



ASTEK Madagascar

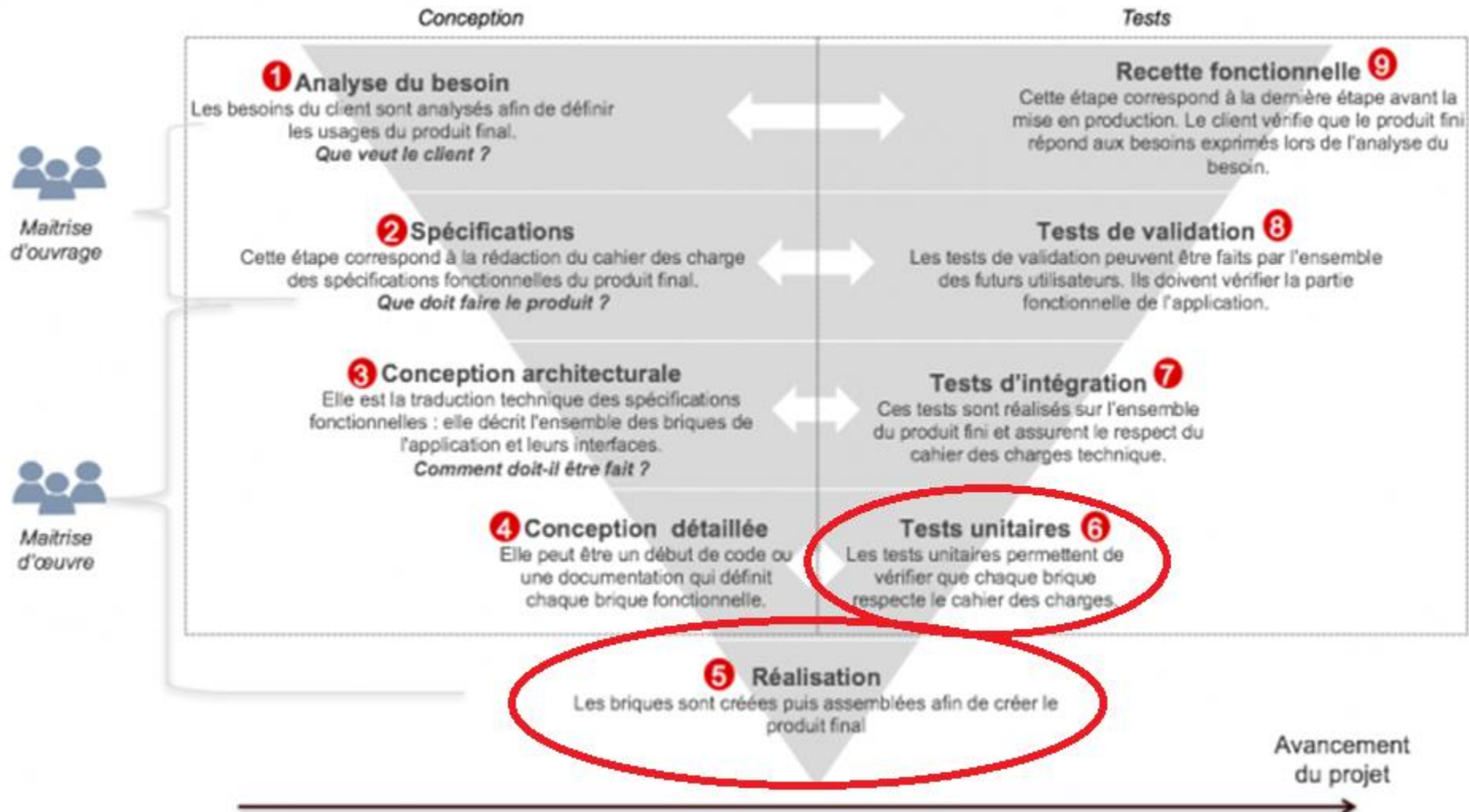
# Cycle V, Scrum et tests unitaires...

