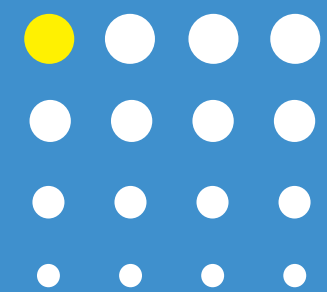
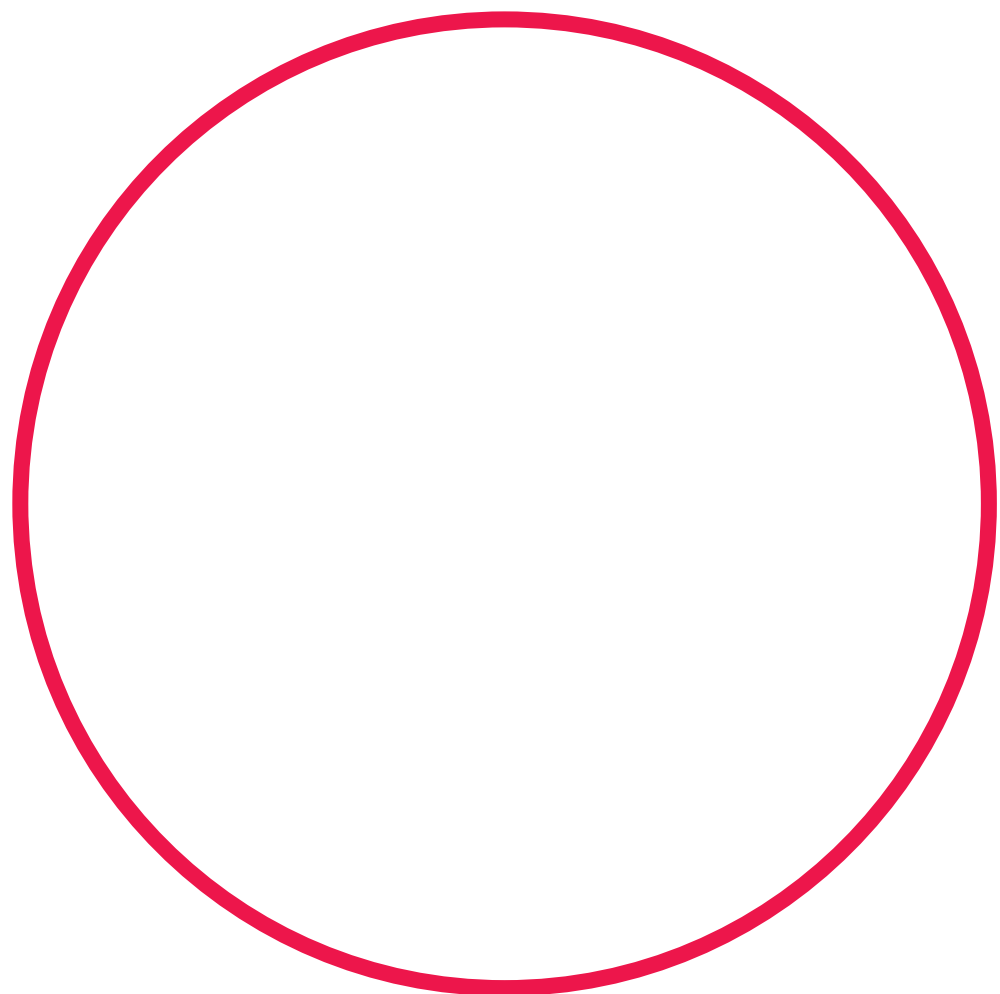


# SCRUM MASTER ACADEMY

Le guide du  
Scrum Master  
d'élite



# Préface

## Par Jeff Sutherland - Co-Créateur de Scrum

As Scrum enters the mainstream, Publicis Sapient France France has become one of the leading Scrum coaching companies for French developers. What they have noticed is that as Scrum expands, the basic ideas become watered down. For example, their Scrum Master Academy Rule #2, is that the Agile Manifesto is not a developers Bill of Rights.

In fact, the Agile Manifesto is not about the rights of a developer, it is about professionalism. This means

- (1) increasing your ability to help the team to a higher level of performance,
- (2) getting bug free software by the end of every sprint,
- (3) involving the customer, or the Product Owner as the representative of the customer, in all key decisions about development that will affect the end user, and
- (4) embracing change as fast as the customer can change his or her mind.

The inability of new agile developers to understand the basics has created significant problems. For example, over half of “agile” teams do not have working software at the end of the sprint. Somehow, they are not reading or understanding the Agile Manifesto.

As Mike Cron, All Blacks Scrum coach, says, “In the old days we used to just smash into something, maybe without quite knowing what we were doing. Here we are trying to get good technique and understand everything we are doing.” Scrum is just like golf, we have to learn how to swing. And it’s not so much about the rules of the game, it is how you play the game.

There are many pitfalls of Scrum to the new developer, and Scrum Master Academy has developed some basic techniques to avoid most of these pitfalls. If you want to be on a winning team in France, take a look at the Scrum Master Academy.



JEFF SUTHERLAND



# Table des matières

<b>Introduction</b>	<b>5</b>
<b>Parlons de valeurs</b>	<b>6</b>
Règle n°1 - Connais, comprends et incarne tes valeurs agiles	7
Règle n°2 - Le manifeste agile n'est pas la déclaration des droits du code du travail	8
Règle n°12 - On travaille toujours en équipe	9
Règle n°19 - No broken window	10
Règle n°21 - Évalue régulièrement où l'équipe en est de sa maîtrise de l'agile	11
<b>Parlons des user stories</b>	<b>12</b>
Règle n°4 - N'oublie jamais la valeur d'une story	13
Règle n°5 - Pas plus d'un sprint pour traiter une story	14
Règle n°6 - Il faut pouvoir prendre plus de 4 stories par sprint	15
Règle n°8 - À la fin du sprint, il y a 2 options pour une story : “done” ou “not done”	16
Règle n°9 - Tu n'estimeras pas jour/homme	17
<b>De l'efficacité des événements Scrum</b>	<b>18</b>
Règle n°3 - Le burndown est publié tous les jours après le daily stand up	19
Règle n°30 - Démon sans slides	21
Règle n°31 - Qui dit démo dit préparation	22
Règle n°34 - Tu connais et tu lis les stories avant le sprint planning	23
Règle n°15 - Choisir et varier les formats des rétrospectives	24
Règle n°25 - Se préparer et avoir sa malette de Scrum Master	25
<b>La résolution des obstacles</b>	<b>26</b>
Règle n°14 - Ne te laisse pas enterrer sous les problèmes	27
Règle n°20 - Il n'y a pas de “eux”, il y a “nous”	28
Règle n°22 - Connais tes détracteurs	29
Règle n°26 - Explorer le chemin des possibles	30
Règle n°27 - Il vaut mieux demander pardon qu'obtenir la permission	31
Règle n°35 - Un Scrum Master sur le carreau est inutile	32

<b>L'attitude et le charisme</b>	<b>33</b>
Règle n°7 - L'énergie de l'équipe c'est toi qui l'apporte	34
Règle n°10 - Dans une réunion, ta place est debout à côté du tableau blanc	36
Règle n°18 - Une équipe autogérée nécessite du Leadership	37
Règle n°29 - Tu es un facilitateur par un dictateur	38
Règle n°13 - Reste à la page	39
Règle n°22 - Évite le blâme et la complaisance	40
Règle n°16 - Tu portes la transparence	41
Règle n°23- Tais-toi et SURTOUT écoute	42
<b>Allez plus loin</b>	<b>43</b>
Règle n°24 - Veille à la pluralité des compétences de l'équipe	44
Règle n°28 - Basez-vous toujours sur les faits et leur impacts	45
Règle n°11 - Ne deviens pas un garagiste	46
Règle n°33 - Tu déploies tous les jours en environnement de test	47
<b>Règle de crise</b>	<b>48</b>
Règle n°17 - Transparence et pas indécence	49
Règle n°40 - Une bonne prédictibilité vient d'une bonne mesure de la vitesse et d'une bonne estimation de l'effort de chaque story	50
Règle n°41 - Gère les pollueurs en revue	51
Règle n°42 - Range ton désordre	52
Règle n°43 - Pas de sprint de refactoring	53
Règle n°44 - Des collaborateurs et non pas des ressources	54
Règle n°45 - À la rétrospective, 2 actions d'amélioration maximum	55
Règle n°46 - Pour garder la force, la motivation tu entretiendras	56
<b>Vos règles du jeu</b>	<b>57</b>
<b>Bibliographie</b>	<b>63</b>
<b>Remerciement et conclusion</b>	<b>65</b>

# Introduction

Ce chapitre explique la genèse et les principes fondateurs de la Scrum Master Academy. L'idée de départ est venue de Gilles Mantel et Jean-Laurent de Morlhon début 2012 alors qu'ils cherchaient un moyen de transmettre son expérience agile à ses collègues dans un format pédagogique, sans écrire un livre de 500 pages et sans donner de certification formelle.

Nous avons acquis de nombreux réflexes agiles "sur le tas" en nous confrontant aux difficultés de certaines situations et sans spécialement lire de livres. L'expérience s'acquiert en faisant des erreurs, en en tirant des enseignements. Les bons réflexes se construisent sur la base d'un vécu. Pour autant, est-il souhaitable de laisser vos collègues se tromper si vous avez de bons conseils à leur donner ? Nous avons constaté à de nombreuses reprises que les personnes fraîchement issues de formations Scrum (certifiantes ou non) font une application naïve, intégriste ou pas assez éclairée des méthodes agiles. Nous avons aussi remarqué que nos entorses à certaines méthodes agiles étaient parfois perçues comme des actes de crime contre l'agilité. Nous avons donc voulu transmettre un message pragmatique dans un format pédagogique et ludique. Mais comment diffuser cette expérience et la rendre accessible aux moins expérimentés sans réinventer la roue ?

Comme souvent dans ce genre de situation, la réponse est là où on ne l'attend pas : sur notre canapé devant la télévision. Pour l'occasion, c'est la série américaine NCIS qui nous a inspirés. Dans cette série, le chef d'une équipe d'enquêteurs sur des affaires criminelles de la marine américaine a créé une liste de 50 règles d'enquêtes basées sur son

expérience du terrain. Il considère que son métier est de les enseigner à ses équipiers, au fil des différentes situations, afin qu'ils deviennent d'excellents enquêteurs. Ces règles ont la particularité de n'être écrites nulle part : elles forment une référence commune à tous ses équipiers et contribuent à renforcer l'esprit de groupe.

Gilles Mantel et Jean-Laurent de Morlhon ont travaillé d'arrache-pied (le mot est faible) pour rassembler leurs expériences agiles depuis 10 ans et définir un ensemble de règles "à la NCIS". Puis ont présenté le fruit de leur travail, challengé par leurs pairs, au Scrum Day 2012.

Une décennie plus tard, Jean-Noël Hany, Alexandre Kuoch, Xavier Spinosi, Thomas Cousin, Gaëlle Mouraret, Olivier Marquet et Morgane Eckert vous proposent une version mise à jour de ce guide.

Toutes les règles s'adressent aux Scrum Masters, elles sont donc souvent formulées au présent de l'impératif. Bien sûr, n'importe quel membre de l'équipe peut s'approprier ces règles mais le Scrum Master est la pierre angulaire de la bonne application de Scrum. Il est garant de la qualité du cadre de travail, même s'il n'en est pas le dépositaire unique. Nous avons donc adressé cette liste de règles à ce dernier.

Les règles ne sont pas classées dans un ordre précis, nous les avons simplement listées dans l'ordre de notre réflexion.

# Parlons de valeurs

Sur la dernière décennie, l'**Agilité** dans le monde est devenue une réalité pour de nombreuses équipes. La courbe d'adoption de l'agilité a franchi le point critique au-delà duquel le sujet devient un courant dominant.

À titre d'exemple : Les enquêtes régulières State Of Agile\*, menées par l'éditeur de logiciel Digital.ai, nous éclairent sur la profondeur de cette adoption au niveau mondial. Nous n'en sommes plus au stade des expériences d'early adopters mais à la généralisation des gros programmes à l'échelle dominés par le framework SAFe, n'en déplaise aux détracteurs de l'agilité.

Cette popularisation entraîne inévitablement une dilution des valeurs et des principes fondateurs liée à l'effet de masse. Les agilistes de la première heure y voient, souvent à juste titre, une perversion du modèle.

