



# Le Bon, la Brute et le Truand Le pouvoir des trois dans l'Agilité

Laurent BOUHIER

# WANTED



**LAURENT BOUHIER**

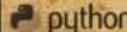
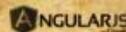
**SENIOR TEST CONSULTANT**

POUR DES ACTIVITES EN INFORMATIQUE :

- INFORMATIQUE EMBARQUEE
- INFORMATIQUE DE GESTION
- WEB STARTUP
- FORMATION ET COACHING AGILE ET TESTS

# REWARD

 [laurent bouhier](#)  [laurent-bouhier-76935b61](#)  [l.bouhier@acpqualife.com](mailto:l.bouhier@acpqualife.com)



# Sommaire



- Rappels sur l'Agilité
- Western: Erreurs fréquentes
- Analyse de ces erreurs
- État d'esprit du testeur Agile
- Conclusion

# Sondage

- Asseyez-vous si une de ces phrases est vraie pour vous :
- Les testeurs ne peuvent pas travailler tant que le développement n'est pas fini
- Les tests arrivent toujours après les développements
- L'automatisation arrive encore plus tard
- Il y a de la pression juste avant les livraisons
- On cherche toujours un coupable pour les bogues (c'est de sa faute...)

# Manifeste Agile

*“Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire.”*

## Valoriser

Les individus et leurs interactions

Des logiciels opérationnels

La collaboration avec les clients

L'adaptation au changement

## Plus que

Les processus et les outils

Une documentation exhaustive

La négociation contractuelle

Le suivi d'un plan

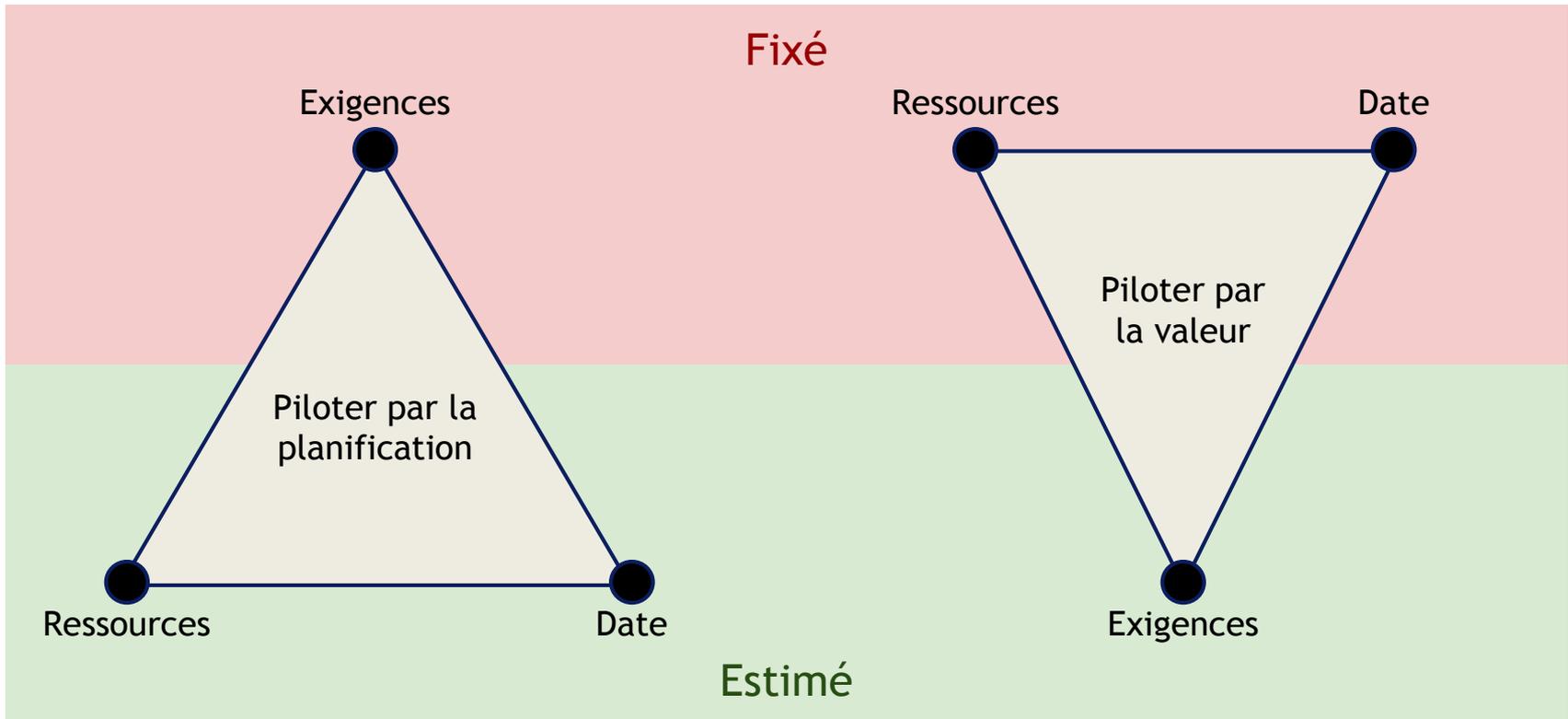
*“Nous reconnaissons la valeur des seconds éléments, mais privilégions les premiers.”*

