

## Définition de Fini

La **Définition de Fini** ou DoD (*Definition of Done*) est un accord co-créé par une équipe, qui s'applique à tous les éléments sur lesquels travaille cette équipe.

Elle inclut un ensemble minimum d'activités ou de conditions à respecter pour atteindre le niveau de finition et de qualité requis pour chaque élément de travail.

C'est un accord de l'équipe, qui peut s'améliorer en grandissant au fil du temps.

## Que sont les DoD Kards ?

Les DoD Kards sont un jeu de cartes dont le but est de réfléchir et de parvenir à un consensus entre tous les membres de l'équipe sur les critères à inclure dans sa définition de fini (DoD).

La portée du jeu est la DoD dans le contexte du développement logiciel, appliquée aux éléments d'un Backlog Produit (PBIs, souvent exprimés comme histoires d'utilisateurs), mais peut être extrapolé à d'autres contextes.

Il est recommandé d'y jouer régulièrement (tous les 2 à 3 mois), afin d'améliorer la DoD de l'équipe.

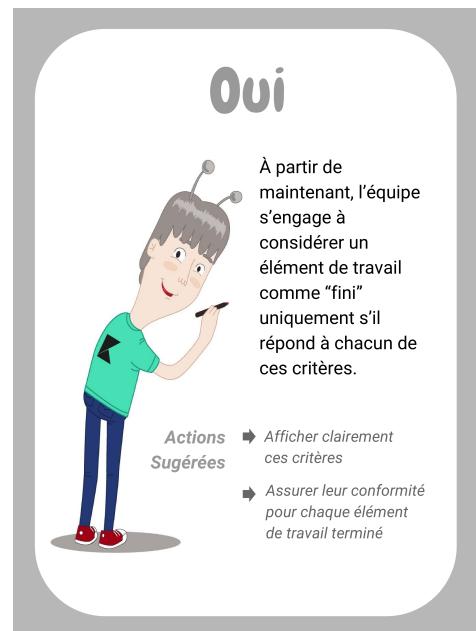
Pour apporter vos commentaires, vous pouvez utiliser le hashtag #DodKards sur [Twitter](#)



Les DoD Kards sont une création de [Kleer](#), par [Camilo Velasquez](#) et [Thomas Waller](#), inspirées [Definition of Done Exercise](#) de [David A. Koontz](#), et sont distribuées sous licence Creative Commons Attribution-ShareAlike 3.0 Unported.

## Comment jouer ?

1. Placez les cartes "Oui", "Plus tard" et "???" en ligne sur la table ou sur le mur.
2. Mélangez toutes les cartes de critères (bleues, noires, vertes, et roses) et placez-les face cachée en tas à la portée de tous.
3. Placez dans un autre tas les cartes de critères vides jaunes.
4. Un par un, chaque joueur :
  - a. Prend la première carte du tas de critères y la lit à haute voix.
  - b. Place la carte dans la colonne qu'il considère la plus correcte ("Oui", "Plus tard" ou "??").
  - c. Si quelqu'un n'est pas d'accord, l'équipe en discute afin de parvenir à un consensus pendant une durée préétablie. S'il n'y a pas de consensus à la fin de cette durée, placez la carte dans la colonne "???".
5. Répétez le point 4 jusqu'à l'épuisement des cartes critères ou jusqu'à ce que l'équipe le décide.
6. À n'importe quel moment un joueur peut écrire un nouveau critère personnalisé sur une carte vide jaune et l'utiliser à son tour.



## Plus tard



L'équipe ne peut pas s'engager pour ces critères pour le moment, mais ils pourraient être pris en compte dans le futur.

- Actions Sugérées**
- ➡ Réviser ces critères à nouveau dans quelques mois
  - ➡ Planifier d'éventuelles actions pour avancer vers l'inclusion de certains de ces critères

## Non



Ces critères ne sont pas clairs pour l'équipe ou ne peuvent pas être appliqués; l'équipe ne les voit pas comme faisant partie de son DoD, ni maintenant, ni dans le futur; ou il n'y a pas eu d'accord à leur propos.

- Actions Sugérées**
- ➡ Faire des recherches postérieures pour mieux comprendre certains de ces critères
  - ➡ Discuter postérieurement à nouveau les critères sans consensus

## Révision

### Tout le code concerné a été revu par des paires

En Révision de Code ou Programmation par Paire



## Révision

### La démonstration correspondante a été préparée

Y compris les données à utiliser, le script et les différents pas à suivre.



