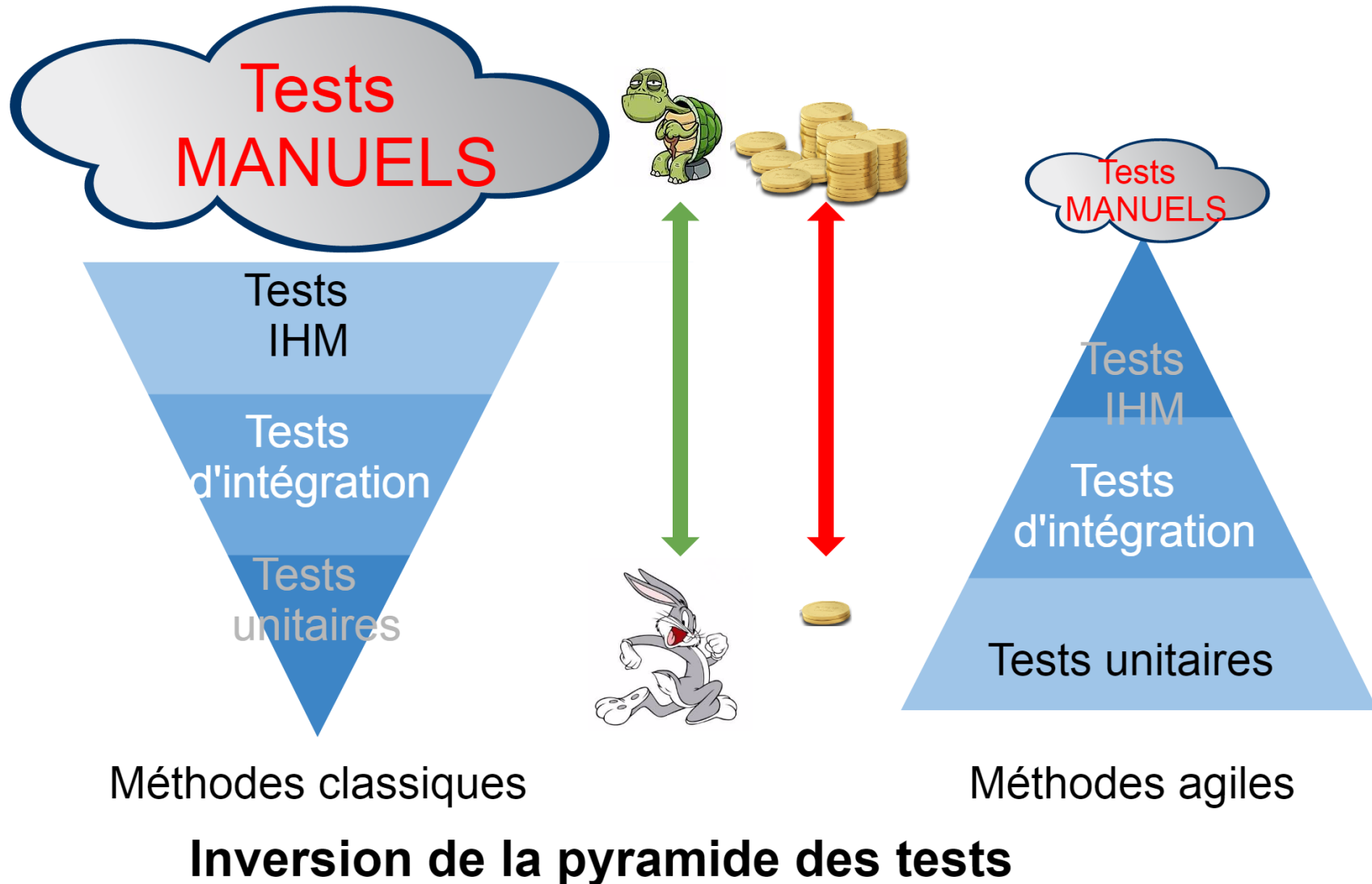
- En contexte Agile, on cherche à :

- Augmenter la **rapidité et la facilité d'exécution** des CT
(Tous les membres de l'équipe peuvent tester)

- **Diminuer le coût** des tests les « plus chers »

On automatise !