<http://agilemanifesto.org/iso/fr/>

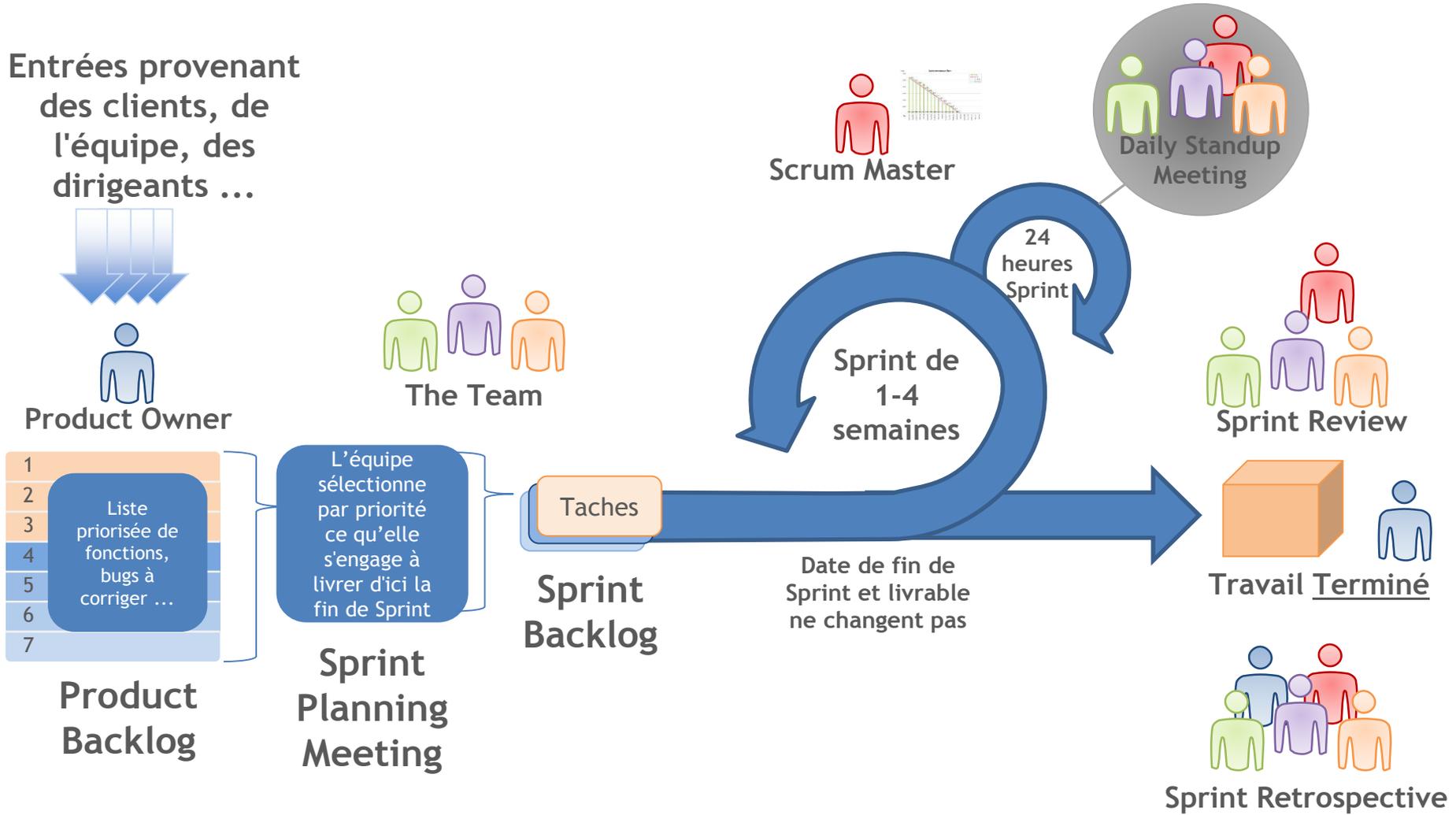
# “Triangle de fer”