26/07/2016



# Cycle en V

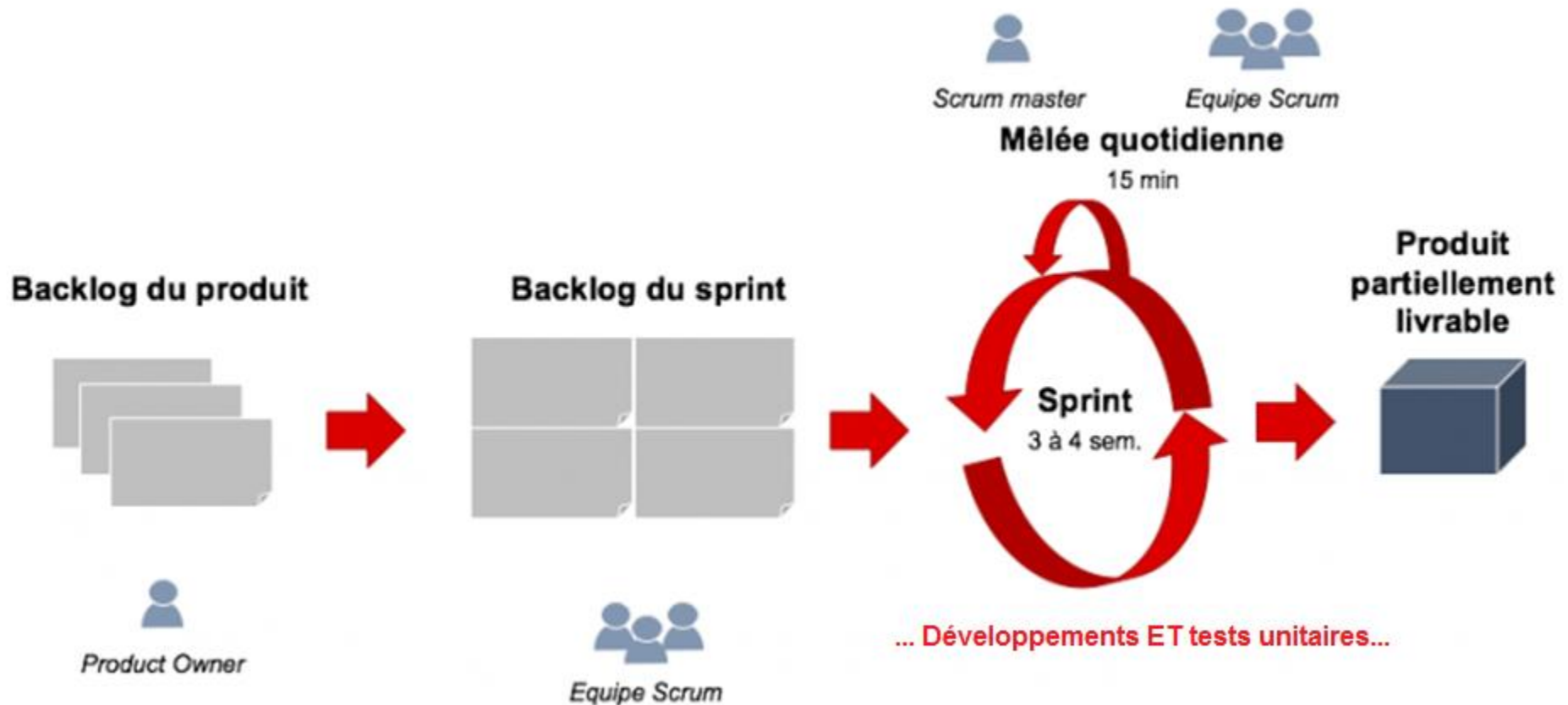
## Développements ET tests unitaires





# Scrum

## Développements ET tests unitaires





# Comparatifs

Thème	Cycle en V	SCRUM
Cycle de vie	Phases séquentielles	Processus itératif
Livraison	À la fin de la réalisation de toutes les fonctionnalités → livraison tardive	Utilisation partielle du produit suite à la priorisation des besoins → livraison plus rapide
Contrôle Qualité	À la livraison finale (fin du cycle de développement) → effet tunnel	À chaque livraison partielle au client
Spécification	Pas de changement possible sans revenir à la phase de spécifications et repasser par toutes les autres phases → délais & coûts supplémentaires	Spécifications plus souples en ajoutant / modifiant les fonctionnalités aux sprints suivants qui n'étaient pas prévues au départ → principal atout de la méthode Agile
Planification	Plans détaillés basés sur des exigences stables définies dès le début du projet	Planification adaptative et ajustements si nécessaires en fonction des nouvelles demandes
Équipe	Intervention uniquement dans la phase de développement, pas de vision globale du projet	Engagements, échanges et prises de décisions collective par l'équipe
Documentation	Quantité importante	Strict nécessaire