L'inversion de la pyramide des tests



La Stratégie de tests en mode Agile

- On utilise l'acronyme **PURIFF** comme concept pour regrouper l'ensemble des activités de tests à mener durant le Sprint.
- Chaque élément de l'acronyme correspond à une catégorie de tests :
 - **P** : pour les tests de Performances
 - **U** : représente les tests Unitaires
 - **R** : désigne les tests de non Régression
 - **I** : couvre les tests d'Intégration
 - **F** : pour les tests Fonctionnels
 - **F** : représente les tests (Non) Fonctionnels

La Stratégie de tests en mode Agile

PURIFF : tests de **P**erformances

- Les tests de performance consistent à **mesurer les temps de réaction de l'application, soumise à une charge d'utilisation donnée.**
- Il existe plusieurs solutions open source et commerciales pour tester les performances. Parmi les plus connus on peut citer : JMeter, HP Load Runner, IBM Rational Performance Tester, etc.
- **Rôle des testeurs :**
 - Aider les développeurs à concevoir les tests,
 - Aider à déterminer les plages à tester
 - Exécuter les tests de performance, avec ou sans l'aide des développeurs

La Stratégie de tests en mode Agile

PURIFF : tests Unitaires

- Les tests unitaires sont écrits par les développeurs pour tester les classes de façon unitaire. Ils s'inscrivent en général dans une stratégie de développement dirigé par les tests (TDD).
- Il existe plusieurs solutions que l'équipe Scrum peut utiliser pour écrire et exécuter les tests unitaires : JUnit, NUnit en sont des exemples.
- **Rôle des testeurs :**
 - Aider les développeurs à exécuter les tests unitaires
 - Automatiser les tests unitaires

La Stratégie de tests en mode Agile

PURIFF : tests de (Non-)Régression

- L'objectif des tests de non-régression est de garantir que des défauts n'ont pas été introduits suite à la correction ou la modification de l'application.
- Les tests de non régression constituent **une activité fastidieuse notamment dans un contexte Agile**, caractérisé par des modifications et des livraisons fréquentes. Exécutés manuellement, ils sont **très consommateurs** de ressources.
- **Il est fortement recommandé d'automatiser ces tests.**
- **Rôle des testeurs :**
 - Concevoir les TNR
 - Automatiser les TNR
 - Adapter les TNR en continu

La Stratégie de tests en mode Agile

PURIFF : tests d'Intégration

- Les tests d'intégration représentent le niveau qui se situe juste au-dessus des tests unitaires.
- Ils consistent à vérifier que les différents modules fonctionnent correctement une fois assemblés.
- Ils sont en général **exécutés sur un environnement d'intégration continue de façon automatisée.**
- **Rôle des testeurs :**
 - Vérifier que les TI ont été exécutés
 - Participer au bilan des tests

La Stratégie de tests en mode Agile

PURIFF : tests Fonctionnels

- Les tests fonctionnels sont écrits et mis en place afin de vérifier que les fonctionnalités de l'incrément répondent aux spécifications définies dans le Backlog du produit.
- Ils correspondent aux **tests des critères d'acceptation des éléments du Backlog (user stories)**.
- Ces tests peuvent être déroulés manuellement ou de façon automatisée.
- **Rôle des testeurs :**
 - Concevoir les tests fonctionnels
 - Les exécuter (mais toute l'équipe peut tester, y compris le PO)
 - Participer au bilan des tests
 - « Toiletter » les tests (*Grooming the tests*) et les adapter
 - Automatiser les tests fonctionnels