Approche traditionnelle

Approche Agile



# Scrum

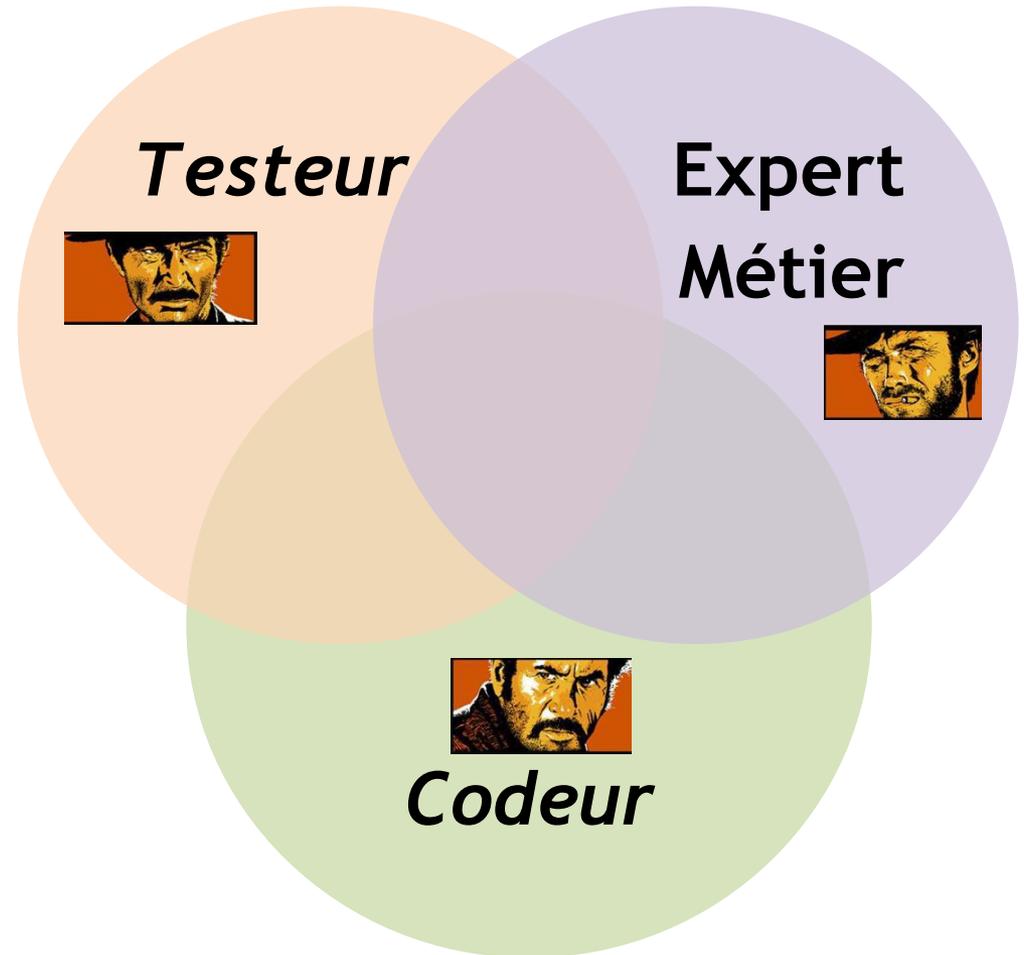


# Sommaire



- Rappels sur l'Agilité
- **Western: Erreurs fréquentes**
- Analyse de ces erreurs
- État d'esprit du testeur Agile
- Conclusion

# Premier principe : Communication !



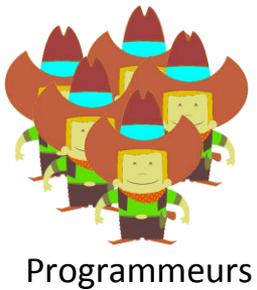
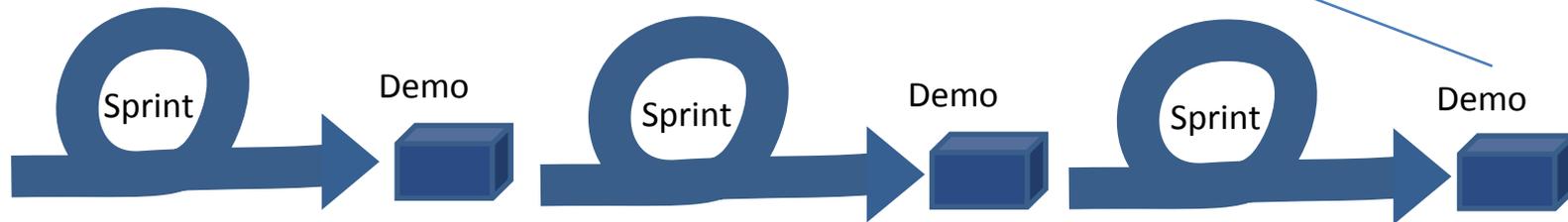
Source: Agile Testing: A Practical Guide for Testers and Agile Teams by Lisa Crispin, Janet Gregory

Une nouvelle application

- Agilité à la mode
- Scrum



Pas ce que veut...  
Le client ne sait pas ce qu'est l'agilité !

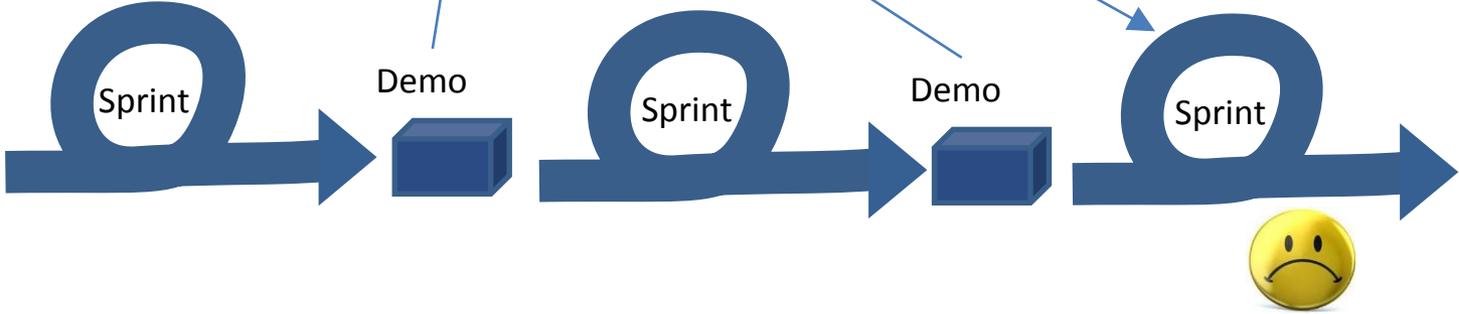
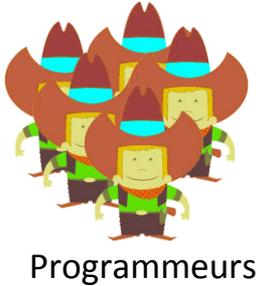




Product Owner n'est plus un proxy



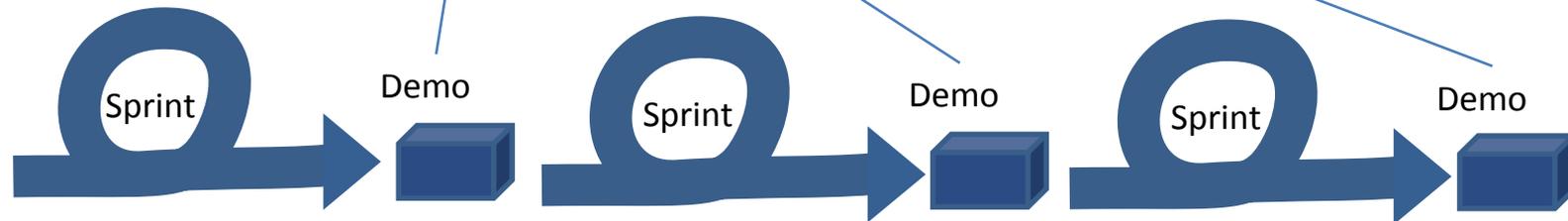
Veut changer le contenu pendant le sprint !