Dans la majorité des cas, cette dérive est portée par des organisations soucieuses de rester dans la course ou simplement par effet de mode/mainstream. Mais parfois, elles ne cherchent pas vraiment à changer la culture établie.

L'Agilité promeut le travail en collaboration, le respect, le partage, le développement incrémental, l'amélioration constante, la fin de la hiérarchie et des silos ou encore la culture du feedback. Beaucoup de bouleversements qui rencontrent une certaine résistance, nécessitent de donner le temps au changement de s'opérer dans les esprits et dans les faits.

Ce chapitre vous présente un ensemble de règles pour ne pas perdre de vue l'essence de l'agilité.



## Règle n° 1 Connais, comprends et incarne tes valeurs agiles

Par où commencer ? Souvent, beaucoup vous répondront "par le début !". C'est la question que nous nous sommes posés : "Le début, c'est quoi ?". Le début, c'est la fondation, la règle de référence.

Nous allons donc enfoncer une porte ouverte, qui sera sûrement une évidence pour vous. Connaissez-vous bien le Manifeste Agile ? Êtes-vous capable de citer, les 4 valeurs et les 12 principes ?

Il ne faut pas négliger les ressources que peut apporter le manifeste agile. Certaines solutions à des problèmes rencontrés quotidiennement pourront y être trouvées.

Nous autres agilistes faisons des compromis avec le framework agile utilisé (Scrum, XP, SAFe, Nexus, ...). Il ne peut pas être appliqué au pied de la lettre. Pour autant, le manifeste agile nous permet de

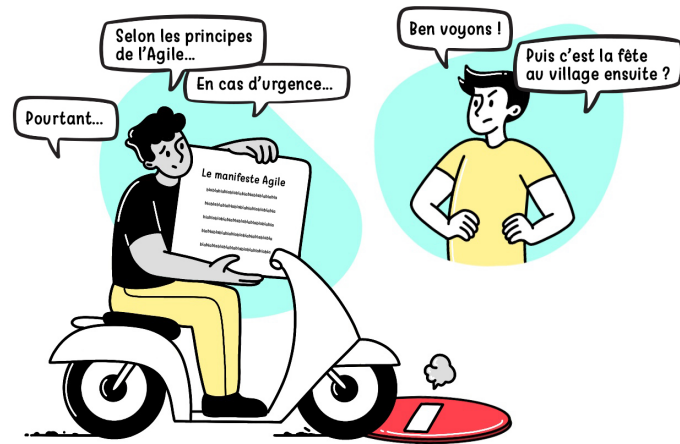
garder un certain bon sens dans son application afin d'assurer le succès du produit. Et la finalité dans tout ça ? Nous devons rester cohérent avec les valeurs. Par exemple, L'acharnement du suivi d'un processus, même agile, ne sert pas à grand-chose si nous ne nous interrogeons pas sur le fond de la démarche : à quoi sert-il ? Quels problèmes résout-il ? Est-ce que les individus qui l'utilisent ont les bonnes interactions ?

Autre exemple : est-ce que votre équipe (personnes du métier et développeurs) est bien co-localisée dans votre entreprise ?

Beaucoup de ce qui sera écrit plus loin ne sera sûrement qu'un nouvel écho du manifeste agile. Si vous devez ressortir de cette lecture avec 1 règle en tête, c'est la n°1. Relisez le manifeste agile et les 12 principes qui le complètent.

\* <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/>

1- <https://blog.engineering.publicissapient.fr/2014/03/28/le-manifeste-agile-est-lopium-du-peuple/>



## Règle n°2

### Le manifeste agile n'est pas la déclaration des droits du code du travail

Qui n'a jamais rencontré un équipier brandissant l'agilité pour faire valoir ses droits ? Quelques exemples :

- ne pas faire d'heures supplémentaires (même lorsque le projet est en pleine tourmente),
- brandir le manifeste agile pour dire "non" au sponsor (quand il n'est pas d'accord avec nous),
- où encore le fameux "j'ai le droit de choisir mes tâches, c'est dit dans le manifeste" et ne choisir que les tâches intéressantes.

Les exemples sont peut-être exagérés (mais finalement pas tant que ça). Ces comportements qui détournent le manifeste agile ternissent l'image de l'agilité plutôt que de la promouvoir et dessert le succès du cadre de travail.

Le manifeste agile n'est pas la déclaration des droits du code du travail. Bien sûr, l'agilité embarque des pratiques plus humaines tant sur le plan organisationnel que managérial : Radical Management, Cynefin Model, Management 3.0, etc. Certains même vont jusqu'à créer de nouveaux mouvements à part entière, tel que STOOS.

Pour autant, cette démarche, bien que louable, ne doit pas faire oublier la courbe d'apprentissage pour maîtriser ces concepts. Apprentissage nécessaire, voire fondamental.

Avant de courir, il faut savoir marcher. Pour un Scrum Master débutant, il faut penser simple. Simple dans son approche et dans son application de l'agilité, quel que soit le framework. Dans le cas contraire, vous pourriez mettre en péril ce que vous essayez de mettre en place.

Le revers de la médaille pourrait être douloureux. Même si l'agilité permet moult choses, elle met aussi en lumière beaucoup plus rapidement les problèmes. Ils porteront des noms variés : "course à la productivité", "manque de budget", "indisponibilité", "visions divergentes", "culture d'entreprise", "décision des actionnaires", etc.

Les réponses à ces questions sont comme la vérité : ailleurs. En tout cas ailleurs que dans le manifeste agile.

Pour résumer cette règle, c'est celle du bon sens et du discernement lors de l'utilisation du mot "agile".

## Règle n°12

### On travaille toujours en équipe



verre, il est aussi possible d'organiser un déjeuner commun si les membres de l'équipe n'ont pas l'habitude de se retrouver pour manger. Ce n'est pas tout, on constate souvent qu'une équipe qui démarre avec Scrum passe par un état d'euphorie lors de son premier sprint. On voit souvent des post-its fleurir à la première rétrospective expliquant combien il est agréable de travailler avec cette équipe.

Souvent cette belle euphorie est fragile et elle a tendance à voler en éclat avec les premières difficultés (c'est à dire parfois dès le sprint 2).

Aujourd'hui, il vaut mieux prendre un peu plus de temps à la première rétrospective pour creuser les raisons de cette euphorie : pourquoi avez-vous apprécié travailler avec cette équipe ? Vous obtiendrez généralement quelques regards vides ou des réponses triviales du type "parce que c'est bien". Il faut alors prendre quelques exemples concrets pour commencer à avoir des éléments intéressants : à quel moment as-tu le plus senti cette osmose ? Pourquoi ? On commence alors à identifier les raisons : "on n'a pas hésité à s'aider", "on a dépassé le cadre de notre rôle", "il n'y avait pas de responsabilités individuelles", "Bruno a ramené les croissants", "on a mangé ensemble", "je n'ai pas eu peur de demander de l'aide", "on est au courant de ce que font les autres", "Estelle a passé du temps à m'expliquer le schéma de la base", etc. Notez alors soigneusement ces éléments sur des post-its et affichez-les ostensiblement dans le bureau, ce sera la "Charte d'Equipe" pour les sprints à venir.

L'équipe est l'élément de base de la réussite d'un projet agile. Assurez-vous que les décisions soient partagées par tous et que personne ne parte en franc tireur sous prétexte "d'aller plus vite". Cette unité d'équipe est difficile à maintenir au cours du temps. Les plus cultivés d'entre vous doivent connaître le modèle de Tuckman qui décrit les différents états par lesquels passe une équipe : forming, norming, storming, performing<sup>2</sup>. L'idée n'est pas ici de vous distiller la théorie, mais de vous donner des éléments concrets.

Il existe de nombreuses façons de créer un esprit d'équipe (de manière colocalisée ou à distance). Une façon assez naturelle de créer du lien est en allant boire un verre avec toute l'équipe (donc y compris le Product Owner) à la fin du premier sprint, ou à toute autre occasion. On discute dans un environnement différent et les informations ne sont pas les mêmes, on parle de nos enfants, de nos expériences similaires, on développe une sympathie pour l'autre, ce qui sera un atout non négligeable dans les coups durs. Avant même ce

<sup>2</sup> [https://en.wikipedia.org/wiki/Tuckman%27s\\_stages\\_of\\_group\\_development](https://en.wikipedia.org/wiki/Tuckman%27s_stages_of_group_development)



## Règle n° 19

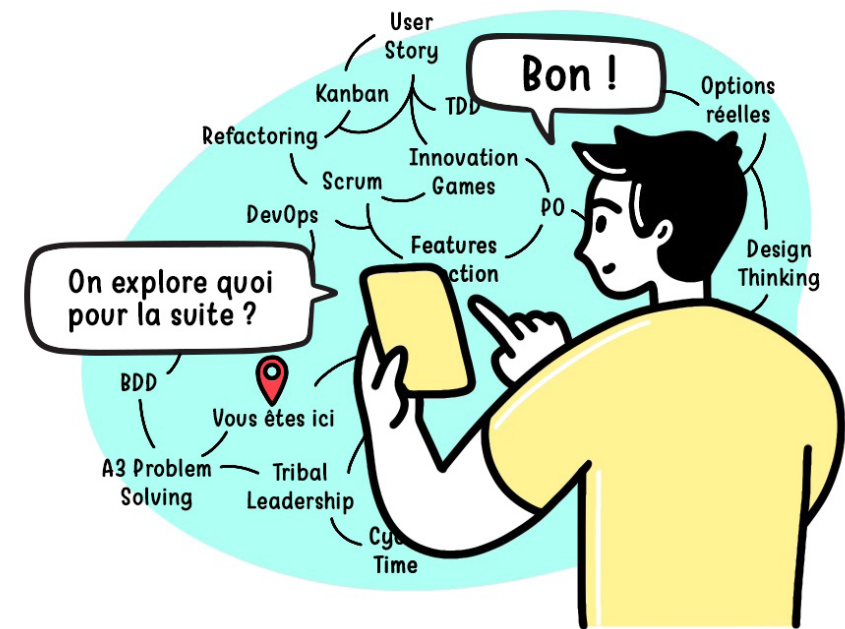
### No broken window

La théorie de la fenêtre cassée<sup>3</sup> est issue d'expérimentations sociales datant du milieu du XX<sup>e</sup> siècle.

Ce principe fût utilisé par le Maire de New York dans les années 90 pour améliorer les conditions de sécurité dans les quartiers difficiles en faisant le lien entre nombre de fenêtres brisées et le taux

de criminalité. Bien que les conclusions de ces expérimentations soient parfois remises en cause, l'application de ce principe possède des vertus indiscutables. Parce qu'un écart toléré, ou un bug non corrigé, entérine le comportement, assurez-vous de toujours recadrer les pratiques ou corriger immédiatement les anomalies qui trainent.

3- [https://fr.wikipedia.org/wiki/Hypoth%C3%A8se\\_de\\_la\\_vitre\\_bris%C3%A9e](https://fr.wikipedia.org/wiki/Hypoth%C3%A8se_de_la_vitre_bris%C3%A9e)



## Règle n° 21

### Évalue régulièrement où l'équipe en est de sa maîtrise de l'agile

Ce n'est pas parce qu'on applique Scrum que l'on a fini sa progression vers l'agilité ou vers l'incarnation des valeurs de Scrum. L'agilité est un état d'esprit plus qu'un but, alors regardez le chemin parcouru et évaluez régulièrement celui qu'il reste à parcourir. Il existe de nombreux modèles d'évaluation de sa maturité agile qui vous permettront de vous situer sur ce chemin.

Pour cela, vous pouvez vous appuyer sur une checklist orientée Scrum mise au point par Henrik Kniberg et utilisée par Jeff Sutherland lui-même<sup>4</sup>, ou encore un questionnaire d'enquête mis au point par Dean Leffingwell<sup>5</sup>. L'idée ici n'est pas d'utiliser un seul support mais de les varier pour avoir différents points de vue. Il n'y a pas une seule agilité, mais toutes les agilités\* peuvent vous apprendre des choses.