La Stratégie de tests en mode Agile

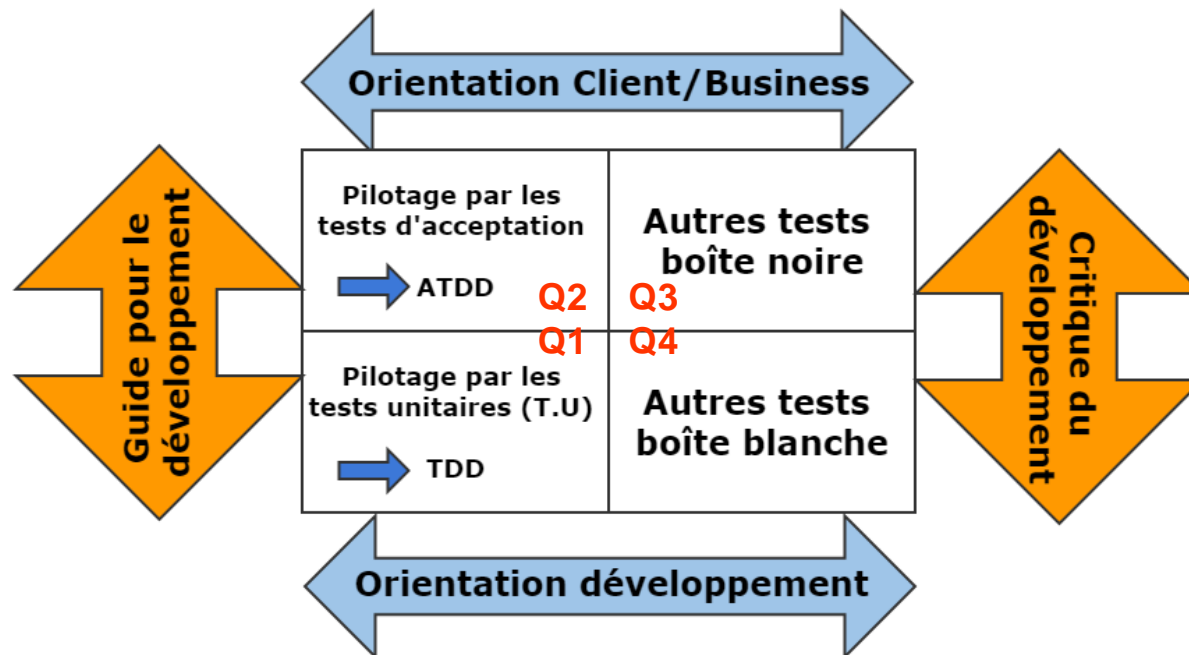
PURIF : tests (NON-)Fonctionnels

- Les tests non fonctionnels permettent de vérifier les aspects non fonctionnels de l'application.
- Ces aspects sont souvent définis comme étant des exigences non fonctionnelles (user stories techniques). Par exemple : les tests de montée en charge, les tests de robustesse, les tests de sécurité, les tests de portabilité, les tests de stress, etc.
- **Rôle des testeurs :**
 - Participer à la création et à l'exécution
 - Participer au bilan des tests

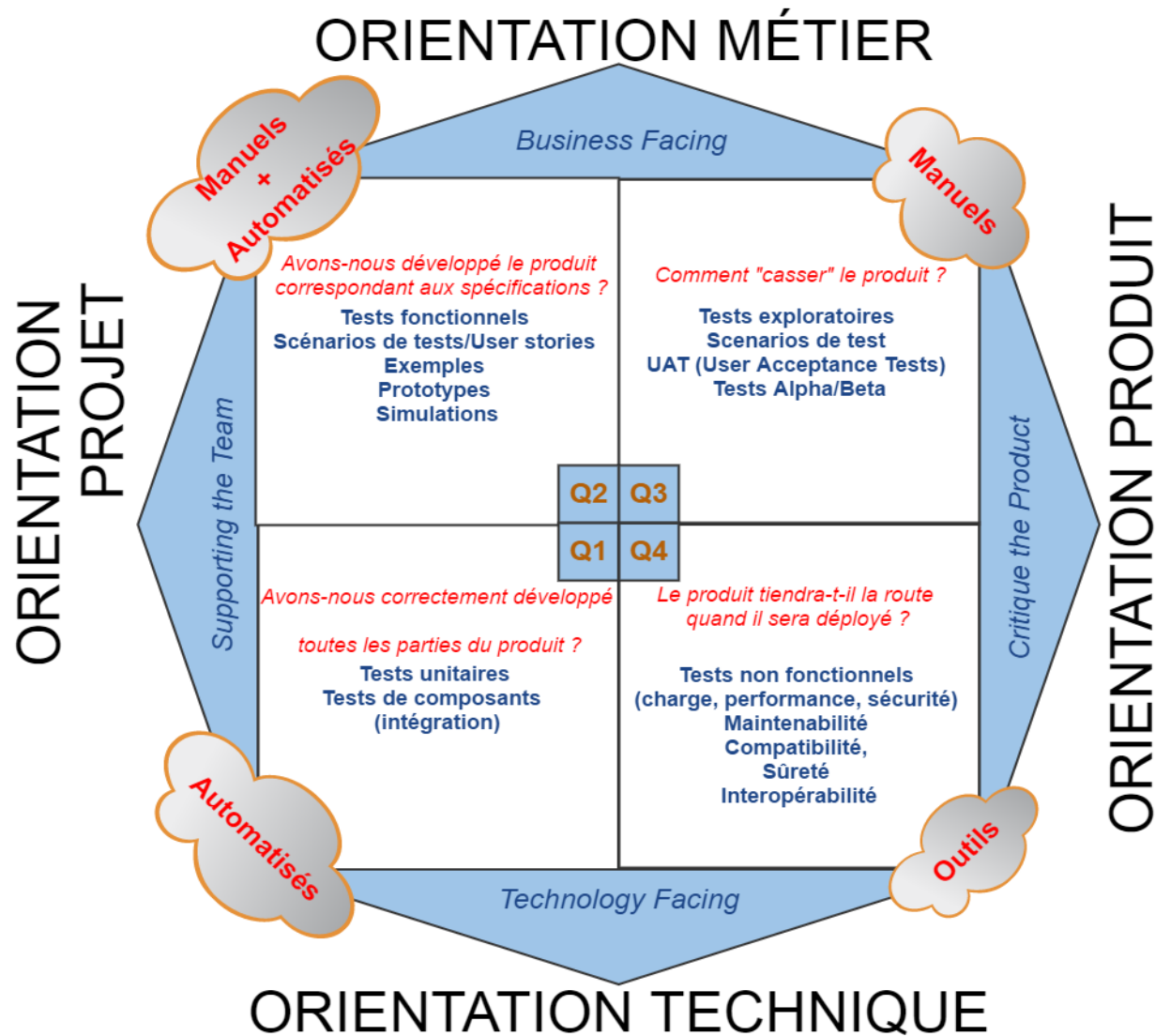
Les quadrants de Marick (1/2)