- ✓ Le client est inclus dans le processus Agile
- ✓ Un P.O. certifié !



Régression  
→ Manque de qualité



Un département QA existe dans mon organisation!  
On va lui demander de tester et d'automatiser



Plus d'Effet



Livraison d'une release testée



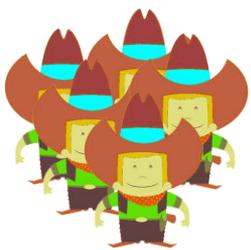
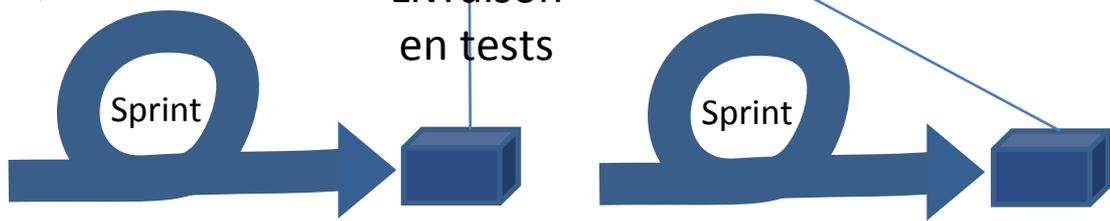
- Tests boîte noire
- Automatisation des tests E2E

tests E2E de maintenance  
Difficulté d'exécution  
→ Acceptation d'un taux de tests « passed » < 100% !



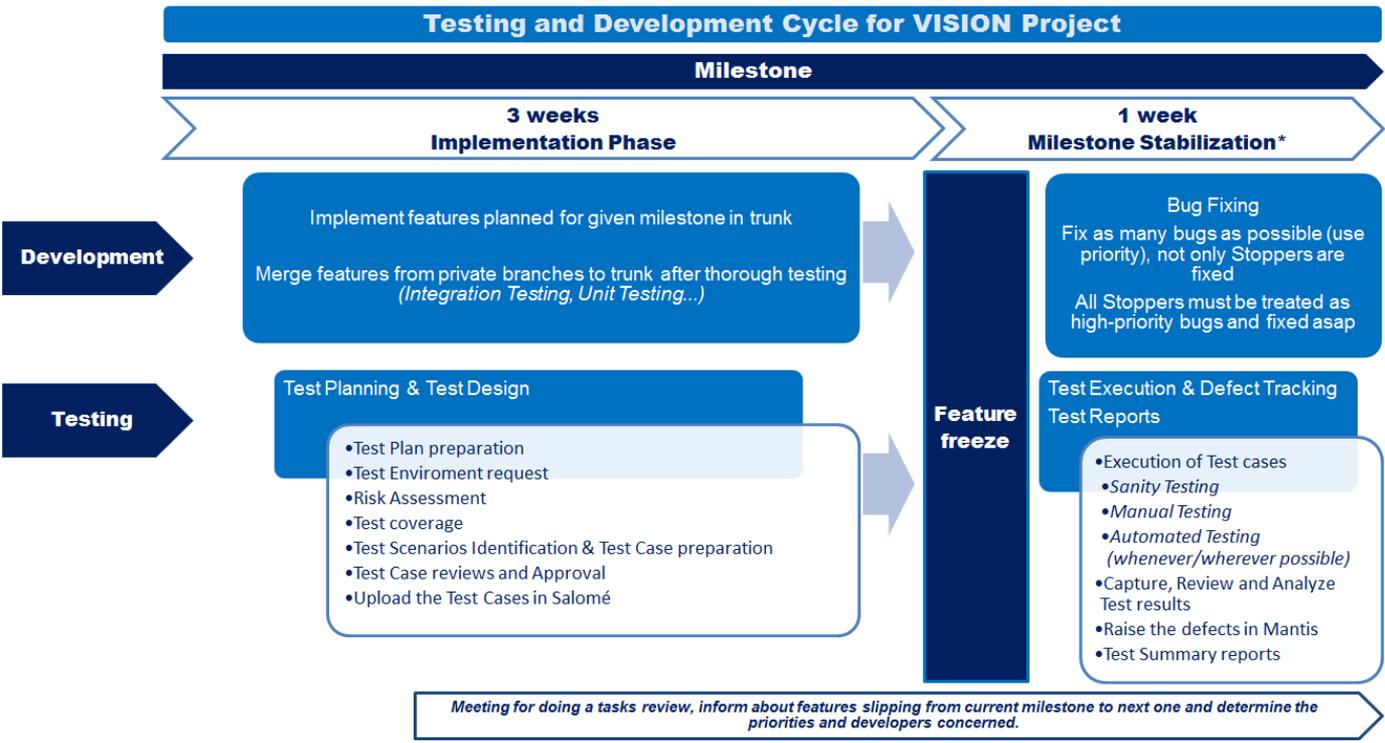
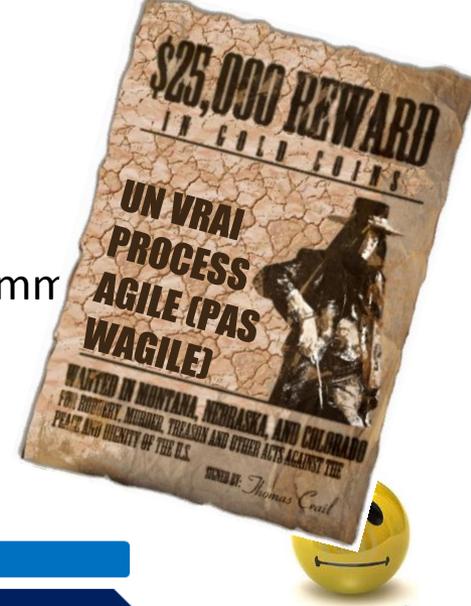
Product Owner  
Bugs à corriger

Livraison en tests



Gestion des Défauts

- Conflit avec le sprint courant
- SCM, Merge, relivraison...



