## **Scrum exercice corrigé : les réponses**

Avant de passer aux réponses, voici comment vous situer :

**8/10 à 10/10** -> vous êtes un expert qui a décortiqué le scrum guide à fond.  
**5/10 à 8/10** -> très bon résultat car ce Scrum exercice corrigé était extrêmement dur.  
**3/10 à 5/10** -> je vous conseille de revoir le scrum guide car il vous manque encore quelques connaissances  
**0/10 à 3/10** -> regardez bien les réponses pour bien comprendre les subtilités qui font du scrum un framework très compliqué à mettre en place

A présent passons aux réponses et voyez si vous avez de bonnes réponses. Si ce n’est pas le cas, vous aurez des liens vers des articles pour vous expliquer.

### **1/ Question 1**

En scrum, qui valide les user-stories pour qu’elles passent de « test » à « done » ?

[ ] le scrum master  
[ ] le product owner  
[ X] l’équipe de développement  
[ ] toute l’équipe

Bien que cela se répand beaucoup car parait plus logique pour ceux qui viennent des méthodes traditionnelles, ce n’est pas le product owner qui teste (dans le sens vérifier le travail de l’équipe de développement) les user-stories avant de les déclarer « done. L’équipe de développement est 100% responsable de fournir des user-stories « done » potentiellement livrable.

### **2/ Question 2**

L’équipe de développement peut-être composée :

[X] de développeurs front et de développeurs back  
[X] de développeurs, d’un business analyst et d’un testeur  
[X] d’un testeur et de développeurs  
[X] de développeurs, d’un PMO et d’un testeur

Toutes les réponses sont bonnes. L’équipe de développement ne veut pas dire équipes de « développeurs » mais équipe de réalisation. Tous ceux qui ne sont pas « product owner » ou « scrum master » et qui participe à la réalisation du produit font parti de l’équipe de développement. Donc même si cela surprend, le Business Analyst ou le PMO si nécessaire (bien que ce rôle est peu aimé) font parti de l’équipe de développement.

Nous pouvons accepter une autre réponse : « aucune case cochée ». En effet, l’équipe de développement ne reconnait aucun titre individuel. Donc les bonnes réponses seraient :

[X] de compétences développeurs front et de compétences développeurs back  
[X] compétences développeurs, de compétences business analyst et des compétences de tests  
[X] de compétences développeurs et de compétences testeur  
[X] de compétences développeurs, de compétences PMO et de compétences tests

Oui c’est subtil.

### **3/ Question 3**

Le product backlog refinement est :

[X] le nouveau nom de la backlog grooming  
[ ] une cérémonie scrum  
[ ] la pratique d’affinage de user-stories et d’estimation en poker planning  
[ ] ce n’est pas officiel en scrum

Seule la première réponse est vraie. Le terme de grooming a une connotation « pédophile » et a donc été remplacé par « product backlog refinement ».

Le product backlog refinement n’est pas une cérémonie bien que ce soit souvent réalisé de cette façon. C’est à l’équipe de définir comment fonctionnera l’affinage des « items » du product backlog ; cela n’est pas forcément sous forme de cérémonies

Si le product backlog refinement est un moment d’affinage, cela concerne des « items ». Bien que la user-story soit une pratique ultra populaire, le scrum ne parle que d’items. C’est l’équipe qui décide de ce qui est un « item ». De plus l’estimation en poker planning est une pratique qui ne vient pas du scrum. Cette réponse était tendancieuse mais ne peut être coché car ce n’est pas comme cela qu’elle est définie.

### **4/ Question 4**

Quels artefacts existent en scrum ?

[X] le definition of done  
[ ] le definition of ready  
[X] le sprint backlog  
[ ] l’increment backlog

Le definition of ready est une pratique répandue mais non reconnue par le scrum guide. Je ne connais pas les raisons de cette absence mais cette pratique amène souvent à rendre le scrum « non agile ».

L’increment backlog n’existe pas ; il ne faut pas confondre avec l’artefact « increment ». L’increment est l’ensemble des éléments en « done » du sprint ajouté à l’incrément du sprint précédent. Il n’y a aucune notion de backlog dans cet artefact.