Les quadrants des tests, définis par Brian Marick [Crispin08], permettent d'aligner les niveaux de tests avec les types de test appropriés dans la méthodologie Agile

- **Q1** : Tests orientés **Technologie** pour **guider** l'équipe dans ses développements
- **Q2** : Tests orientés **Business** pour **guider** l'équipe dans ses développements
- **Q3** : Tests orientés **Business** pour **critiquer** le produit développé
- **Q4** : Tests orientés **Technologie** pour **critiquer** le produit développé



Les quadrants de Marick (2/2)



Le Manifeste du Testeur Agile (et pourquoi pas ?)

« Nous découvrons comment mieux tester des logiciels par la pratique et en aidant les autres à le faire.

Ces expériences nous ont amenés à valoriser :

- **Tester tôt**
*plutôt que **Tester à la fin***
- **Prévenir les bugs**
*plutôt que **Trouver les bugs***
- **Construire le meilleur système**
*plutôt que **Casser le système***
- **L'équipe est responsable de la qualité**
*plutôt que **Le Testeur est responsable de la qualité***

Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers. »

NB : Toute ressemblance avec un Manifeste connu serait totalement fortuite.

Le Manifeste du Testeur Agile

Tester tôt *plutôt que* Tester à la fin

- Le **caractère incrémental du développement agile** implique un **haut degré de solidité/stabilité applicative** obtenue dans le sprint N-1.
- **Les tests de bas niveau, automatisés**, sont exécutés en **permanence** et permettent de conserver ce degré de stabilité dans le code, comme **les tests fonctionnels/TNR garantissent la stabilité fonctionnelle**.
- Les tests fonctionnels exécutés au plus tôt permettent :
 - De guider le développement en lui indiquant la direction à suivre
 - De diminuer le coût, en concevant et en exécutant des tests
 - D'adapter et modifier les tests en fonction des changements fonctionnels demandés par le PO pendant le projet.