4- <https://www.crisp.se/gratis-material-och-guider/scrum-checklist>

5- <https://scalingsoftwareagility.wordpress.com/2008/02/01/resource-agile-process-self-assessment-survey/>

# Parlons des user stories

Les 5 règles de ce chapitre se focalisent sur un élément qui s'avère paradoxalement souvent mal maîtrisé dans le cycle de vie des produits Agiles : les user stories. Ce n'est pas à proprement parler un élément du cadre de travail Scrum, cette pratique nous vient de l'eXtreme Programming, mais elle est largement utilisée dans le cadre de projets menés en Scrum au point de devenir la référence de nombreuses formations certifiantes. Les user stories paraissent, à première vue, simples et à la portée de tous, c'est d'ailleurs exactement ce qu'on leur demande : simplicité et concision. Si vous n'êtes pas encore familier avec la notion de user story, on en trouve de nombreux exemples sur le Web, la forme la plus populaire étant le modèle :

“En tant que rôle, je veux *faire quelque chose* dans le but *d'obtenir de la valeur*.”

Mais même ce modèle simpliste peut être mal utilisé, voilà quelques difficultés que l'on peut rencontrer :

- **Faut-il exprimer un besoin ou une solution ?** On constate que plus la personne qui rédige la story est éloignée du développement informatique, plus la story est exprimée en termes de besoin à un niveau abstrait. De même, lorsque la dimension exploratoire du produit développé est très forte, on a tendance à formuler des besoins plus que des solutions. La réponse n'est pas si claire et chacun ira de sa recommandation en fonction de son vécu.
- **Comment mesurer la valeur métier ?** C'est souvent difficile à mesurer voire intangible et, par conséquent, on se retrouve parfois avec des stories de ce genre : “En tant qu'utilisateur je veux supprimer un enregistrement dans le but d'effacer un enregistrement.”

- **Détournement du modèle pour exprimer une story technique :** parfois le Product Owner n'exprime pas des stories qui paraissent fondamentales pour l'équipe de développement, ces derniers glissent donc dans le backlog des stories de ce genre : En tant que développeur je veux avoir un outil d'intégration continue dans le but de mieux faire mon travail.

On aime la définition suivante :

*Une user story est une invitation à avoir une conversation avec son client sur un sujet particulier.*

Cette définition a le mérite de recadrer l'utilisation de stories techniques : Quelle genre de discussion allez-vous avoir avec votre client sur la plateforme d'intégration continue ou la migration du modèle de données ? Elle a aussi le mérite de préciser l'objectif de la user story : la conversation. Ce n'est donc ni un besoin, ni une solution, c'est une conversation. Ces caractéristiques se retrouvent dans les caractéristiques des 3C (carte, conversation et confirmation), qui sont les références à garder à l'esprit pour exprimer de bonnes user stories.

Toutes ces difficultés décrites plus haut nous ont amené à exprimer les 5 règles suivantes, qui globalement tendent toutes vers le même objectif : une bonne collaboration entre l'équipe technique et l'équipe produit.



## Règle n° 4

### N'oublie jamais la valeur d'une story

Cette règle vise 3 objectifs pour le Scrum Master : assister le Product Owner pour la gestion de son backlog, éviter l'inondation de stories techniques dans le backlog et guider l'équipe lors du découpage de stories trop grosses. La partie “dans le but de...” est souvent lésée lors de l'expression des stories. Soit parce que la valeur nous paraît (trop) évidente, soit parce qu'à l'inverse, la valeur est très difficile à estimer en argent sonnante et trébuchante. Dans les 2 cas, on peut appliquer quelques astuces simples pour estimer la valeur.

D'une part, on peut considérer que la valeur d'une story est proportionnelle au nombre de parties prenantes qui la demandent, qu'ils soient utilisateurs, managers, commerciaux, dirigeants

etc. On peut ainsi suggérer à son Product Owner d'inviter les différentes parties prenantes à faire un poker planning de business points ou bien à participer à un atelier Buy-a-Feature.

D'autre part, on peut également estimer la valeur d'une story en terme de risque de ne pas faire ou en mesurant le coût du délai<sup>7</sup>. Ces dernières s'appliquent souvent pour les stories qui découlent d'obligations réglementaires, mais on peut l'étendre à n'importe quel type de stories. Dans le framework SAFe, WSJF<sup>8</sup> (weighted shortest job first) est une technique de priorisation par la valeur qui tient compte de deux paramètres : l'effort nécessaire à accomplir le travail et le coût du délai.

7- coût du délai : <https://blog.engineering.publicissapient.fr/2014/10/23/revue-de-presse-xebia-2014-43/#RevueDePresseXebia-Unquantifiedcostofdelayinpriorization>  
<https://blog.engineering.publicissapient.fr/2014/08/20/les-differents-usages-de-buy-a-feature/?msclkid=e0a66d28a54611ecbc3551815794f8b6>

8- WSJF en SAFe : <https://www.scaledagileframework.com/wsjf/>





## Règle n° 5

### Pas plus d'un sprint pour traiter une story

Cette règle est un simple rappel de ce que l'on enseigne classiquement dans un cours Scrum ou une formation agile. Il nous paraît cependant nécessaire de le rappeler car ce n'est pas forcément toujours bien appliqué.

On pourrait en fait reformuler cette règle de la manière suivante : si la story est trop grosse pour être développée en un seul sprint, ce n'est pas une story. L'intérêt de cette règle est surtout ce qu'elle sous-entend : le découpage des stories trop grosses. Une Epic (ou une feature / capability dans le frameworks SAFe) ne peut pas être traitée comme telle par l'équipe de développement, il faut la découper en unités plus petites qui vont rentrer

totalemment dans un sprint. Vous trouverez sur le site de SAFe, le framework à l'échelle de la Scale Agile, un exemple de découpage et ses différentes granularités<sup>9</sup>.

On se retrouve parfois confronté à la situation où le PO, ou l'équipe, considèrent qu'on ne peut pas découper plus finement certaines stories et découpent donc la story en tâches techniques planifiées sur plusieurs sprints. Pourtant, il existe de nombreuses stratégies (comme présenté ici avec cette page pour aller plus loin sur le sujet<sup>10</sup>), pour descendre à un niveau encore plus fin sans tomber dans le découpage technique (se reporter à la règle n°4 si vous êtes tentés par l'expérience).



## Règle n° 6

### Il faut pouvoir prendre plus de 4 stories par sprint

Vous ne trouverez sans doute pas mention de cette troisième règle dans la documentation Scrum classique. C'est une règle de routards de l'agilité, issue de l'expérience.

Cette règle va encore plus loin que la précédente dans la mesure où elle nous demande de découper les stories à un grain suffisamment fin pour en prendre plus de 4 dans un sprint. En allant même plus loin, il faut qu'un développeur puisse travailler sur plus d'une story pendant le sprint. Si une story occupe un développeur sur la totalité du sprint, il se crée un effet tunnel qui risque de mettre le sprint en péril. De plus, si celle-ci n'est pas démontrée lors de la revue, elle ne bénéficiera pas du feedback des utilisateurs.

Les estimations c'est un peu le jeu des devinettes. Nous savons qu'elles sont globalement vraies mais précisément fausses. Cette marge d'erreur peut conduire à ne pas pouvoir terminer une story dans le sprint où nous la planifions. Si vous ne prenez que 4 stories dans un sprint et que vous n'en terminez pas une, c'est le quart de l'incrément qui n'est pas réalisé. Ce qui est relativement important, à plus forte raison si la story est de taille importante. N'hésitez pas à redécouper vos grosses stories, même si elles tiennent dans un sprint. Cela facilitera le suivi, les tests et le niveau de confiance sur l'atteinte des objectifs du sprint.

<sup>9</sup>- SAFe, exemple de découpage : <https://www.scaledagileframework.com/features-and-capabilities/>

<sup>10</sup>- Pourquoi découper vos besoins en petite user stories : <https://blog.engineering.publicissapient.fr/2015/12/22/pourquoi-decouper-vos-besoins-en-petites-user-stories/>

## Règle n° 8

### À la fin du sprint, il y a 2 options pour une story : "done" ou "not done"

Il s'agit encore d'une règle évidente qui est enseignée dans de nombreuses formations agiles. Et pourtant, les tentations d'y déroger sont nombreuses : "il ne reste plus qu'à intégrer", "on y est presque", "on n'a pas testé mais j'ai confiance", "on n'a qu'à dire que c'est done et on ajoutera des tâches techniques sur le sprint suivant".

Toutes ces tentatives de passage en force mettent à mal la Définition du Done (DoD) qui est pourtant un élément essentiel de confiance entre le PO et l'équipe.

La transparence, premier pilier de Scrum permet la mise en place du cercle vertueux d'inspection et d'adaptation. Se voiler la face avec des statuts intermédiaires au "done" en fin de sprint, dénature Scrum et ses bienfaits.

La User story terminée est également un élément de mesure dans Scrum car elle contribue au calcul de la vélocité. Mettre à mal sa DoD est une promesse de complications à court terme. Que faire alors avec ces stories "not done" ? Normalement, on les planifie sur le sprint suivant, mais faut-il reporter l'intégralité des points alors que la moitié du travail est déjà fait ?

On vous propose de fonctionner ainsi :

Reportez totalement les points, quel que soit l'état d'avancement des stories concernées car :

- Bien souvent, celles-ci avaient de toute façon été sous-estimées.
- Si vous ne reportez pas les points, vous risquez à nouveau de sous-estimer le contenu du sprint suivant et donc, de vous retrouver à nouveau avec des stories non terminées. Votre vélocité ne



sera alors plus significative.

- Les projections ne signifient alors plus rien, donc le PO ne comprend plus l'avancement et tout cela risque de se finir en un vaste règlement de comptes.

Pour résumer : ne comptez pas une story "not done" dans la vélocité constatée d'un sprint. Comptez la entièrement dans les points prévus pour le sprint suivant. Enfin, le total de points comprenant la story reportée, doit toujours être cohérent avec la vélocité constatée des derniers sprints.

Cette méthode peut paraître un peu radicale (ou inutile pour les détracteurs de la vélocité), mais si vous appliquez bien la règle n°6, l'impact ne sera jamais vraiment significatif car toutes vos stories prises dans le sprint sont suffisamment petites pour éviter les grosses variations de vélocité.

Lien : <https://blog.engineering.publicissapient.fr/2020/10/15/une-definition-of-ready-dor-ou-une-definition-of-done-dad-en-30-minutes/>



## Règle n° 9

### Tu n'estimeras pas en jour/homme

Utilisez les story points<sup>11</sup> ou tout autre échelle relative pour vos estimations sur les éléments du Product Backlog. Vous gagnerez énormément de temps et vous obtiendrez des estimations plus fiables. Une estimation relative se fait par comparaison avec une unité de travail étalonnée. Pour initier vos stories étalon, à la fin de votre premier sprint demandez à votre équipe d'attribuer des points aux stories qui ont été terminées en utilisant la suite de Fibonacci. La plus petite sera l'étalon de base avec 1 point. Les autres seront mesurées comparativement à l'étalon 2, 3, 5 voire 8 ou 13 points. Gardez à l'esprit qu'une story doit être traitée dans un unique sprint.

N'utilisez pas les jours/hommes pour les estimations, ils sont trop rapidement mal interprétés et amènent à des erreurs souvent bien plus importantes. Un expert du sujet traitera certainement une même story bien plus rapidement que la personne qui vient de rejoindre l'équipe. Au-delà des stories, évitez le chiffrage des tâches en heures, ce travail n'apporte que peu de valeur et dérape trop souvent dans la durée.

11 - Estimer en story points : <https://blog.engineering.publicissapient.fr/2018/05/23/estimations-et-story-points/>

# De l'efficacité des événements Scrum

Dans ce chapitre, nous allons revenir sur les bases de l'activité du Scrum Master. Peut-être avez-vous adopté Scrum depuis quelques temps et cela a donné de bons résultats, mais progressivement les pratiques se sont diluées, le momentum du début est retombé, vos post-its sont moches (si si, avouez-le, et de toute façon vous n'en avez plus), et votre tableau d'obstacles n'est plus à jour depuis longtemps. Et de fait votre priorité n'est plus de faire du produit en Scrum mais de faire du logiciel tout court. Comme le disent souvent les rugbymen "no scrum no win" et pourtant vous êtes bien meilleurs que des rugbymen avec un clavier dans les mains. Alors que se passe-t-il ? C'est le moment de revenir aux fondamentaux !



## Règle n°3

Le burndown est publié tous les jours après le daily stand up

Scrum en tant que framework agile prône du logiciel opérationnel comme seul indicateur d'avancement mais le burndown permet d'afficher des éléments factuels pour montrer l'avancement avant la revue de sprint.