- Sensible aux retards
- Effet mini waterfall
- "truck factor" = 1
- difficulté de maintenance des scripts
- Problématique de gestion des tests de régression



***THE END***

**des scenarii “comment louper son scrum !”**

# Sommaire



- Rappels sur l'Agilité
- Western: Erreurs fréquentes
- Analyse de ces erreurs
- État d'esprit du testeur Agile
- Conclusion

# Expert Métier / Product Owner

Veut être  
chef de  
projet

Pas un vrai  
Product  
Owner  
(proxy)



Veut changer  
le contenu  
pendant le  
sprint

Ne sait pas définir les  
critères d'acceptance  
ni prioriser le backlog  
(tout est important !)

# Codeur

Ne veut PAS tester (même les tests unitaires !)

Considère que seuls des codeurs sont nécessaires dans le processus de développement



N'applique et ne connaît pas les principes Agile (TDD, pair programming...)

Ne s'intéresse pas au fonctionnel

# Testeur

Pas assez technique

Ne veut pas partager ses tests (sinon il n'y aura pas de bugs)



Agile est pro-actif, les métriques de tests standards ne le sont pas !

Ne s'intéresse pas aux tests unitaires (problème du codeur)

# Pourquoi ces problèmes ?

## SCRUM GUIDES

Scrum team = Product Owner+Scrum Master+Développeurs

## MAIS

Développeur  $\neq$  Codeur  $\Rightarrow$  manque de compétences  
**Développement = Spécifications + Code + Tests**

- Manque de maturité
  - Pas de formation, ni de coaching
  - Pas de testeur, ni de stratégie de tests...
  - Mauvaise définition du « Done » (« Terminé »)
  - Mauvaise stratégie d'automatisation (pas ou peu de tests unitaires)

# Sommaire



- Rappels sur l'Agilité
- Western: Erreurs fréquentes
- Analyse de ces erreurs
- État d'esprit du testeur Agile
- Conclusion

# Agile = Une seule équipe

- Le processus Agile donne la responsabilité à une équipe intégrée et auto-organisée
- Cette équipe est responsable de la qualité du produit (la qualité est l'affaire de chacun !)
- Cela permet une meilleure efficacité dans la correction des bugs et dans la maintenance des tests
- Les tests sont primordiaux dans cette approche !

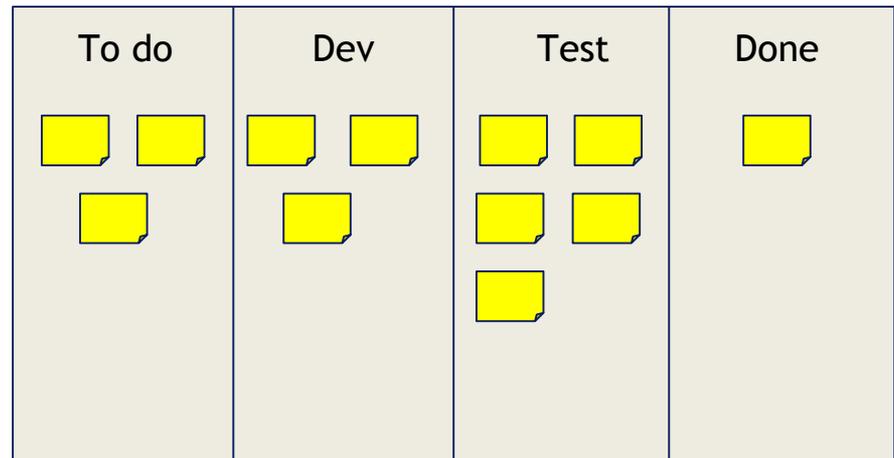
# Scrum board...