Le Manifeste du Testeur Agile

- **Prévenir les bugs** *plutôt que* **Trouver les bugs**
- En testant tôt, et en déterminant au plus tôt les critères d'acceptation des US, **les testeurs aident les développeurs à améliorer la qualité du code, et à prévenir ainsi les erreurs** susceptibles de causer des défaillances et par voie de conséquence les anomalies.
- Les testeurs, dans le contexte de l'équipe intégrée, doivent adopter un **état d'esprit agile en privilégiant la communication et l'interaction avec les développeurs.**

Le Manifeste du Testeur Agile

Construire le meilleur système *plutôt que* Casser le système

- Les activités de tests, dans un contexte traditionnel, sont vues par les développeurs comme une activité « destructrice ».
- Dans un contexte agile, c'est la **collaboration entre testeurs, développeurs et PO** qui **permettra de prendre les décisions** portant sur le Test.
- Grâce à cette collaboration, et à l'implication des testeurs sur la totalité du process Qualité, les tests contribuent à construire ensemble la qualité de produit.

Le Manifeste du Testeur Agile

- **L'équipe est responsable de la qualité**
plutôt que **Le Testeur est responsable de la qualité**
- « Attention continue à l'excellence technique et à une bonne conception », « Amélioration continue à intervalles réguliers » et co-localisation de l'équipe contribuent à faire de la qualité un sujet relevant de tous les membres de l'équipe, traité en même temps et à la même régularité que les autres tâches.
- En mode traditionnel, la « phase de qualification » en fin de processus tend à déresponsabiliser les développeurs, et à laisser le sujet aux équipes de recette.



Être testeur agile : Les compétences complémentaires

- L'expérience a montré que le profil de testeur le mieux adapté à la démarche agile est
 - **celui qui a des compétences techniques et/ou de développement** (ou des appétences confirmées par une pratique)
 - **et qui s'est orienté vers la qualification applicative.**
- Ceci permet au testeur de cohabiter et de communiquer d'une manière fluide et sans risque de divergence avec les développeurs de l'équipe.
- **Toutes les compétences et l'expérience d'un testeur fonctionnel « traditionnel » confirmé sont nécessaires et utilisées dans le contexte agile.**

Être testeur agile : Les compétences complémentaires

- Les techniques de test classiques sont toujours pratiquées par les testeurs au sein d'une équipe agile :
 - **Partitions d'équivalence**
 - **Analyse des valeurs limites**
 - **Tests par tables de décision**
 - **Tests de transition d'états**
 - **Tests de cas d'usage**
- **Les exigences non-fonctionnelles** sont elles aussi à tester. Elles peuvent :
 - Être documentées comme des US à part entière
 - Une US peut contenir des exigences de performance ou de fiabilité, envisagées comme critères d'acceptation de l'US.

Être testeur agile : Les compétences complémentaires

- **Tests exploratoires** : Dans le contexte d'une équipe agile, les tests exploratoires permettent, dans un temps réduit, d'explorer et comprendre le SUT, en même temps que de découvrir les défauts.

Tests exploratoires

Tests où le testeur :

- Contrôle activement la **conception** des tests **en même temps que ceux-ci sont exécutés**
- Utilise **l'information obtenue pendant les tests** pour **concevoir de nouveaux et meilleurs tests**



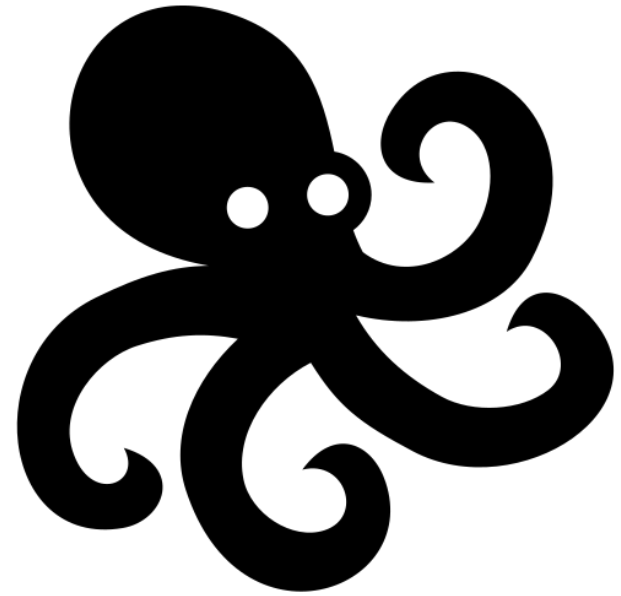
Être testeur agile : Les compétences complémentaires

- Les tests exploratoires induiront des tâches différentes en fonction de la stratégie qui les sous-tend :