C'est un indicateur facile à produire et à mettre à jour : il n'y a pas de formule compliquée, la collecte des informations est triviale, et son tracé peut se faire à main levée. Nous imaginons bien que les créateurs de Scrum ont voulu garder une bonne balance entre simplicité et transparence maximum. Paradoxalement, nous constatons trop souvent l'absence de burndown affiché sur le mur des équipes agiles (ou sur les boards virtuels dans des contextes distribués). Quand nous posons la question, les explications tombent souvent dans l'une des deux catégories suivantes.

### 1. "LE BURNDOWN EST DANS JIRA (OU VERSIONONE, OU ...), IL SUFFIT D'ALLER LE CONSULTER"

Les outils électroniques ont un certain nombre d'avantage :

- Automatiser certaines tâches fastidieuses.
- Un meilleur partage des informations dans un mode distribué.
- Contenter les amateurs de traçabilité.

En revanche,

- Ils dissimulent l'information car ils nécessitent l'accès à un ordinateur et des comptes utilisateurs.
- Ils sont en général peu souples sur le calcul des indicateurs (attention à l'ajout de tâche en cours de sprint).
- Son affichage a le grand mérite de générer de la discussion.



Si vous êtes sur site, il suffit d'imprimer le graphique tous les jours et de le scotcher bien en vue (ou tracer quotidiennement la courbe au crayon sur une seule feuille de papier).

Sinon dans le cas où l'équipe est à distance (ce qui est de plus en plus le cas depuis l'année 2020), il vous faut : soit être très rigoureux sur la gestion du sprint dans votre outil (initialisation et maintiens), soit le tracer vous-même via des tableaux ou d'autres outils de collaboration.

## 2. "ÇA NE SERT À RIEN, ON SAIT OÙ ON EN EST"

Nous remarquons deux gros problèmes avec cette explication :

- Partir du principe que le burndown ne sert qu'à l'équipe de réalisation n'est pas correct. Le burndown est un élément essentiel de transparence dans, ET en dehors de l'équipe. Un burndown à jour et bien en vue évite un questionnement récurrent de la part des parties prenantes et participe à instaurer une confiance entre l'équipe et son environnement.
- Un manque de discipline sur l'estimation en tâches ou en points. Le burndown ne joue pas son rôle de révélateur du déroulement du sprint. Une absence de celui-ci peut même être caractéristique d'une organisation défaillante.

Pour vous aider à analyser ou faire analyser le/les burndowns vous pouvez utiliser "l'électrocardiogramme agile". Cet atelier vous permet d'étudier, de trouver les symptômes et les solutions sur le déroulé de votre burndown.



## Règle n° 30

### Démos sans slides

La première dérive observée des événements Scrum est souvent d'utiliser des présentations sur slides pour démontrer l'incrément produit dans la revue de sprint. L'objectif de la démonstration est bien de montrer du logiciel opérationnel, bannissez alors les slides et préparez des scénarios d'utilisation réels avec les données adéquates ou encore laisser les parties prenantes ou utilisateurs finaux manipuler le produit à votre place. N'hésitez pas à vous inspirer des techniques de storytelling pour mettre en scène votre incrément produit.

En admettant que nous soyons en train de développer un nouveau système de navigation pour l'automobile et que nous venons de livrer la fonctionnalité "recherche vocale de l'adresse", voici un exemple de Storytelling :

"Vous êtes samedi matin, 8h30, vous partez pour un petit week-end amoureux en Normandie. Vous avez un objectif : le Mont St Michel, comment faites vous ?"

On pose un problème, et on observe le comportement des utilisateurs dans leur résolution de ce problème via notre feature. Vont-ils échanger

comme on l'a imaginé ? Vont-ils vouloir préciser "je ne veux pas passer par des péages", ou encore "je désirerais faire une étape par le restaurant étoilé de Jean-Luc Tartarin". On cherche à comprendre "pourquoi vous avez fait ceci ?" et on récupère les feedbacks. Toutes les résolutions de problèmes vont permettre à l'utilisateur d'avancer dans son parcours et, fur et à mesure, il va rencontrer de nouveaux problèmes. La revue de sprint est là pour valider la solution proposée, mais aussi capter les nouveaux besoins et ajuster le backlog. Votre "scénario" doit être suffisamment immersif pour faire plonger l'utilisateur dans un environnement familier. Faites qu'il soit accessible, court pour être compréhensif et non frustrant. Vous aurez de meilleurs retours. Faites preuve de liberté et ne conditionnez pas le choix des utilisateurs. Ne les guidez pas, laissez-les explorer.

Souvenez-vous qu'une belle revue de sprint, fluide et bien organisée, laissera une impression positive et un sentiment de sérieux qui vous servira dans les coups durs.

Lien : <https://openseriousgames.org/electrocardiogramme-agile/>

Vidéo : <https://www.youtube.com/watch?v=qAJ-tsJD-0>



## Règle n°31

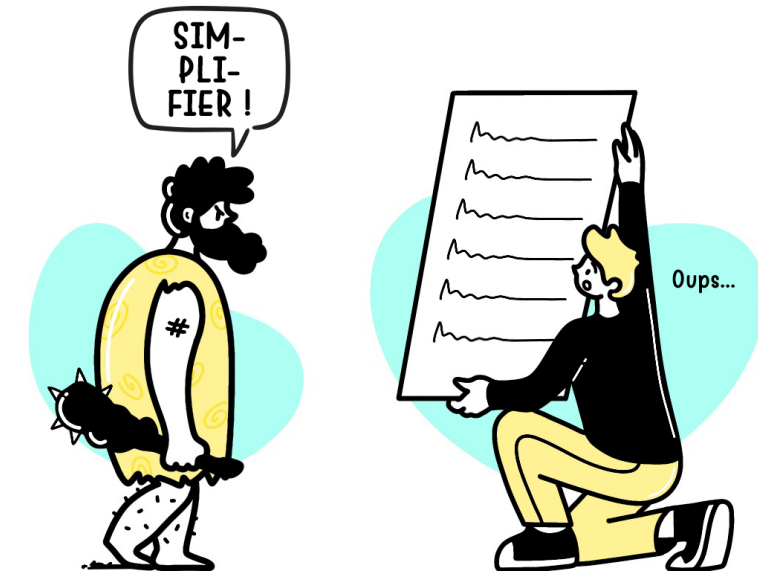
### Qui dit démo dit préparation

Nous sous-estimons régulièrement les détails logistiques pour organiser la démonstration de fin de sprint. Une heure avant le début de la réunion, assurez-vous personnellement que tout soit en place. Si vous prenez une salle de réunion, réservez-la 15 min avant l'heure et prévenez les éventuels occupants qui vous précèdent de libérer les lieux à l'heure dite. Envoyez un rappel de réunion avec un ordre du jour c'est toujours plus sympa que "revue de sprint N°31" aux participants.

De plus, nous préconisons de vous prévoir un temps de préparation pour cette démo. En effet, cela vous permettra de vous organiser sur les temps de passage, les tours de parole et de faire le point sur les rôles de chacun. Cette organisation est encore plus importante quand vous êtes dans un cadre d'agilité à l'échelle. Il faudra vous coordonner et vous réunir en amont afin que cette démonstration soit la plus fluide possible et montre aux participants la valeur

de ce nouvel incrément. De plus, cette préparation montrera à toute l'équipe la petite histoire et comment démontrer l'item (si une absence arrive le jour même, tout le monde sait quoi faire). Durant cette préparation, testez le matériel nécessaire et lancez les environnements. Faites-le aussi 15 minutes avant la démonstration (ça ne fait pas de mal) et prévoyez un plan B en cas de soucis matériel (capture d'écrans, vidéo, etc).

N'oubliez pas que, malgré l'effet de mode, l'agilité et Scrum ont encore mauvaise presse et sont souvent associés au vite-fait-mal-fait. Tous les détails comptent pour laisser à vos interlocuteurs une impression de maîtrise et de sérénité.

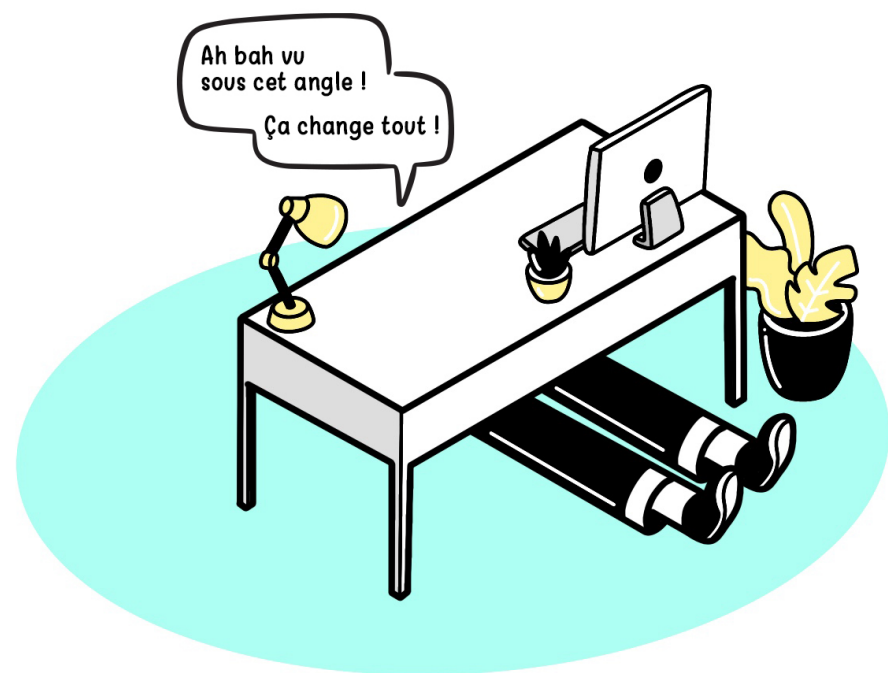


## Règle n°34

### Tu connais et tu lis les stories avant le sprint planning

Le sprint planning est un événement clé qui nécessite également de la préparation. En tant que Scrum Master, vous avez un rôle d'accompagnement du Product Owner sur ses activités liées à Scrum. Bien connaître les stories en arrivant à cette réunion permettra une animation efficace, même si le Scrum Master n'est pas le détenteur du savoir fonctionnel et technique. Vous pourrez rebondir lors des échanges, relancer les participants et leur rappeler les décisions prises.

Cela vous permettra aussi de vous assurer que les stories sont prêtes pour être développées dans un sprint. N'oubliez pas de rappeler à l'équipe qu'on peut poser toutes les questions que l'on veut, qu'il n'y a pas de questions bêtes, au contraire cela permet un alignement de l'équipe.



## Règle n° 15

### Choisir et varier les formats de rétrospective

Parce qu'un format unique et systématique peut limiter la façon d'aborder les problèmes, varier de temps en temps les formats permettra de changer de perspective et renouvellera l'énergie de cet événement. Évitez quand même de jouer avec tous les formats possibles et imaginables, vous n'êtes pas en train de faire une collection de timbres ou un tableau de chasse. Ce n'est pas la variété et le nombre qui importent mais l'efficacité. N'hésitez pas à aborder directement un thème ou un problème spécifique par le biais d'une

rétrospective. Par exemple, si vous voulez aborder un thème spécifique comme "Les tests", vous pouvez très bien faire un keep, drop, start rien que sur ce thème. De plus, en connaissant votre équipe vous pourrez choisir / inventer des formats qui pourraient les sortir d'une certaine monotonie et d'avoir un format qui leur correspond (Par exemple une rétro sur le thème jurassic park). Restez simple et n'essayez pas de résoudre la faim dans le monde.

Lien rétro jurassic : <https://blog.engineering.publicissapient.fr/2019/10/28/38316/>

lien de rétros : <https://blog.engineering.publicissapient.fr/2017/11/08/agile-a-chaque-sprint-sa-retrospective-les-classiques/>



## Règle n° 25

### Se préparer et avoir sa malette de Scrum master

Un des outils préférés des Scrum Masters est ... les post-its. Vous connaissez cet outil emblématique qui est de toutes les couleurs, de toutes les formes et qui vous sert en toute occasion.

Comment s'imaginer partir à une rétro sans post-it. Ce serait dommage d'avoir des ateliers sans les outils adéquats. Ou encore partir dans une salle sans feutres vélédas et vouloir prendre des notes sur un tableau blanc, vous faites comment ? Ou faire du management visuel sans scotch ou fil, vous tracez des traits au stylo bille sur le mur ? Tout ça pour dire : ayez toujours sur vous, dans votre sac, besace de super Scrum master de quoi travailler avec vos équipes.