DEV

TEST

⚠ Le test est une phase

J'ai ma propre colonne



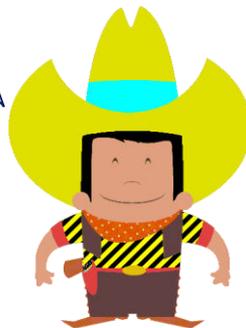
# Scrum board : bonne pratique

User Story

DEV

TEST

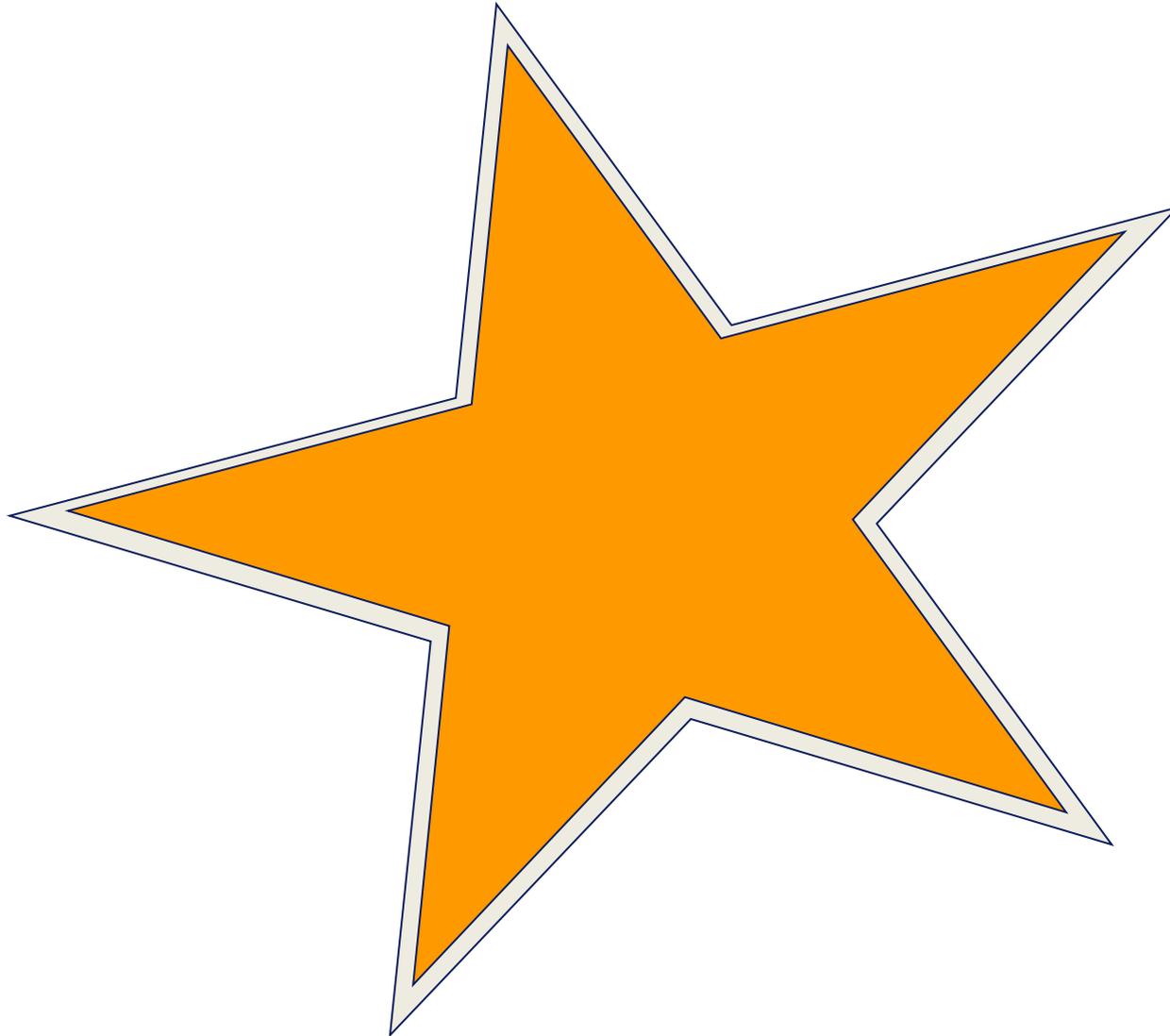
Je fais partie du board



## Le test est une ACTIVITÉ !

To do	Dev	Show me	Done
			    
   		 	
    			
   			

# Combien de coins ?



# Prévenir les bugs !

Quoi ?

Pour qui ?

Comment ?



Pourquoi ?

Quand ?

# Une user story ?

Pouvez-vous développer cette User Story ?

As who, I want what so that why

“L’affinage” est important

Non !

Seulement une Carte, il manque:  
la Conversation et la Confirmation  
3C principe!



# Vérifier... ou tester

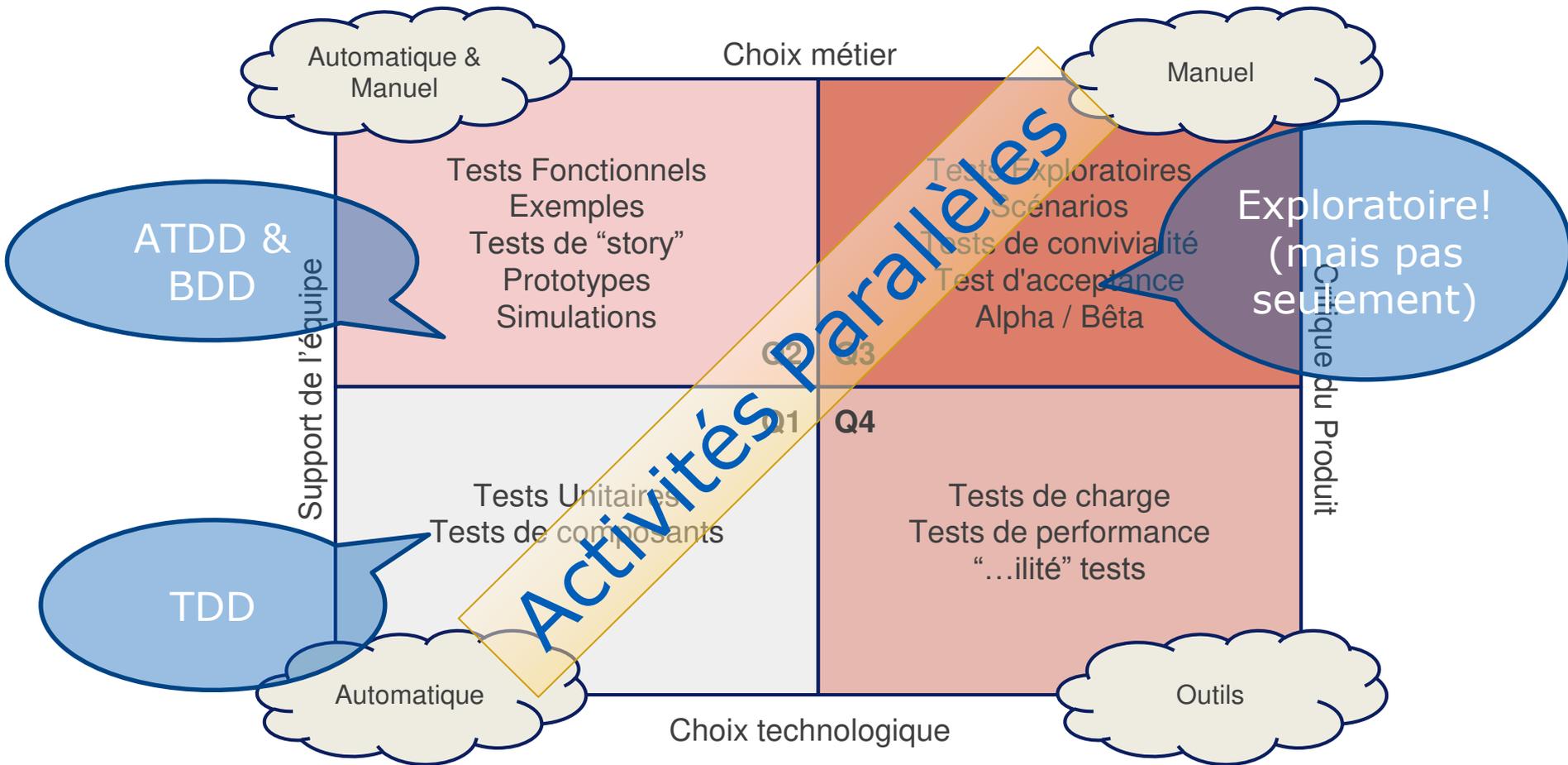
- Les testeurs traditionnels n'aiment pas Agile car comment tester sans spécification ?
- Certains considèrent que le métier de testeur consiste à vérifier que le système marche comme il est spécifié
- MAIS cela ne fait QUE vérifier que le codeur a suivi les spécifications, mais ne prouve RIEN quant à la qualité du logiciel !