Stratégie	Tâches induites
Basée sur les risques (Analytique)	Identifier/analyser des risques produit Concevoir des tests visant à réduire les risques En fonction des risques, viser une optimisation de la couverture (par ex. Tests exploratoires)
Basée sur les spécifications (Analytique)	Identifier les conditions de test Concevoir des tests visant à couvrir les conditions
Basée sur l'expérience (Taxonomie de défauts)	Concevoir des tests ciblant les défauts les plus probables

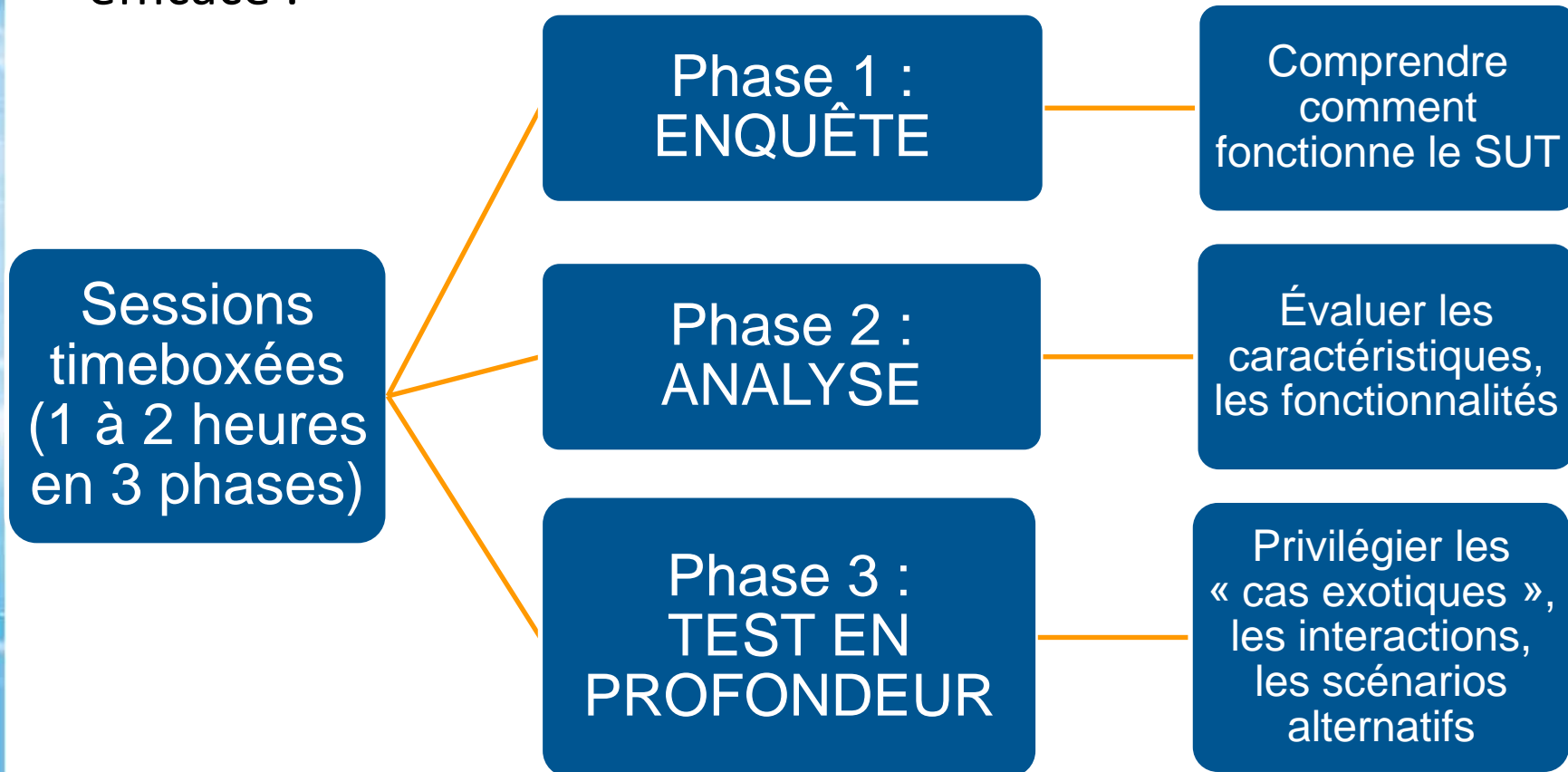
Être testeur agile : Les compétences complémentaires