Nous venons de parler d'outils en physique mais avec l'évolution de nos métiers. Nous sommes parfois plus souvent en remote qu'en physique. Les post-its, le scotch, les feutres disparaissent peu à peu pour laisser place à des outils virtuels. Ici même idée qu'en physique, nous vous conseillons de vous préparer des templates de rétrospectives, d'ateliers (delegation poker, squad health check) sur différents outils (et oui, certains outils sont bloqués dans certaines entreprises).

Vous l'aurez compris, l'idée est de se préparer, d'avoir des outils à disposition et de ne pas se retrouver au dépourvu !

Lien : <https://blog.engineering.publicissapient.fr/2020/03/23/pepito-kit-de-survie-en-travail-a-distance/>

# La résolution des obstacles

Vous n'avez peut-être pas réalisé que Scrum met en lumière les problèmes très rapidement. Ou peut-être vous l'avait-on dit, mais vous n'y croyiez pas, ou vous pensiez que c'était tant mieux. Maintenant que vous êtes dans le vif du sujet, vous voyez soudain apparaître des tas de petits, mais surtout des gros tracas. Tous les jours, il y en a des nouveaux : les membres de l'équipe doivent exposer ce qui les ralentit à chaque Daily Scrum. Et ils ne se privent pas de le dire ! Le Français est râleur, alors si en plus on lui demande explicitement d'extérioriser ce qui l'embête, il s'exprime ! À peine avez-vous adressé un problème que trois nouveaux apparaissent.

Gardez votre calme et attachez votre ceinture, les règles de ce chapitre devraient vous aider à y voir plus clair.



## Règle n° 14

### Ne te laisse pas enterrer sous les problèmes

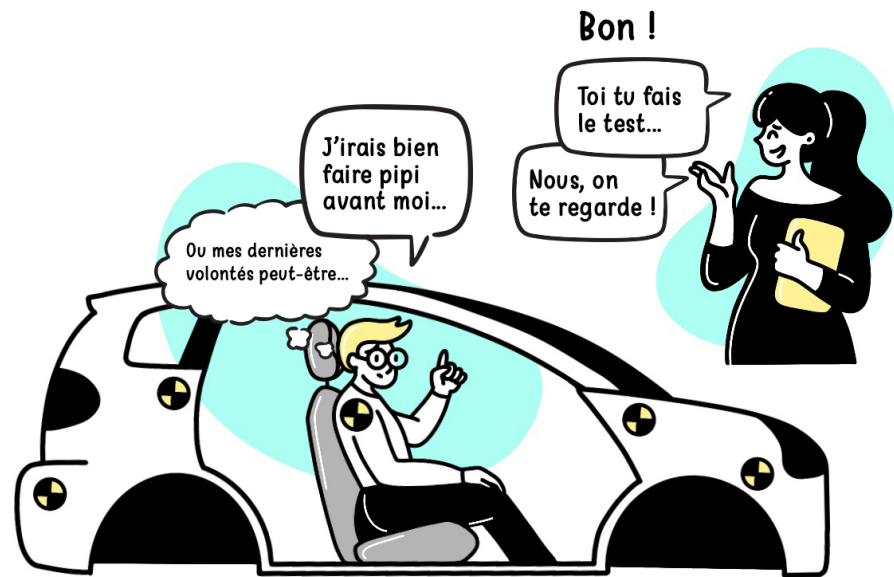
Cette règle nous a été inspirée par le livre *J'Arrête De Râler*<sup>12</sup> dans lequel on trouve cette petite fable : "Un paysan possède un puits à sec qu'il veut reboucher. Un jour son âne tombe dans le puits, le paysan se dit que son âne est vieux et que le sortir du puits va lui coûter cher, il décide de reboucher le puits avec l'âne dedans. L'âne râle pour sortir, mais le paysan commence à jeter de la terre dans le puits. Au bout de 5 mn, il n'entend plus l'âne se plaindre. Intrigué, il regarde dans le puits et découvre avec stupeur le comportement de l'âne : à chaque pelletée de terre, l'âne se secoue pour faire tomber la terre de son dos, puis il grimpe dessus. Petit à petit l'âne sortira du puits." Il est courant de penser que les problèmes qui nous ralentissent sont insolubles ou hors de notre contrôle. Gardez la tête froide et priorisez votre liste d'obstacles, cela ne sert à rien de vous éparpiller et d'essayer de tout résoudre.

Vous pouvez par exemple utiliser la matrice de Merill-Covey<sup>12</sup> pour classer vos obstacles selon l'importance et l'urgence, et garder votre attention sur un sous-ensemble. Avancez par petits pas, et n'hésitez pas à demander de l'aide. Il peut-être intéressant de conserver un affichage des obstacles résolus dans un "cimetière d'obstacles", cela vous donnera un sentiment d'accomplissement.

Cherchez aussi à identifier les obstacles "poupées russes", ceux qui partagent la même cause et dont la résolution supprimera plusieurs obstacles d'un seul coup. Aussi n'oubliez pas que parfois il peut être utile de se concentrer plus sur les solutions que sur l'origine du problème. Cela permet souvent d'identifier les premières étapes pour avancer (approche Solution Focus).

12- Cf bibliographie P 63

Lien : <https://blog.engineering.publicissapient.fr/2014/05/02/projet-it-kanban-episode-5-parie-i-gerer-les-obstacles-la-fiche-kaizen/>



## Règle n°20

### Il n'y a pas de "eux", il y a "nous"

Les réponses aux problèmes que l'on rencontre sont souvent plus claires lorsque nous reformulons nos griefs en utilisant "Nous" à la place de "Ils" ou "Eux". Par exemple : "il ne sait pas donner de réponse à notre question" essayez "nous ne savons pas donner de réponse à notre question". La signification est toute autre et vous comprenez que ce n'est pas un individu qui est en cause, mais une situation. Deuxième exemple : "il n'a pas fait son travail" essayez "nous n'avons pas fait notre travail". Ici aussi la reformulation suggère que vous avez peut-être manqué de proximité avec ce fameux "il" qui vous pourrit la vie. Nous sommes des acteurs à part entière du système social que représente l'organisation d'un projet ou d'une équipe produit. Nous avons donc une influence, aussi petite soit-elle sur ce système. Lorsque nous formulons nos griefs vis-à-vis des "Ils" ou des "Eux", assurons-nous d'avoir joué le jeu de la collaboration et de comprendre leur situation.

Quand bien même la collaboration est rompue, avons-nous escaladé correctement le problème en le présentant de manière factuelle aux parties prenantes ?

Souvenez-vous également que le Product Owner fait partie intégrante de l'équipe Scrum, il ou elle ne doit pas être un "il". Et cela est également valable pour les autres équipes avec lesquelles l'équipe Scrum collabore, surtout s'il s'agit d'agilité à l'échelle où plusieurs équipes collaborent ensemble pour délivrer un incrément de valeur unique. La collaboration et la notion d'équipe s'étend alors, le "nous" est l'illustration de cet état d'esprit.



## Règle n°22

### Connais tes détracteurs

Il y aura toujours des opposants à Scrum et à l'agilité, quand bien même vous avez le support de votre management et, bien sûr, de votre Product Owner. La relative tranquillité dont l'équipe dispose pour mener ses sprints en autogestion peut cacher des opposants qui vous attendent au tournant. Tout cela est normal, nous amenons un changement et celui-ci est accompagné, entre autres, de premiers adeptes (ceux qui disent oui) et de désengagés (ceux qui disent non) qui peuvent devenir détracteurs (ceux qui sabotent). Cherchez activement à identifier vos détracteurs et à connaître leurs attentes et leur façon de penser.

Ce faisant, prenez les devants sur la communication à effectuer auprès d'eux (cf règle 16) avec un vocabulaire qui correspond à leur champ lexical. N'oubliez pas que ces personnes vous en apprendront peut-être plus sur la réalité d'une situation ou sur vos faiblesses et votre marge d'amélioration que les observations que vous et votre équipe pouvez faire.

Pour aller plus loin, vous pouvez lire "Alerte sur la banque ! Réussir le changement dans n'importe quelles conditions".





## Règle n°26

### Explorer le chemin des possibles

Pour réussir, il faut savoir dépasser le cadre qui nous est fixé ou que l'on s'est fixé. Cette règle nous rappelle l'histoire des singes mouillés<sup>13</sup> que l'on retrouve dans L'Encyclopédie du Savoir Relatif et Absolu de Bernard Werber. Lorsque vous êtes bloqué, explorez toutes les possibilités, même celles qui ne sont pas envisageables parce que "c'est comme ça". Le fait d'explorer ces pistes va d'abord vous permettre de réévaluer les solutions envisageables avec un autre œil, vous découvrirez peut-être aussi que vous vous êtes autocensuré

sur certaines solutions. Échangez avec des pairs pour avoir leur avis et n'hésitez pas à faire de l'introspection, une pensée limitante peut apparaître. Apprenez à changer les règles du jeu pour gagner en efficacité.

13- [http://www.dailymotion.com/video/x8utcp\\_les-singes-mouilles-et-les-puces\\_tech#UKliw-MSWzI](http://www.dailymotion.com/video/x8utcp_les-singes-mouilles-et-les-puces_tech#UKliw-MSWzI)



## Règle n°27

### Il vaut mieux demander pardon qu'obtenir la permission

L'origine de cette règle nous vient de l'informaticienne américaine Grace Hopper : "Si c'est une bonne idée, allez-y et faites-la. Il sera plus facile de s'excuser que d'obtenir la permission"<sup>14</sup>. Parce que les sprints sont courts et que l'on peut rapidement se retrouver bloqué par de nombreuses

questions ou autorisations en attente de réponses, prenez les devants pour éviter de perdre du temps. L'attentisme est une source d'échec sur les projets agiles, prenez l'initiative. En revanche, conférez-vous à la règle 35 pour les limites d'application de cette règle.

14- [https://en.wikiquote.org/wiki/Grace\\_Hopper](https://en.wikiquote.org/wiki/Grace_Hopper)



## Règle n° 35

### Un Scrum Master sur le carreau est inutile

Corollaire de la règle n°27, votre pro-activité à gérer les obstacles ne doit pas vous conduire à votre mort professionnelle. Agissez dans les limites de ce que vous autorise la loi, la bienséance, le

respect des individus, et les politiques internes de sécurité. Exploitez, en revanche, les zones de flou à l'avantage de l'équipe.

## L'attitude et le charisme

Le charisme est un élément difficile à évaluer objectivement. Il est pourtant un ingrédient important pour qu'un Scrum Master réussisse. Une partie de son succès repose sur un savoir-être plutôt qu'un savoir-faire. Les règles de ce chapitre ne vont pas vous transformer en leader charismatique qu'on suivrait n'importe où par magie. En revanche, elles peuvent vous apporter des points de charisme supplémentaires pour vous aider à faire la différence.



## Règle n°7

### L'énergie de l'équipe c'est toi qui l'apporte

L'une des caractéristiques d'une équipe agile est de travailler dans une atmosphère dynamique. Un bouillonnement serein se dégage généralement des équipes agiles expérimentées. Cette tension positive est normalement provoquée par deux facteurs. Le premier est le fait de travailler en sprints de durée fixe qui définissent des échéances à court terme : cela oblige à produire un résultat opérationnel dans un laps de temps court. Le deuxième facteur est basé sur la motivation intrinsèque des individus. Dans son livre intitulé *La Vérité Sur Ce Qui Nous Motive*<sup>15</sup>, Daniel Pink<sup>16</sup> nous explique que la motivation vient de la combinaison de trois ingrédients : la finalité, l'autonomie et la maîtrise. Ces trois ingrédients se retrouvent clairement énoncés dans les principes du Manifeste Agile<sup>17</sup> :

- **Autonomie** : "Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés" et "Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées".
- **Maîtrise** : "Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité" et "À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence".
- **Finalité** : "Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée" et "Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet".