### **5/ Question 5**

Le product owner est responsable :

[ ] de valider que les items sont bien « done »  
[X] de la gestion du product backlog  
[X] que l’équipe de développement comprenne bien les items  
[ ] de rendre transparent le sprint backlog

Le product owner est responsable du product backlog et de sa compréhension. En revanche, il n’a aucune responsabilité sur le sprint backlog ; ce dernier est 100% sous la responsabilité de l’équipe de développement.

Le product owner n’est pas responsable des développements. Seule l’équipe de développement doit mettre en place des exigences de qualités et vérifier par elle même que ces critères sont respectés à 100% avant de mettre l’item en « done ».

### **6/ Question 6**

Le scrum master doit :

[X] avoir des connaissances avancées sur la gestion d’un product backlog  
[ ] faciliter les cérémonies scrum  
[X] travailler avec les autres scrum master de l’entreprise  
[X] coacher l’équipe de développement si nécessaire

Si vous avez coché la première case également, il est vrai que c’est tendancieux (mais c’est faux). Le scrum guide dit :

« S’assurer que le Product Owner sait comment organiser le Backlog produit pour maximiser la valeur ; »

Il dit qu’il doit s’assurer que le product owner sache organiser le backlog mais ne dit pas que c’est lui qui doit lui apprendre. Donc mot pour mot, rien n’interdit au scrum master de faire appel à un autre product owner ou à un coach agile pour aider sur cette partie.

Légende urbaine, le scrum master n’a pas à faciliter les cérémonies. Il ne le fait que si l’équipe fait appel à lui ou que cela est nécessaire.

### **7/ Question 7**

Quelles sont les phrases qui sont vraies autour du sprint review

[X] le product owner y invite l’équipe scrum et les principales parties prenantes  
[X] l’équipe discute des problèmes rencontrés et comment ils ont été résolus  
[X] l’équipe y démontre le travail « fini »  
[X] la revue des délais et des budgets doit être fait

Toutes les réponses sont bonnes, elles font parties de la liste de la sprint review proposée par le scrum guide.

### **8/ Question 8**

Le scrum :

[X] est un cadre de travail  
[ X ] est une méthode agile  
[ ] ne s’occupe pas de la maintenance applicative  
 [ X ] est un cadre léger

Scrum est un cadre de travail léger pour le développement, la livraison et la maintenance de produits complexes. Il n’est en aucun cas une méthode bien que c’est souvent comme cela qu’on le qualifie.

### **9/ Question 9**

Le scrum permet dans des cas exceptionnels d’annuler un sprint. Quelles phrases sont vraies ?

[X] l’équipe de développement peut demander au product owner d’annuler le sprint  
[X] seul le product owner est autorisé à annuler un sprint  
[ ] un sprint recommence la semaine suivante pour ne pas perturber les dates des cérémonies  
[ ] seul le scrum master est autorisé à annuler un sprint

En scrum, seul el product owner est autorisé à prendre la décision d’annuler un sprint (soit d’y mettre fin). Cependant l’équipe de développement peut parfaitement influencer pour aller vers cette annulation. En revanche après un travail de fermeture de ce sprint annulé, l’équipe recommence directement un nouveau sprint.

### **10/ Question 10**

Quelle(s) affirmation(s) sont vraies concernant la sprint planning ?

[] tous les items choisi en sprint planning ne sont pas forcément finalisés.  
[X] seule l’équipe de développement valide sa capacité de faire (le nombre d’items du sprint)  
[X] un sprint planning peut durer jusqu’à 8h  
[X] l’équipe de développement doit définir comment elle va réaliser les items

Toutes les réponses sont bonnes. Attention ! En aucun cas, les items doivent être 100% ready pour pouvoir être dans le « todo ». L’équipe de développement peut demander de nouveaux items nécessaires à l’atteinte de l’objectif du sprint ou accepter des items non finalisées qu’elle est sûre de pouvoir faire si ils sont finalisés en cours de sprint par le product owner. Le fait de dire que tous les « items » sont « ready » pour pouvoir être pris en sprint rend votre scrum non agile.