- Des tests exploratoires, c'est la réalisation **simultanée** des tâches suivantes :
 - Découverte du produit
 - Découverte des défauts
 - Planification du travail de test
 - Conception des tests
 - Exécution des tests
 - Reporting



Être testeur agile : Les compétences complémentaires

- Pour « réussir » des tests exploratoires (≠ Monkey tests), la méthode de gestion par sessions est considérée comme la plus efficace :



Être testeur agile : Les compétences complémentaires

- Le contexte agile demande également des **compétences complémentaires**, acquises par l'expérience ou l'auto-formation :
- **Compétences en automatisation des tests** (il s'agira aussi d'aider les développeurs dans l'automatisation des TU et TI)
- **Compréhension du Développement** piloté par les tests (TDD), et du Développement piloté par les tests d'acceptance (ATDD)
- **Compétences en tests « boîte blanche »** autant qu'en « boîte noire »
- **Compétences basées sur une expérience** solide, afin de garantir l'efficacité des tests exploratoires

Être testeur agile : Les compétences complémentaires

De solides **compétences relationnelles** sont également nécessaires:

- **Être positif** et orienté solution
- Avoir un **esprit critique**
- **Obtenir de l'information** auprès des personnes
- **Donner la visibilité** sur les résultats, l'avancement et la qualité
- **Contribuer, avec l'équipe, à la définition** des US et de leur testabilité
- **Collaborer**, savoir travailler en **binôme**
- Être **réactif**
- Être **organisé** et **AUTONOME**

Être testeur agile : Les difficultés culturelles

Tout changement nécessite une adaptation ... plus ou moins aisée ! Parmi les points durs, on peut citer :

Peur de la perte d'identité : Liée à la peur du changement, à l'expérience acquise en mode traditionnel, de ne plus être « reconnu », de perdre un « statut » de qualificateur.

Manque de formation : Il concerne l'ensemble de l'équipe, y compris le PO. Les compétences de chacun doivent évoluer pour atteindre le niveau d'équipe intégrée.

Incompréhension des concepts agiles : Il est nécessaire de clarifier les concepts et les processus, en équipe, en raison de leurs multiples implémentations. Il faut également savoir ne pas savoir ... et savoir demander !

Évolution du management : L'ancien « CP MOA » devient « PO ». Il passe du management au coaching, ce qui implique de laisser le pouvoir à l'équipe. Ne pas évoluer conduit à un management inadapté, et met en danger l'équipe et le projet.

Comportement : Ouvert, à adapter/remettre en question en permanence.

Être testeur agile : Les risques pour le testeur

Travailler en contexte agile peut amener, pour le testeur, des risques liés aux organisations de test :

- En travaillant en étroite proximité avec les développeurs, **le testeur risque de perdre sa mentalité propre** de testeur.
- Face à l'inefficacité ou au faible niveau des pratiques de l'équipe, les testeurs insuffisamment intégrés **risquent de devenir tolérants ou silencieux**.
- Avec de trop nombreux changements, dans le cadre d'itérations contraintes dans le temps, **les testeurs risquent de ne pas garder le rythme**.

Pour pallier ces risques, les organisations peuvent considérer différentes solutions pour préserver l'indépendance des testeurs.

Par exemple, dans un cadre de « Scrum of Scrums », une équipe Scrum de test indépendante et identifiée, au service de plusieurs équipes de développement.

Être testeur agile : Le rôle du testeur dans l'équipe

Des activités qui génèrent et fournissent du « feedback » :

- **Collaborer** avec les développeurs et les parties prenantes métier
- **Coach** les autres membres de l'équipe sur les aspects pertinents du test
- **Assurer la prise en compte du test** lors des plannings de Release et de Sprint
- Contribuer à la **stratégie** de test
- Contribuer à la **DOR et la DOD (Definition of Ready/Done)**
- Contribuer à la **création des US et valider leur testabilité**
- **Gérer les données et environnements** de test
- **Gérer les résultats** des tests et les **communiquer** à l'équipe
- **Mesurer la couverture** de test
- Participer à **l'automatisation des tests**, y compris TU et TI
- Participer aux **rétrospectives** de l'équipe et y **suggérer des améliorations**

The Agile Mindset – L'état d'esprit Agile