On aurait donc tendance à penser qu'avec les deux facteurs que sont la timebox et la motivation intrinsèque, un groupe d'individus réunis à l'occasion d'un projet agile sera naturellement dynamique et audacieux. Et pourtant, cela n'est souvent pas suffisant. Nous considérons qu'il y a un troisième facteur qui dynamise une équipe, c'est le **charisme** du Scrum Master. Un élément important de charisme est, à mon sens, l'énergie que l'on est capable de transmettre aux autres. Le Scrum Master doit être la première source d'énergie pour chacun des membres de l'équipe. Nous n'allons pas faire l'inventaire ici des pratiques que vous pouvez mettre en œuvre pour apporter de l'énergie au groupe, vous pouvez trouver de nombreux conseils par exemple avec le management 3.0<sup>18</sup>, dans des livres<sup>19</sup> ou des formations. Vous trouverez également un exemple de pratique dans la règle n°10. Rappelez-vous seulement d'alimenter régulièrement le dynamisme de l'équipe en apportant votre propre énergie dans le travail du groupe.

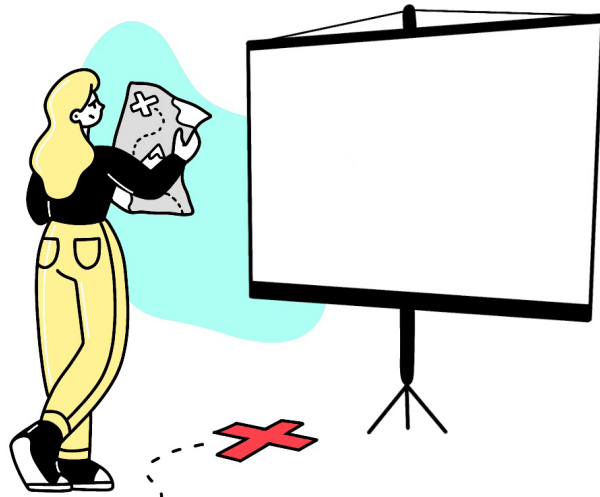
15- Cf bibliographie P.63

16- <http://www.danpink.com/>

17- <http://agilemanifesto.org/principles.html>

18- <https://management30.com/energize-people/>

19- Cf bibliographie P.63



## Règle n° 10

### Dans une réunion, ta place est debout à côté du tableau blanc

Cette règle n'est pas tant un comportement à suivre aveuglément qu'un conseil sur une posture à adopter et des actions à mener en réunion. Tout d'abord, en étant debout, l'énergie qu'on transmet est bien supérieure qu'en étant assis. Cela permet aussi d'être plus libre de ses mouvements et de changer son point de vue en se déplaçant si nécessaire. Cela vous installe tout de suite dans le rôle du facilitateur vis-à-vis des autres.

N'hésitez pas si cela est possible d'utiliser des outils de collaboration comme un paperboard ou du tableau blanc. Placez-vous à côté et animez votre réunion à l'aide de celui-ci. Le paperboard est utile pour prendre des notes et matérialiser les éléments échangés au cours de la réunion. La différence entre une personne assise qui prend des notes sur son PC et une personne debout qui prend des notes sur un paperboard peut sembler anecdotique et pourtant l'impact est bien supérieur. En notant sur un paperboard, vous faites preuve de transparence et vous avez un feedback instantané sur ce que vous écrivez. Certains utilisent un vidéoprojecteur

pour afficher en direct les notes qu'ils prennent. Cela transmet une toute autre énergie que lorsque l'on est debout. D'un point de vue extérieur, le fait de vous voir debout donne l'impression que vous prenez davantage les choses en main.

N'hésitez pas aussi, au vu de l'évolution de nos manières de travailler avec le télétravail, d'animer d'une autre façon vos ateliers. Demandez à vos collègues d'activer leur caméras, préparez vos supports... et inspirez-vous de nos articles sur l'animation à distance<sup>20</sup>.

Bien sûr, le Scrum Master n'est pas forcément l'animateur de toutes les réunions et cette règle peut viser n'importe quel membre d'une équipe agile, mais elle s'adresse en priorité au Scrum Master.

Vous pouvez renouveler votre façon d'animer des réunions en utilisant des éléments issus de Gamestorming, du Visual Meetings ou de Bikablo<sup>1</sup>.

En complément, vous pouvez vous référer aux règles 7 et 29.



## Règle n° 18

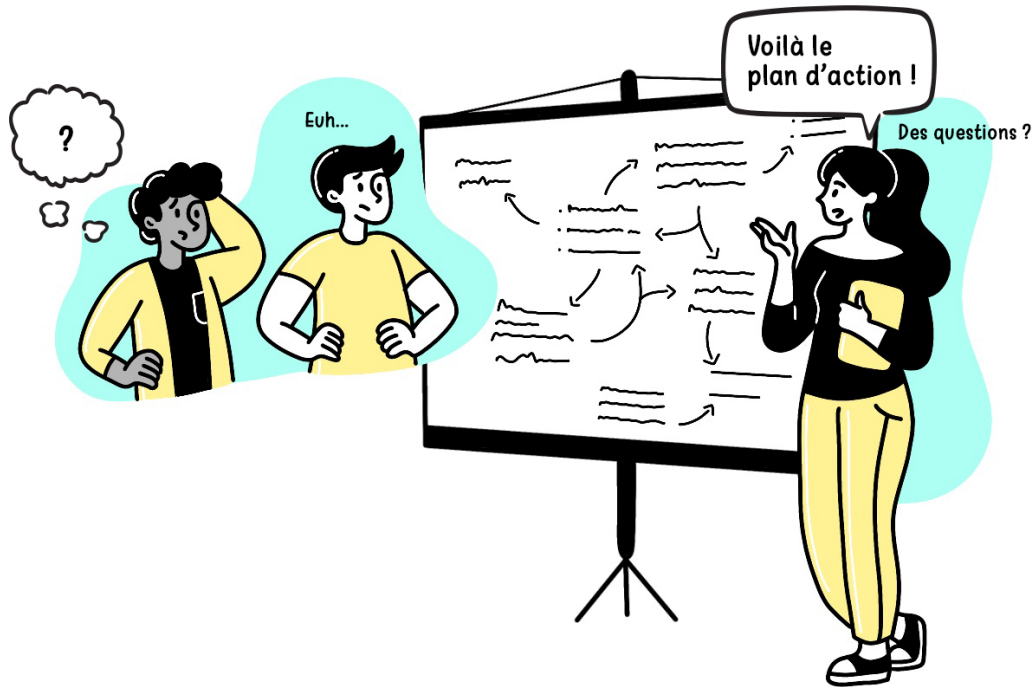
### Une équipe autogérée nécessite du Leadership

Un des aspects les plus compliqués du travail de Scrum Master est de permettre le travail en autonomie de chacun des membres, tout en s'assurant que les résultats sont alignés avec les attentes des parties prenantes. Nous trouvons cet aspect difficile car il nécessite une qualité rare : savoir gérer les échecs. Une équipe auto-organisée ne va pas magiquement produire voire dépasser les résultats attendus. Il est même probable qu'elle apparaisse de prime abord moins productive qu'une équipe organisée par spécialités, avec une division et une distribution stricte du travail à faire. Surperformer est un chantier de longue haleine et l'auto-organisation n'est pas synonyme de solitude ou de "débrouillez-vous". Il faut de la pro-activité et du courage pour exposer et affronter les problèmes. Les organisations de projet classiques avaient cette facilité que le chef de projet était responsable de presque tout et qu'on pouvait lui faire porter le chapeau en cas de problème.

Avec Scrum, c'est l'équipe qui est responsable, c'est à dire tout le monde (autrement dit personne, sans vouloir faire de cynisme déplacé). Nous avons souvent observé des équipes agiles ignorer consciemment les principaux problèmes qui les ralentissent plutôt que d'affronter courageusement la réalité. Ce n'est pas de la mauvaise volonté, mais plutôt un penchant naturel à se focaliser sur ce que l'on peut résoudre simplement et rapidement. Nous attendons d'un Scrum Master qu'il ne soit pas seulement un super-administrateur des événements Scrum d'une équipe, mais aussi la personne qui va pousser tout le monde à regarder la réalité en face quand il y a un décalage avec les attentes : l'objectif de sprint n'est pas atteint, le contenu de la release est décevant, les anomalies sont nombreuses, la vitesse décline, les relations sont tendues, untel ne travaille pas en équipe, etc.

<sup>20</sup> <https://blog.engineering.publicissapient.fr/tag/scrum-a-distance/>

\* Cf bibliographie P 63



## Règle n°29

### Tu es un facilitateur pas un dictateur

Cette dernière règle de savoir-être doit agir comme un contrepoids de la règle n°18. Malgré le fait que le Scrum Master soit le garant de la bonne application de Scrum et du respect du cadre Scrum (on ne négocie pas avec le cadre Scrum !), il n'est pas un dictateur. Il n'est pas question qu'il décide seul du travail à faire ou des actions à mener pour atteindre un objectif ou s'améliorer. Lorsque les résultats patinent ou que les erreurs s'accumulent, certains Scrum Masters (souvent ex Chef de Projet) sont tentés par deux expériences : adopter un mode de gestion directif de leur équipe ou

définir des processus stricts pour éviter les erreurs. Même si cela peut donner des résultats à court terme, on ne va pas dans le sens de responsabiliser les individus. Le dictateur se différencie du facilitateur par le fait qu'il arrive avec des solutions toutes faites à appliquer sans discernement et sans le consentement de l'équipe. En tant que Scrum Master, vous pouvez suggérer ou apporter des solutions qui ont fonctionné par le passé, mais assurez-vous du consentement de l'équipe sur l'application de ces recettes toutes faites.



## Règle n°13

### Reste à la page

Une formation Scrum Master ne saurait donner toutes les informations pour réussir dans votre métier. Lisez des livres, des blogs (celui de Publicis Sapient France est un très bon choix<sup>21</sup>), regardez des vidéos, venez participer aux conférences agiles

ou aux meetups, échangez avec vos pairs, créez des groupes d'intérêt dans votre organisation. En résumé, ne vous contentez pas du contenu du cours et de votre expérience. Et pensez aussi à regarder les mise à jour du Scrum Guide

21- <https://blog.engineering.publicissapient.fr>

Lien : <https://scrumguides.org>



## Règle n° 32

### Évite le blâme et la complaisance

Corollaire de la règle n°19, cette règle est souvent appliquée dans les techniques de négociation raisonnée : il s'agit d'être dur avec la situation mais doux avec les personnes. Le Scrum Master a une position délicate par rapport aux autres membres de l'équipe : ce n'est pas leur supérieur, mais ce n'est pas non plus un simple équipier.

Pour qu'un Scrum Master ait une influence sur la dynamique de groupe, il doit inspirer le respect et cela se construit avec 2 ingrédients : la loyauté et la cohérence.

Pour être respecté, un Scrum Master ne doit ni manier le blâme, au risque de se mettre des coéquipiers à dos, ni cacher la vérité, au risque de perdre le respect de ses coéquipiers.



## Règle n° 16

### Tu portes la transparence

Contribuez et vérifiez que les informations visuelles, écrites, et orales permettent aux parties prenantes de comprendre ce qu'il se passe : affichage physique, diffusion de synthèses de sprint, CR de réunion, etc. La propreté et l'organisation du tableau d'affichage sont souvent laissés pour compte et l'accent est mis sur le reporting papier. Pourtant, un tableau d'affichage qui ressemble à une envolée de post-its de toutes les couleurs ne donne ni une impression de sérieux, ni un sentiment de confiance. On préférera toujours confier un produit à une équipe qui a un tableau agréable et lisible, où les post-its suivent le même format et sont correctement alignés. Pour paraphraser Jurgen Appelo dans l'un de ses articles, si votre

tableau d'affichage donne l'impression qu'un tas de feuilles a fait une soirée disco, ça ne va pas vous aider à être pris au sérieux.

Il va sans dire qu'avec le travail à distance, la transparence est aussi nécessaire, en s'appuyant sur des tableaux de suivi, outils de ticketing, etc. afin de permettre une meilleure communication au sein de l'équipe et aussi autour d'elle.

Ces éléments sont d'autant plus vrais dans le cas d'une équipe d'une organisation à l'échelle.



## Règle n°23

### Tais-toi et SURTOUT écoute

Quand on vous expose un problème, ne sautez pas trop vite aux conclusions et prenez le temps d'écouter jusqu'au bout votre interlocuteur. Laissez un temps de silence lorsqu'il a fini (ex : 3 secondes), n'ayez point peur du silence et reformulez ses propos. Il est peut-être en train de vous donner la solution.

Cette règle s'applique bien sûr en rétrospective, mais également en toute occasion : rencontre en tête à tête avec les membres de l'équipe, réunion de travail, discussions informelles, etc.