# Qualité du code et tests unitaires

## OBJECTIFS : Testabilité, maintenabilité et évolutivité

Le testing unitaire repose au fond sur un principe très simple : **un système aussi gros et complexe qu'il soit est toujours composé de différentes parties plus petites que lui. Il est indispensable (mais pas suffisant) que chacune de ces parties remplisse correctement sa fonction pour que l'ensemble soit fonctionnel et fiable**

- **Qualité du code :**
  - ✓ <https://rules.sonarsource.com/csharp>
  - ✓ Modularité (=> Non redondance, réutilisabilité)
  - ✓ Intégrer l'extension SONAR <https://www.sonarlint.org/> ou resharper (payant)
  - ✓ Usage et respect des conventions de développement de l'entreprise
  - ✓ Chez ASTEK, la qualité des codes sources est contrôlée au quotidien en mode batch avec Sonarcube
- **Tests unitaires des modèles et des contrôleurs** : projets de tests unitaires
- Tests unitaires des vues : tests manuels de l'IHM pendant la phase de développement.  
**Font partie intégrante des tâches du développeur.**
- **Le testeur officiel de l'application n'est pas là pour faire les tests unitaires à la place des développeurs. Le testeur exécute les tests d'intégration et/ou les tests de recette fonctionnelle.**



# Tests unitaires manuels des vues pendant la phase de développement

## OBJECTIFS :

- Vérifier la conformité de chaque vue (écran, page WEB...) avec les spécifications
- Vérifier le bon fonctionnement et la robustesse de chaque vue
- Pré-requis : bonne compréhension du besoin
- Aspect visuel de l'IHM : conformité avec le design/maquettage initial + présentation élégante et uniforme (libellés, champs, onglets...)
- Tests de tous les contrôles/validations côté client et (contrôles Javascript essentiellement)
- Tests de tous les cas d'affichage des messages applicatifs (information, questions, warnings, erreurs...)
- Tests cas standards : **Données standards en entrée + actions avec la souris**
  - ✓ Création + enregistrement OK ?
  - ✓ Modification + enregistrement OK ?
  - ✓ Recherche OK ?
  - ✓ Visu OK ?
  - ✓ Suppression OK ?
  - ✓ Autres fonctionnalités OK ?



# Tests unitaires manuels des vues pendant la phase de développement

- Tests cas limites : **Données moins standards en entrée + usage claviers**

- ✓ Champs texte : saisir des caractères spéciaux
- ✓ Champs texte : saisir une chaîne très longue
- ✓ Champs numérique : saisir des chaînes ou des décimales . ou ,
- ✓ Champs date : saisir une date mal formatée
- ✓ Emails ou numéro de téléphone : données mal formatées
- ✓ Vérifier si la touche Entrée valide la page
- ✓ ...

Vérifier le bon fonctionnement de toutes les fonctionnalités en mode « cas limites ».

Intérêt de positionner le maximum de contraintes de validation au niveau du modèle car certains frameworks comme .NET permettent de générer aussi automatiquement les contrôles côté client (contrôles Javascript / Ajax) - en plus des contrôles côté serveur.



# Leitmotiv

## **!!! FAIRE BIEN DU PREMIER COUP !!!**

- Spécifications « OK » (Sprints Scrum ou Cycle en V)
- Maquettage et design « OK »
- Architecture « OK »
- Conception « OK »
- \*\*\* DEVELOPPEMENTS ET TESTS UNITAIRES \*\*\*
- Qualité du code « Bien du premier coup »
- Tests unitaires (modèles et contrôleurs) « OK »
- Tests unitaires manuels des vues « OK » et exhaustifs
- \*\*\*\*\*
- Tests d'intégration « OK »
- Recette fonctionnelle « OK »
- De façon générale : **FAIRE BIEN DU PREMIER COUP** dans notre état d'esprit et nos comportements au quotidien 😊

**Résultats** : Minimums de bugs et de retours

Peu ou pas de dépassement de délai / charge / budget

Codes sources de bonne qualité

Client satisfait ! Tous les intervenants satisfaits ! 😊