

# Challenge « 30 jours sur le Software Craftsmanship »

Par Christophe Moustier

1	<i>Donnez votre définition du Software Craftsmanship</i>
2	<i>Regardez une vidéo d'Uncle Bob sur Youtube</i>
3	<i>Lisez un article qui décrit le TDD sur la base du "Rouge/Vert/Refactor"</i>
4	<i>Tentez d'évaluer la proportion de temps que vous passez à lire du code vs écrire du code et ce qu'en disent les développeurs sur le net (essayez les mots clés "time spent to read code")</i>
5	<i>Retrouvez et lisez l'article " Naming as a Process" de Arlo Belshee</i>
6	<i>Trouvez les "3 lois du TDD" d'Uncle Bob</i>
7	<i>Allez sur le site <a href="https://sourcemaking.com/">https://sourcemaking.com/</a> et identifiez les "code smells" et les techniques de refactoring que vous connaissez"</i>
8	<i>Essayez de mettre en œuvre le TDD</i>
9	<i>Mettez sur votre liste d'achat un livre sur le Sw Craftsmanship tels que [Martin 2008], [Fowler 1999] ou [Kokaina 2019] (ce dernier est en français)</i>
10	<i>Listez tous les raccourcis claviers que vous connaissez sur votre IDE et apprenez en 3 de plus</i>
11	<i>Partagez votre situation sur ce challenge avec vos collègues et au moins un réseau social</i>
12	<i>Citez les principes SOLID</i>
13	<i>Mettez en œuvre un outil comme Mockito ou EasyMock ou son équivalent dans votre langage favori</i>
14	<i>Allez sur <a href="https://github.com/emilybache/Tennis-Refactoring-Kata">https://github.com/emilybache/Tennis-Refactoring-Kata</a>, téléchargez le kata de tennis et faites un refactoring de ce code en TDD</i>
15	<i>Trouvez un collègue et découvrez / pratiquez ensemble une session de Ping-pong Pair Programming</i>
16	<i>Lire le manifeste du Sw Craftsmanship (<a href="http://manifesto.softwarecraftsmanship.org/#/fr-fr">http://manifesto.softwarecraftsmanship.org/#/fr-fr</a>) et le signer</i>
17	<i>Identifier un code du projet sur lequel vous travaillez sur lequel le Sw Craftsmanship serait bénéfique</i>
18	<i>Essayer de faire un kata en mode TCR</i>
19	<i>Demander à vos collègues s'ils connaissent quelques katas et demandez à partager</i>

20	<i>Se renseigner sur la différence entre la couverture de code, la couverture de branche, la couverture décisionnelle</i>
21	<i>Lire le serment du développeur d'Uncle Bob (<a href="https://blog.cleancoder.com/uncle-bob/2015/11/18/TheProgrammersOath.html">https://blog.cleancoder.com/uncle-bob/2015/11/18/TheProgrammersOath.html</a>)</i>
22	<i>Se renseigner sur la notion de dette technique, partagez-là avec vos collègues</i>
23	<i>Identifier les différences entre ATDD, TDD et BDD</i>
24	<i>Se familiariser avec la programmation offensive et défensive</i>
25	<i>Identifier ce qu'il manquerait dans la définition du "terminé" de votre équipe scrum</i>
26	<i>Renseignez vous sur les tests par mutation et essayer les outils DESCARTES et DSPOT du projet STAMP hébergé sur GitHub</i>
27	<i>Construisez-vous (ou récupérez) une cheatlist sur le Sw Craftsmanship et partagez-la</i>
28	<i>Comprenez à quoi sert le design pattern PageObject et imaginez une version SOLID puis comparez-la au design pattern "ScreenPlay"</i>
29	<i>Organisez un dojo en mode mob programming et TDD avec le kata String Calculator (<a href="http://codingdojo.org/kata/StringCalculator/">http://codingdojo.org/kata/StringCalculator/</a>)</i>
30	<i>Venez mettre un commentaire sur LinkedIn sur ce challenge</i>

Ce challenge est sous licence CC-BY 4.0. Vous pouvez librement l'utiliser, à la condition de l'attribuer à l'auteur en citant son nom

<https://creativecommons.org/licenses/by/4.0/legalcode.fr>