**Vérifier** = Automatiser

**Valider** = Comprendre

**Tester** = **Vérifier & Valider** !

# Agile Testing Quadrants



Agile Testing Quadrants in "Agile Testing: A Practical Guide for Testers and Agile Teams" by Lisa Crispin , Janet Gregory

# Activités de Tests

## Durant un sprint

Automatiser  
les tests de  
critères  
d'acceptance

Définition de  
« Done », représente  
des exigences  
exécutables

Automatiser  
les tests  
Unitaires

Pilote le design,  
représente des  
spécifications  
exécutables

Tests  
Exploratoires

Découvrir les risques  
tôt

# Ne cassez pas le système !

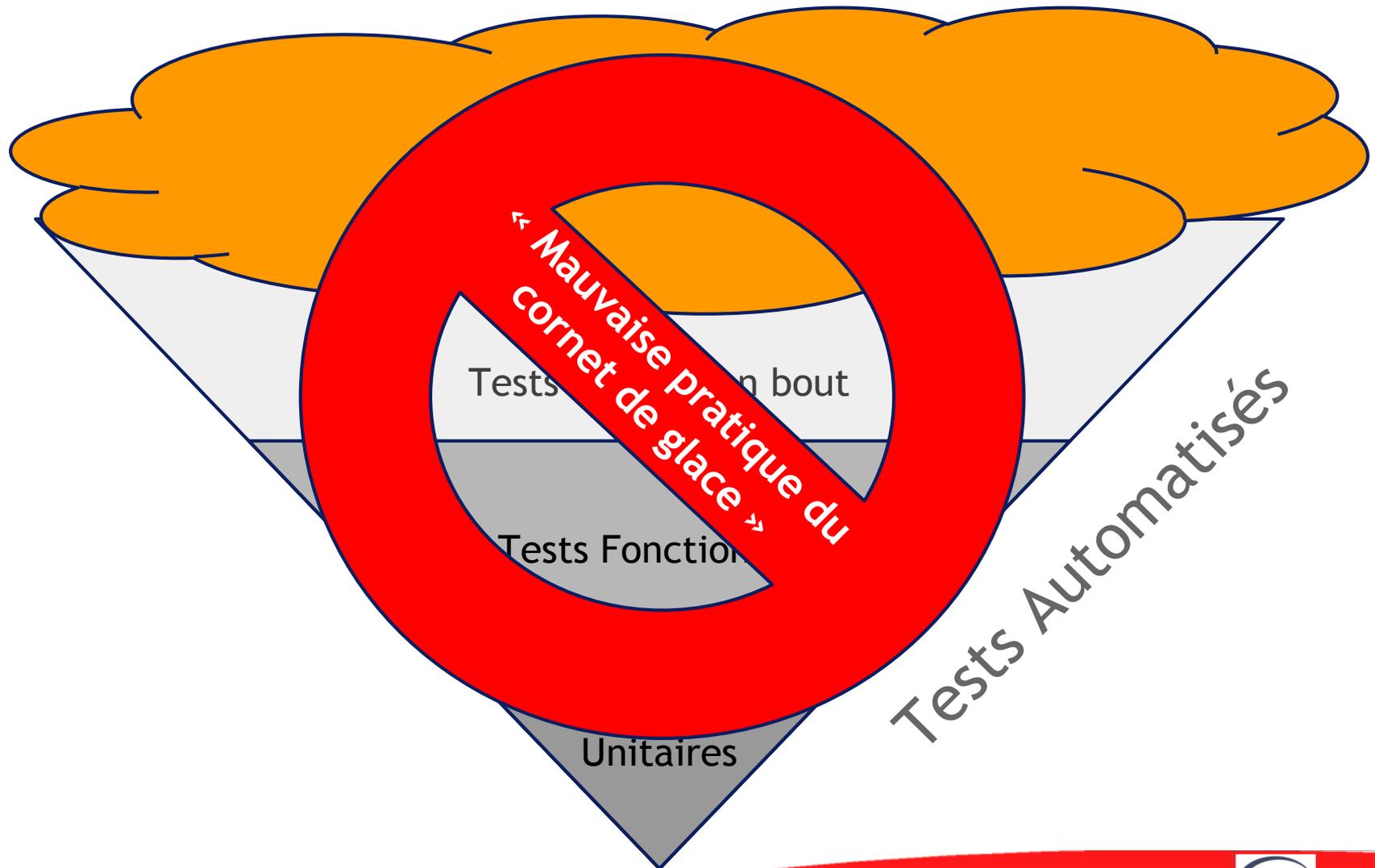


Le testeur traditionnel  
veut casser le système !