Révision

## Une pré-démonstration (démonstration à blanc) a été réalisée

Entre les membres de l'équipe pour préparer et améliorer la démonstration réelle



Révision

## La dette technique n'augmente pas (sans justification)

L'équipe estime que l'élément terminé n'introduit pas de nouvelle dette technique, ou, le cas échéant, qu'il y a une bonne justification



Révision

## Le Product Owner a vu une démonstration ou fait les tests nécessaires

De l'histoire d'utilisateur et l'a acceptée



Qualité

## Les tests nécessaires ont été exécutés

Et il ne reste pas d'incident majeur à résoudre



Qualité

## L'élément a des tests automatisés

Et ils ont tous été exécutés avec succès

 kleer™

Qualité

## L'élément a des test unitaires

Qui couvre plus de \_\_\_% du code

 kleer™

Qualité

## L'élément a été vérifié par le serveur d'Intégration Continue

Sans provoquer d'incident

 kleer™

Qualité

## Les standards de code sont respectés

... lorsqu'ils s'appliquent, éventuellement grâce à des vérifications automatisées

 kleer™

## Environnements

### Le code est stocké dans un contrôleur de source

Avec les commits, tags et/ou commentaires correspondants.



## Environnements

### Le build a été déployé dans l'environnement de test

... sans erreurs et est disponible pour les test dans cet environnement.



## Environnements

### Le build a été exécuté dans l'environnement

—  
... sans erreurs et est disponible pour son exécution dans cet environnement.



## Environnements

### Le build a été déployé dans l'environnement productif

... sans erreurs et est disponible pour son exécution dans cet environnement.



#### Environnements

### L'environnement de livraison a été préparé correctement

Suivant le cas, l'environnement de test, stage ou production a été installé et configuré correctement.



#### Diffusion

### La documentation correspondante a été écrite ou actualisée

Par exemple: définition d'API, Guide de l'utilisateur, Décisions importante de Design, Release Notes.



#### Diffusion

### Les personnes ou secteurs concernés ont été prévenus

S'ils avaient besoin de savoir que l'élément correspondant a été résolu.



#### Diffusion

### Les radiateurs d'information correspondants ont été actualisés

Par exemple: outils de gestion (Trello, Jira, etc.), tableau physique des tâches, etc.



Diffusion

## Les utilisateurs concernés ont été formés

Pour pouvoir utiliser correctement le logiciel livré.

 kleer™

Diffusion

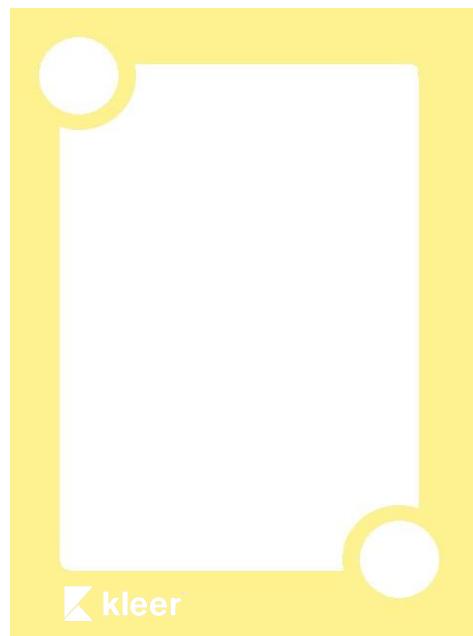
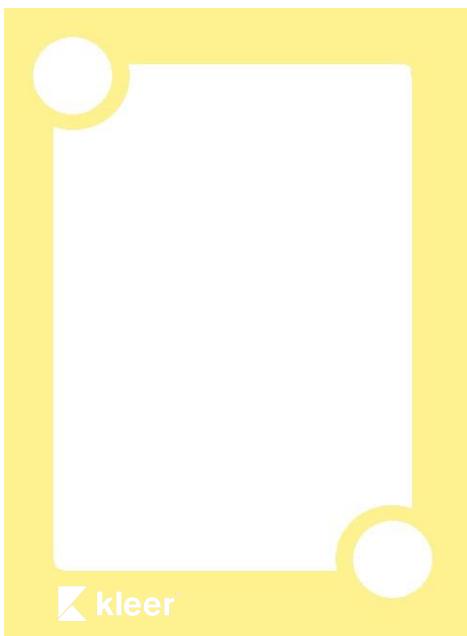
## L'information nécessaire a été communiquée au reste de l'équipe

Comme par exemple l'explication fonctionnelle de l'élément livré, certaines considérations techniques importantes, etc.

 kleer™

 kleer™

 kleer™



**kleer™**