Vous pouvez aussi, lorsqu'un sujet tourne en rond, relancer la discussion via des questions ouvertes. N'hésitez pas à creuser le problème avec le/les interlocuteurs en utilisant les 5 pourquoi<sup>22</sup>. Cela peut vous permettre en outre de trouver la cause originelle et par la suite de discuter sur des solutions.

# Allez plus loin !

Ce chapitre va clore la série des règles "standard" qui constituent le corps de nos règles du Scrum Master. Vous allez trouver ici des règles qui peuvent être difficiles à mettre en œuvre et qui nécessitent quelques trophées sur le tableau de chasse de votre expérience.

<sup>22</sup> <https://qualiblog.fr/outils-et-methodes/la-methode-des-5-pourquoi-pour-eradiquer-vos-problemes/>



## Règle n°24

### Veille à la pluralité des compétences de l'équipe

Pour assurer un travail en autonomie et livrer un incrément opérationnel du produit final, l'équipe Scrum doit être pluridisciplinaire. Le Scrum Master doit veiller à évaluer régulièrement que les compétences de l'équipe couvrent les compétences requises pour livrer un morceau du produit opérationnel en fin de sprint.

Une matrice de compétences peut être utilisée à cet effet pour cartographier les compétences et identifier les besoins.

De plus, il est important de veiller à ce que ces compétences soient partagées entre plusieurs personnes de l'équipe pour éviter tant que possible le risque associé au "bus factor" (en cas d'indisponibilité du seul sachant, l'équipe se retrouve bloquée dans son avancement).

On appelle compétences en T le fait d'être expert d'un sujet et d'avoir une connaissance moins approfondie sur un certain nombre d'autres sujets permettant de pouvoir intervenir en cas de nécessité. Des pratiques comme le travail en binôme ou plus (pair / mob programming) permettent de partager les compétences, les connaissances et les pratiques et par conséquent de disposer de compétences en T.

S'il manque des compétences dans votre équipe, ajoutez un obstacle à votre liste et agissez dessus.

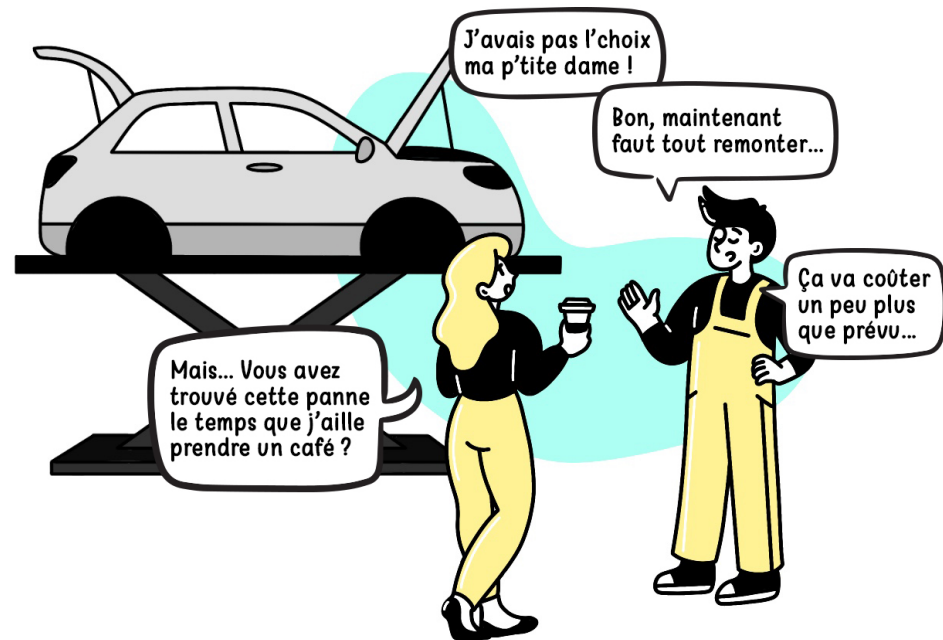
## Règle n°28

### Basez-vous toujours sur les faits et leurs impacts

Cette règle s'applique en toute circonstance et permet de dépassionner les débats. Assurez-vous de passer du monde des opinions au monde des faits. Limitez l'expression de jugement et proposez aux équipiers d'exprimer leur ressenti par rapport

à une situation factuelle. Proposez leur d'anticiper les impacts possibles. C'est particulièrement vrai pour le travail de rétrospective, mais n'importe quelle discussion est sujette à cette règle.





## Règle n° 11

### Ne deviens pas un garagiste

La gestion de la dette technique doit se faire en accord avec le Product Owner, en évitant les opérations obscures, il est indispensable d'être très transparent à ce sujet au risque de provoquer une "catastrophe nucléaire". Tout le monde aime savoir à quoi sert son argent.

Il faut alors comprendre le point de vue financier de la construction d'un produit numérique. Si un coût est engagé c'est dans le but d'avoir un retour sur investissement. Alors oui, certains produits ne seront pas utilisés à des fins lucratives, mais il s'agira dans tous les cas de faire un investissement conséquent et la gestion budgétaire s'en veut assidue.

Par conséquent en tant que Scrum Master, appuyez-vous sur les aspects financiers, ou à défaut sur les impacts que les dettes génèrent pour vendre les plans d'action sur la dette technique ou sur des opérations de maintenance à votre PO. Il doit être convaincu du bien fondé de ces actions et doit lui-même prioriser ces travaux face à des nouvelles fonctionnalités à ajouter au produit.

Lien : <https://blog.engineering-publicissapient.fr/2019/10/15/38240/>



## Règle n° 33

### Tu déploies tous les jours en environnement de test

La seule mesure d'avancement en agile est le logiciel opérationnel. Avoir la discipline de déployer régulièrement en test vous poussera à renforcer vos pratiques de tests et à lever les problèmes au plus tôt.

Déployer régulièrement en production vous fera mettre la barre encore plus haut en termes de qualité et vous obligera à considérer tous les aspects de votre produit (déploiement, migration, configuration, sécurité, performance).

Une démarche d'intégration continue et de déploiement continu vous aideront à fluidifier les passages des développements en environnement de test puis sur des environnements supérieurs jusqu'à la production. Le mouvement DevOps prône ce genre de pratiques en automatisant tout le processus, ceci est appelé pipeline de CI/CD<sup>23</sup>.

23- pipeline de CI/CD : <https://blog.engineering-publicissapient.fr/2020/01/06/quattrendre-de-votre-outil-de-ci-cd-en-2019/>

# Règles de crise

Les règles de crise ne sont à utiliser qu'en situation exceptionnelle. Leur objectif est d'amener un effet choc et de ne pas traîner pour prendre les décisions qui s'imposent. En effet, en situation de crise, plus vous traînez pour prendre des mesures radicales, plus la situation devient irréversible. Ce sont bien souvent des situations où la confiance est entachée et la difficulté pour la restaurer est proportionnelle à la durée de la crise. Parfois, même en ayant rétabli une situation compliquée, l'échec n'est pas évitable. Initialement, les règles de crise étaient numérotées de 40 à 45, mais avec le temps et quelques remarques pertinentes, nous avons transformé une règle principale en règle de crise car elle ne méritait pas une attention continue.



## Règle n° 17

### Transparence et pas indécence

En situation de crise, assurez-vous que les informations transmises par l'équipe sont formulées d'une façon qui ne porte pas offense au mandat de vos parties-prenantes. Parcourez votre affichage visuel avec les yeux de votre détracteur (cf règle 22) et assurez-vous que la formulation n'apporte pas une ambiguïté qui pourrait être interprétée en votre défaveur. L'idée n'est pas ici de réduire la transparence mais plutôt de vérifier la bonne formulation des informations qui émanent de l'équipe.

Prenons l'exemple qui consisterait à épingler sur le tableau d'obstacles de l'équipe un post-it qui mentionnerait "manque de maîtrise de la base de données". Ceci pourrait laisser croire au management que l'équipe entière ne maîtrise pas la base de données. Alors qu'en réalité l'obstacle ne concernerait qu'un manque de documentation sur la base de données qui freinerait les nouveaux arrivants dans leur maîtrise du modèle de données.



## Règle n° 40

Une bonne prédictibilité vient d'une bonne mesure de la vélocité et d'une bonne estimation de l'effort de chaque story

Dans les situations conflictuelles, il arrive que le story point ait une signification obscure pour les néophytes. Cela contribue à entretenir un flou autour de la capacité de l'équipe. C'est alors que la vélocité de l'équipe ou que l'effort nécessaire pour réaliser une story donnée est remis en cause.

C'est là que vous, Scrum Master, vous devez démontrer que ces mesures sont scientifiques. Vous devez avoir l'historique de vos sprints précédents pour pouvoir justifier votre vélocité. De même vous devez avoir une base de références de user stories "done" vous permettant de justifier l'effort estimé pour les stories à venir (par simple comparaison).

Si vous êtes challengé pour estimer en jour/homme (en autre pour de la gestion budgétaire), tentez de ne pas céder à cette "fausse" facilité, vous en perdriez en prédictibilité et en facilité d'estimation. Si nécessaire pour retrouver le coût d'une Epic, on peut utiliser le ratio jour/homme-point d'effort très simplement.

Enfin, si vous sentez votre prédictibilité en berne, vous pouvez afficher pour chacun de vos engagements de sprint un chiffre (entre 1 et 5) représentant votre confiance par rapport à l'atteinte de vos objectifs. Si vous êtes capable d'expliquer le chiffre par des éléments factuels, cela peut vous aider à communiquer sur vos difficultés et éventuellement obtenir de l'aide !



## Règle n° 41

Gère les pollueurs en revue

La revue de sprint est une étape importante de collaboration entre développeurs et business. Elle peut avoir un effet extrêmement positif sur les équipes mais également se révéler très démotivante si vous ne prenez pas un minimum de précautions. Dans les situations de crise, selon le contexte :

- Exceptionnellement, n'autoriser que les personnes qui ont participé au sprint planning à venir à la revue d'itération.
- Préparer et partager le déroulement de la revue en prévoyant les moments pour les questions plutôt vers la fin.

- Partager si besoin les difficultés rencontrées par l'équipe en basant sur des faits et des chiffres à l'appui.

Cela pour plusieurs aspects :

- Éviter le risque de laver son linge sale devant les parties prenantes.
- Ne pas porter préjudice à la confiance et l'image du livrable vis à vis des parties prenantes.
- Repartir sur des bases saines.
- Ne pas dégrader la confiance entre le PO et l'équipe.



## Règle n° 42

### Range ton désordre

L'agilité et Scrum laissent une certaine marge de manœuvre à l'expérimentation et la gestion des échecs. En revanche, en situation de crise il faut faire face à la réalité du business et parfois, faire l'effort nécessaire pour revenir à une situation propre : absorption de dette technique, livraison

de stories indispensables, nettoyage des bureaux, automatisation des tests et des déploiements, etc. Une surcharge temporaire de travail ne remet pas en cause la notion de rythme soutenable ou de priorisation.

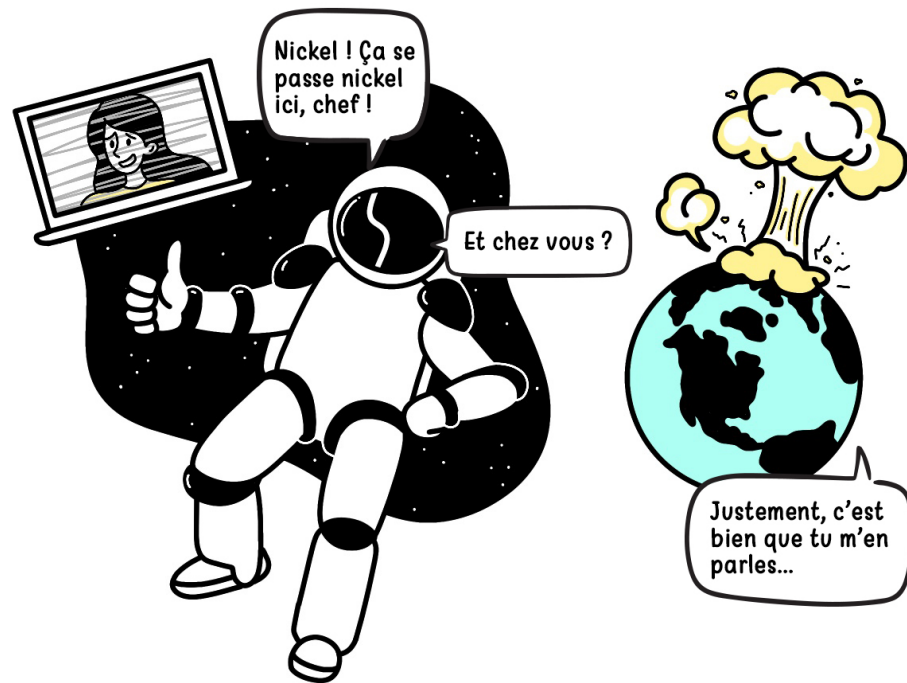


## Règle n° 43

### Pas de sprint entier de refactoring

Les sprints entiers de refactoring font passer un mauvais message auprès des donneurs d'ordre. En situation de crise, c'est l'aveu que vous avez perdu le contrôle de votre produit. Si votre dette technique

est trop importante, essayez d'identifier au moins 2 sujets fonctionnels qui pourraient être adressés avec de la dette technique.