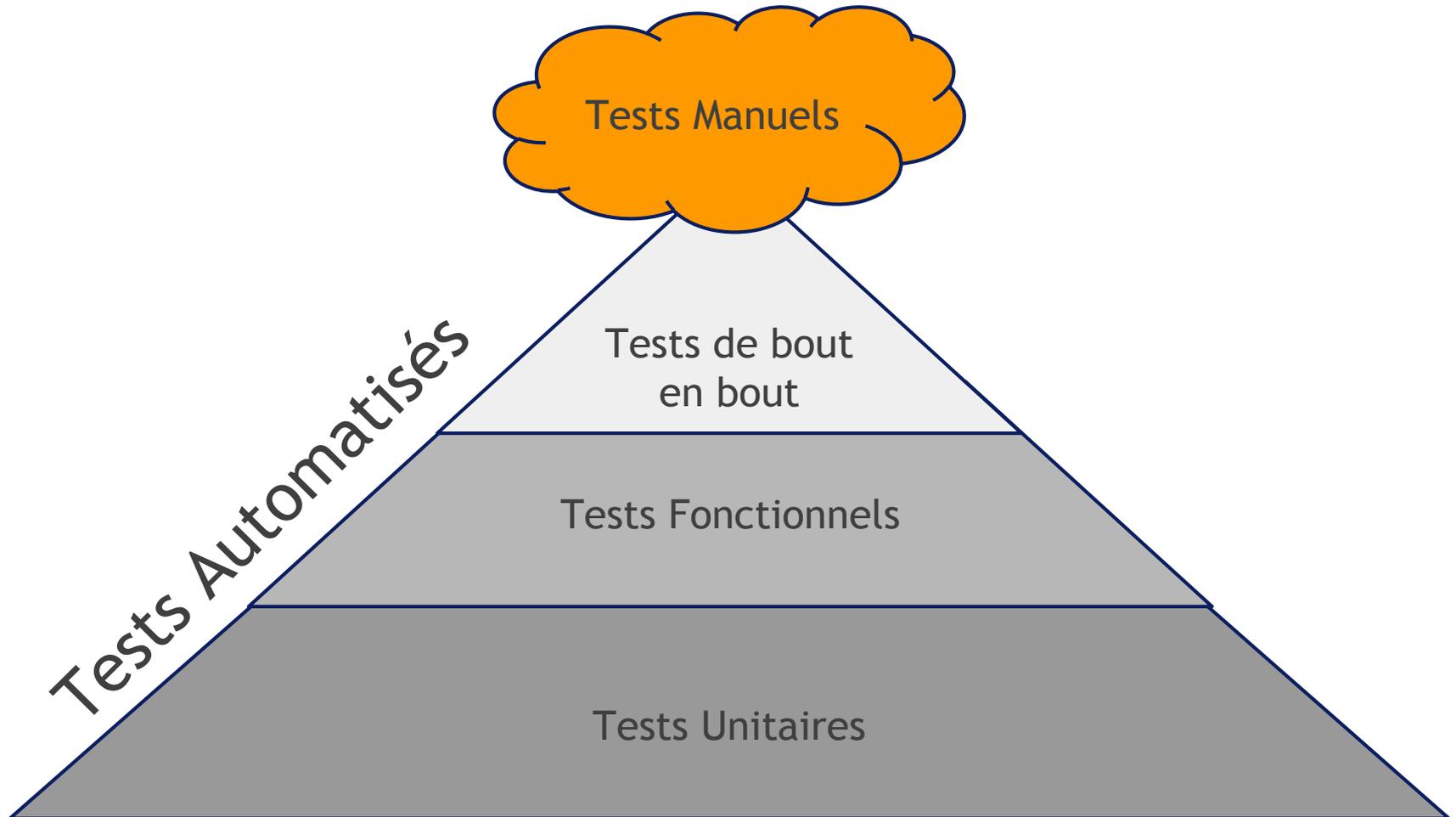
Le testeur Agile doit fournir  
les outils pour construire le  
meilleur système (ATDD)



# Automatiser oui mais...



# Privilégier l'automatisation de bas niveaux



Agile Pyramid - Mike Cohn in "Succeeding with Agile"

# Sommaire



- Rappels sur l'Agilité
- Western: Erreurs fréquentes
- Analyse de ces erreurs
- État d'esprit du testeur Agile
- **Conclusion**

# État d'esprit en projet Agile

- Tester en projet Agile ne veut pas dire tester comme avant mais dans un sprint !
- L'état d'esprit de toute l'équipe concernant le test doit changer :
  - Le test n'est plus une phase mais une activité
  - Prévenir les bugs plutôt que les trouver
  - Tester c'est plus que vérifier
  - Ne pas essayer de casser le système mais aider à construire le meilleur système possible
  - Toute l'équipe est responsable de la qualité

# Manifeste du Testeur Agile

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

**Valoriser**

Tester tôt

Prévenir les bugs

Construire le meilleur système

L'équipe est responsable de la qualité

**Plus que**

Tester à la fin

Trouver des bugs

Casser le système

Le testeur est responsable de la qualité

# Testeur dans un projet Agile ?



- Compétences classiques de tests, mais pas « seulement tester »
  - Grandes compétences de communication
  - Citoyen de première classe (fait partie de l'équipe)
  - Compétences techniques et d'automatisation
  - Connaissances Agile (TDD, ATDD...)
  - Coach pour les autres membres de l'équipe dans tous les aspects pertinents de tests (y compris les tests unitaires !)

Merci