## Règle n° 44

### Des collaborateurs et non pas des ressources

On ne le répètera jamais assez, les produits se construisent avec des personnes. En situation de crise, il est d'autant plus important de considérer l'humain pour restaurer un climat serein et de confiance entre les membres de l'équipe.

Si vous identifiez des dysfonctionnements, il va de soi qu'il faut les régler au plus vite avant que cela ne se gangrène : provoquer des discussions constructives avec les personnes concernées.

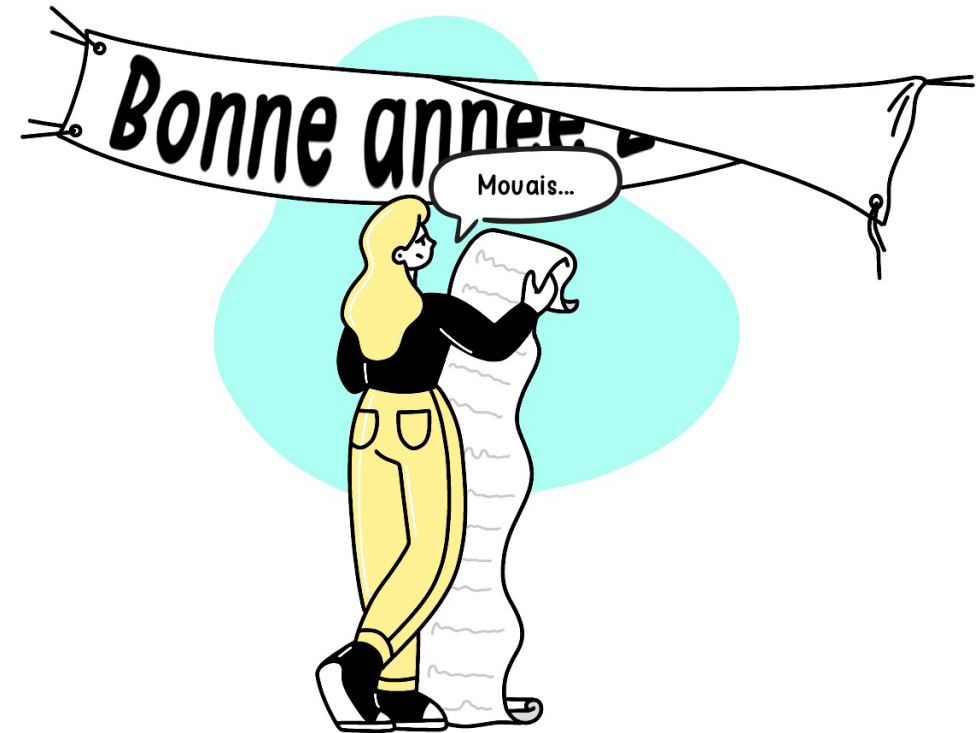
Pour ressouder l'équipe, n'hésitez pas à créer des moments de vie. Qu'ils soient studieux comme des travaux en groupe, voire plus conviviaux, tels que des moments cafés ou déjeuners.

L'important, c'est de vivre des événements plus ou moins positifs tous ensemble en tant qu'équipe.

Dans le cas où votre équipe est distribuée, cela reste possible de manière numérique. Et pourquoi ne pas proposer de se retrouver tous ensemble dans un même site de temps à autre ?

N'hésitez pas à vous inspirer de nos articles sur Scrum à distance<sup>24</sup>.

<sup>24</sup> - <https://blog.engineering-publicissapient.fr/tag/scrum-a-distance/>



## Règle n° 45

### À la rétrospective, 2 actions d'amélioration maximum

Les situations de crise ont cette caractéristique surprenante qu'elles génèrent un nombre de problèmes incalculables. Nous faisons du cynisme mal placé, ce n'est pas étonnant à vrai dire. La crise est exothermique, si vous ne trouvez pas un moyen de la contenir elle va devenir hors de contrôle, c'est un réacteur nucléaire en fusion dont vous n'imaginez pas les réactions en chaîne. Dans ces situations, on a une tendance naturelle à passer beaucoup de temps en rétrospective: il y a des ressentis forts qui se créent et ont besoin de s'exprimer, le nombre de problèmes

à résoudre peut être énorme. Allonger le temps de ses rétrospectives ou bien prendre un grand nombre d'actions est exactement la bonne façon de prolonger la crise. Quand vous êtes dans le réacteur nucléaire, vous avez plutôt intérêt à consacrer tout le temps possible à stopper la fusion plutôt qu'à réfléchir comment résoudre les problèmes de sécurité des centrales atomiques face au tsunami. Limitez la durée de la rétrospective (par exemple à 1h) et dégagez au plus 2 actions d'amélioration sur les problèmes les plus graves.

## Règle n° 46

### Pour garder la force, la motivation tu entretiendras



On ne le répètera jamais assez, les produits se Nous avons abordé le sujet de la motivation dans la Règle n°7 - L'énergie de l'équipe c'est toi qui l'apportes ; mais il faut savoir que la motivation d'une équipe n'est pas quelque chose de gravée dans le marbre. En tant que Scrum Master, comment puis-je savoir si mon équipe manque de motivation ? Il n'y a malheureusement pas de recette magique pour trouver une réponse claire à cette question, mais certains éléments de réponses peuvent être plus évidents :

- Si ces éléments se répètent plusieurs fois, penchez-vous sur la motivation de votre équipe

- Rater un objectif de sprint  
- Baisse de vitesse apparente  
- Baisse de participation à la rétrospective  
- Manque de communication de manière générale  
- Manque d'engagement et/ ou participation en sprint planning, affinage, ...

- Un excès de négativité en rétrospective,
- Des tensions au sein de l'équipe et/ou en dehors non traitées
- Etc...

Lorsque l'on remarque un manque de motivation, il faut arriver à trouver rapidement à quoi ce manque est lié. Nous n'étayerons pas les différentes causes possibles mais pour mieux les comprendre, vous pouvez regarder du côté des motivations intrinsèques (le sens) plutôt que les motivations extrinsèques (rémunération). Les personnes manquent-elles d'autonomie, de maîtrise ou de sens dans ce qu'elles font ?

Selon les différentes raisons possibles, voici quelques actions :

- Renforcez la positive attitude ! Promouvoir le remerciement des participants aux événements énoncer ce pour quoi les gens ont de la gratitude. Ça renforce un sentiment de cohésion d'équipe tout en mettant en valeur ses forces et qualités.

- Changez le format de rétro, l'endroit, le contexte, commencez par un ice-breaker pour redynamiser et enlever une certaine monotonie. Assurez-vous que les sujets et problèmes remontés sont suivis. Qu'il n'y ait pas d'effet "parole en l'air"

- Communiquez ! On ne misera jamais trop sur la communication. Organisez des discussions en individuel de façon informel pour récupérer des informations. Faites de simples discussions afin que tous les membres fassent connaissance et créent cette appartenance à une équipe. Ces discussions peuvent être aussi en petits groupes. L'essentiel est de trouver des points d'entrées pour ramener de la motivation et créer un nouveau cercle vertueux.

Un problème d'autonomie de l'équipe ? De maîtrise du produit ?... Les actions qui en découlent peuvent être simples comme aborder un certain sujet lors de la rétrospective ou plus subtiles comme renforcer le sentiment d'environnement de sécurité au sein de l'équipe.

Les différents problèmes/solutions possibles sont légion et nous n'entrerons pas dans le détail, mais il est important d'être conscient des variations dans la motivation de l'équipe pour pouvoir agir. Pour cela, le cycle d'Hudson vous permet de mieux appréhender ces variations en les reconnaissant.

## Vos règles du jeu



Chers Scrum Masters,

*Vous avez lu avidement ce livre depuis le début, et nous vous en remercions. Si vous possédez un semblant d'esprit mathématique, vous avez dû remarquer qu'il manque 9 règles pour compléter la liste. Nous vous en avons promis 50 et n'en avons donné que 41.*

*Ce n'est pas par hasard si le titre de ce chapitre ressemble à un slogan pour parfum. Les 9 dernières vont avoir une saveur légèrement différente, c'est ici et maintenant que vous entrez en scène : ces dernières règles sont les vôtres !*

*Scrum n'est efficace que s'il est contextualisé à votre organisation et aux individus qui la composent. Utilisez les numéros 36 à 39 pour formaliser vos propres règles, celles que vous avez apprises avec votre expérience et surtout celles qui correspondent à votre style de Scrum Mastering. N'hésitez pas à nous faire part de vos règles en nous envoyant un message.*

*Les règles 47 à 50 sont réservées à vos règles de crise. Indiquez dans cette séquence les règles que vous avez apprises à vos dépens dans des situations douloureuses, conflictuelles ou négatives. N'utilisez ces règles qu'en situation exceptionnelle.*

## Règle n° 38



---

## Règle n° 39



---

## Règle n° 47



---

## Règle n° 48



---

## Règle n° 49



## Règle n° 50



# Bibliographie

*Alerte sur la banquise Réussir le changement dans n'importe quelles conditions* - Holger Rathgeber et John Kotter - PEARSON FRANCE

*Bikablo 2.0* - Holger Scholz , Martin Haussmann - Neuland

=> (*Bikablo* en français ) Uzmo - Martin Haussmann - Eyrolles

*Gamestorming*- Sunni Brown, Dave Gray, James Macanujo - DIATEINO EDS

*Jarrête de râler* - Christine Lewicki – Eyrolles

*La vérité sur ce qui nous motive* – Daniel H. Pink - FLAMMARION

*Management 3.0* - Jurgen Appelo - Addison-Wesley Professional



# Remerciements et conclusion

Vous arrivez à la fin de ce guide. Nous espérons que cette lecture vous aura autant plu qu'à nous de l'écrire.

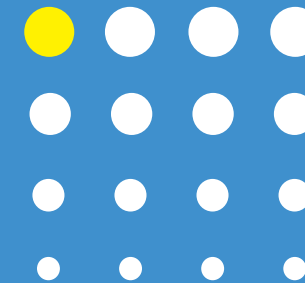
Cette réécriture nous aura demandé beaucoup d'énergie c'est pourquoi nous souhaitons remercier tous les contributeurs qui nous ont permis d'atteindre ce résultat :

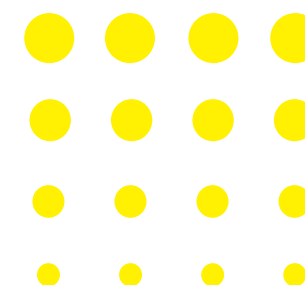
- Merci à Gilles Mantel et à Jean-Laurent de Morlhon qui ont été à l'origine de la première version et sans qui ce travail d'équipe n'aurait pas été possible,
- Merci à toute l'équipe 2021 Publicis Sapient France qui nous a aidé dans la relecture,
- Merci à l'équipe Marketing de Publicis Sapient France qui nous a aidé à mettre en forme nos idées et à créer cet objet dont nous sommes fiers,
- Merci également à Camille Pfaender pour la création des illustrations de ce guide.

Ce guide est notre héritage que nous entretenons . D'une première version qui nous a énormément inspirés dans notre quête d'épanouissement en tant que Scrum master, nous vous transmettons le flambeau à notre tour. Et nous espérons qu'il vous sera tout aussi utile.

Agilement votre,

Morgane Eckert, Olivier Marquet, Gaëlle Mouraret, Thomas Cousin, Xavier Spinosi, Jean-Noël Hany, Alexandre Kuoch et Florent Adolphe.





**Publicis Sapient**

94 Avenue Gambetta, 75020 Paris

Toutes les informations sur :  
[publicissapient.fr](http://publicissapient.fr)